Algebraic Approach to Promise Constraint Satisfaction

LIBOR BARTO and JAKUB BULÍN, Charles University, Czechia
ANDREI KROKHIN and JAKUB OPRŠAL, Durham University, UK

The complexity and approximability of the constraint satisfaction problem (CSP) has been actively studied over the past 20 years. A new version of the CSP, the promise CSP (PCSP), has recently been proposed, motivated by open questions about the approximability of variants of satisfiability and graph colouring. The PCSP significantly extends the standard decision CSP. The complexity of CSPs with a fixed constraint language on a finite domain has recently been fully classified, greatly guided by the algebraic approach, which uses polymorphisms—high-dimensional symmetries of solution spaces—to analyse the complexity of problems. The corresponding classification for PCSPs is wide open and includes some long-standing open questions, such as the complexity of approximate graph colouring, as special cases.

The basic algebraic approach to PCSP was initiated by Brakensiek and Guruswami, and in this article, we significantly extend it and lift it from concrete properties of polymorphisms to their abstract properties. We introduce a new class of problems that can be viewed as algebraic versions of the (Gap) Label Cover problem and show that every PCSP with a fixed constraint language is equivalent to a problem of this form. This allows us to identify a "measure of symmetry" that is well suited for comparing and relating the complexity of different PCSPs via the algebraic approach. We demonstrate how our theory can be applied by giving both general and specific hardness/tractability results. Among other things, we improve the state-of-the-art in approximate graph colouring by showing that, for any $k \geq 3$, it is NP-hard to find a $(2k-1)$-colouring of a given k-colourable graph.

CCS Concepts: • **Theory of computation → Problems, reductions and completeness**;

Additional Key Words and Phrases: Constraint satisfaction, promise problem, approximation, graph colouring, polymorphism

ACM Reference format:
Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. 2021. Algebraic Approach to Promise Constraint Satisfaction. *J. ACM* 68, 4, Article 28 (July 2021), 66 pages.
https://doi.org/10.1145/3457606

Preliminary versions of parts of this article were published in the proceedings of STOC 2019 [38] and LICS 2019 [10].
Libor Barto has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant Agreement No. 771005 ERC, CoCoSym). Jakub Bulín was supported by the Austrian Science Fund project P29931 FWF, the Czech Science Foundation project 18-20123S, Charles University Research Centre program UNCE/SCI/022 and PRIMUS/SCI/12. Andrei Krokhin and Jakub Opršal were supported by the UK EPSRC grant EP/R034516/1. Jakub Opršal has also received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant Agreement No. 681988, CSP-Infinity).

Authors' addresses: L. Barto, Department of Algebra, Faculty of Mathematics and Physics, Charles University, Sokolovská 83, 186 00, Prague, Czechia; email: libor.barto@gmail.com; J. Bulín, Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00, Prague, Czechia; email: jakub.bulin@mff.cuni.cz; A. Krokhin and J. Opršal, Department of Computer Science, Durham University, Upper Mountjoy, South Road, DH1 3LE, Durham, UK; emails: {andrei.krokhin, jakub.oprsal}@durham.ac.uk.

0004-5411/2021/07-ART28
https://doi.org/10.1145/3457606

1 INTRODUCTION

What kind of inherent mathematical structure makes a computational problem tractable, i.e., polynomial time solvable (assuming $\mathsf{P} \neq \mathsf{NP}$)? Finding a general answer to this question is one of the fundamental goals of theoretical computer science. The ***constraint satisfaction problem (CSP)*** and its variants are extensively used towards this ambitious goal for two reasons: On the one hand, the CSP framework is very general and includes a wide variety of computational problems; and on the other hand, this framework has very rich mathematical structure providing an excellent laboratory both for complexity classification methods and for algorithmic techniques.

The basic aim of a CSP is to find an assignment of values from some domain to a given set of variables that satisfies given constraints, where each constraint is given as a predicate that involves a given subset of variables. Usually, the CSP is formulated as a decision problem: The goal is to decide whether there is such a satisfying assignment. Important variants of the CSP include counting and optimisation (both exact and approximate) and extensions of the basic framework, e.g., by using real-valued functions instead of relations/predicates (to specify valued constraints) or allowing global constraints (see surveys in Reference [70]). Since the basic CSP is NP-complete (and, for other variants, as hard as it can be) in full generality, a major line of research in the CSP focuses on identifying tractable cases and understanding the mathematical structure enabling tractability (see Reference [70]).

One particular family of CSPs that receives a great amount of attention consists of the CSPs with a fixed *constraint language* [48, 70], i.e., with a restricted set of types of constraints. Since constraints are usually given by relations, a constraint language is simply a set Γ of relations on a domain A. The restricted CSP where only relations from Γ can specify constraints is denoted by $\mathrm{CSP}(\Gamma)$. Many computational problems, including various versions of logical satisfiability, graph colouring, and systems of equations can be represented in this form [48, 70]. It is well-known [48] that the basic CSP can be cast as a homomorphism problem from one relational structure to another (the latter is often called a *template*), and we will use this view. Problems $\mathrm{CSP}(\Gamma)$ correspond to the case when the template structure is fixed. There is an active line of research into CSPs with infinite A (see, e.g., References [20, 22, 82]), but throughout this article, we assume that A is finite (unless specified otherwise).

In Feder and Vardi [48] conjectured that for each finite constraint language Γ, the (decision) problem $\mathrm{CSP}(\Gamma)$ is either in P or NP-complete. This conjecture inspired a very active research programme in the past 20 years, which recently culminated in two independent proofs of the conjecture, one by Bulatov [36] and the other by Zhuk [92, 93] (along with similar classification results for other CSP variants, e.g., References [14, 35, 67, 90]). All of these proofs heavily use the so-called algebraic approach to the CSP. On a very high level, this approach uses multivariate functions that preserve relations in a constraint language (and hence solution sets of problem instances), called *polymorphisms*. Thus, polymorphisms can be seen as high-dimensional "symmetries" of solution sets. Roughly, lack of strong enough symmetries implies hardness of the corresponding problem, while presence of such symmetries implies tractability. This approach was started in a series of papers by Jeavons et al., e.g., [61, 62], where the key role of polymorphisms was established. It was then taken to a more abstract level by Bulatov et al. [33, 34], where an abstract view on polymorphisms was used, through universal algebras and varieties formed by algebras—this allowed a powerful machinery of structural universal algebra to be applied to the CSP. Another important general methodological improvement was Reference [16], where it was shown that special equations of simple form satisfied by polymorphisms govern the complexity of CSPs. Even though Reference [16] did not impact specifically on the resolution of the Feder-Vardi conjecture, it strongly influenced the present article.

A new extended version of the CSP, the so-called ***promise constraint satisfaction problem* (*PCSP*)**, has recently been introduced [7, 27, 29], motivated by open problems about (in)approximability for variants of SAT and graph colouring. Roughly, this line of research in approximability concerns finding an approximately good solution to an instance of a (typically hard) problem when a good solution is guaranteed to exist (see discussion and references in Reference [28]). Approximation can be understood in terms of relaxing constraints or in terms of counting satisfied/violated constraints—in this article, we use the former. Specifically, in the PCSP, each constraint in an instance has two relations: a "strict" one, and a "relaxed" one, and one needs to distinguish between the case when an instance has a solution subject to the strict constraints and the case when it has no solution even subject to the relaxed constraints. One example of such a problem (beyond CSPs) is the case when the only available strict relation is the disequality on a k-element set and the corresponding relaxed relation is the disequality on a c-element set (with $c \geq k$)—the problem is then to distinguish k-colourable graphs from those that are not even c-colourable. This problem (and hence the problem of colouring a given k-colourable graph with c colours) has been conjectured NP-hard, but the question in full generality is still open after more than 40 years of research. We give more examples later. Note that if the strict form and the relaxed form for each constraint coincide, then one gets the standard CSP, so the PCSP framework greatly generalises the CSP.

The problem of systematically investigating the complexity of PCSPs (with a fixed constraint language) was suggested in References [7, 27, 29]. We remark that, beyond CSPs, the current knowledge of the complexity landscape of PCSPs is quite limited, and we do not even have analogues of full classification results for graph homomorphisms [57] and Boolean CSPs [86]—which were the most important basic special cases of CSP complexity classifications that inspired the Feder-Vardi conjecture. The quest of complexity classification of PCSPs is of great interest for a number of reasons. It brings together two very advanced methodologies: analysing the complexity of CSPs via algebra and the approximability of CSPs via PCP-based methodology, hence the possibility of fruitful cross-fertilisation and influence beyond the broad CSP framework. It is perfect for further exploring the thesis that (high-dimensional) symmetries of solution spaces are relevant for complexity—which is certainly true for most CSP-related problems, but may be applicable in a wider context. Finally, this quest includes long-standing open problems as special cases.

Related work. An accessible exposition of the algebraic approach to the CSP can be found in Reference [15], where many ideas and results leading to (but not including) the resolution [36, 92, 93] of the Feder-Vardi conjecture are presented. Reference [70] contains surveys about many aspects of the complexity and approximability of CSPs.

The first link between the algebraic approach and PCSPs was found by Austrin, Håstad, and Guruswami [7], and it was further developed by Brakensiek and Guruswami [26, 27, 29, 30]. They use a notion of polymorphism suitable for PCSPs to prove several hardness and tractability results. Roughly, the polymorphisms of a PCSP (template) are multivariate functions from the domain of its "strict" relations to that of its "relaxed" relations that map each strict relation into the corresponding relaxed relation. For example, the n-ary polymorphisms of the PCSP template corresponding to k vs. c graph colouring (we say polymorphisms from $\mathbf{K}_k$ to $\mathbf{K}_c$) are the homomorphisms from the nth Cartesian power of $\mathbf{K}_k$ to $\mathbf{K}_c$, i.e., the c-colourings of $\mathbf{K}_k^n$. It is shown in Reference [30] that the complexity of a PCSP is fully determined by its polymorphisms—in the sense that two PCSPs with the same set of polymorphisms have the same complexity up to log-space reductions.

Much of the previous work on the complexity of PCSPs was focused on specific problems, especially on approximate graph and hypergraph colouring and their variants. We describe this in more detail in Examples 2.7–2.13 in the next section. Let us note here that, despite much effort, there is a huge gap between known algorithmic and NP-hardness results for colouring 3-colourable graphs with c colours: The best-known NP-hardness result (without additional assumptions) prior to this article went only as far as $c = 4$ [53, 64], while the best (in terms of c) known efficient algorithm uses roughly $O(n^{0.199})$ colours to colour an n-vertex 3-colourable graph [63]. There are also hardness results concerning hypergraph colouring with a super-constant number of colours, e.g., References [5, 52], but problems like this do not fall directly into the framework that we consider in this article.

We remark that appropriate versions of polymorphisms have been used extensively in many CSP complexity/approximability classifications: standard polymorphisms for decision and counting CSPs, for approximating Min CSPs and testing solutions (in the sense of property testing) [14, 15, 35, 36, 40–42, 92, 93], fractional polymorphisms for exact optimisation problems [67, 90], α-approximate polymorphisms for approximating Max CSPs [32]. In all cases, the presence of nice enough polymorphisms (of appropriate kind) leads directly to efficient algorithms, while their absence leads to hardness. Interestingly, it was shown by Brown-Cohen and Raghavendra [32] that the Unique Games Conjecture is equivalent to the NP-hardness of approximating Max CSPs beyond a specific numerical parameter of their (nice enough) approximate polymorphisms.

Our contribution. The main contribution of the present article is a new abstract algebraic theory for the PCSP. A crucial property of polymorphisms for PCSPs is that, unlike in CSPs, they cannot be composed (as functions). The ability to compose polymorphisms to produce new polymorphisms was used extensively in the algebraic theory of CSPs. This could be viewed as a serious limitation on the applicability of the algebraic approach to PCSPs. Alternatively, it might indicate that the ability to compose is not that essential and that a composition-free abstract algebraic theory for PCSPs (and hence for CSPs) can be developed. Our results suggest that the latter is in fact the case.

We show that certain abstract properties of polymorphisms, namely, systems of minor identities (i.e., function equations of a simple form) satisfied by polymorphisms, fully determine the complexity of a PCSP. This shifts the focus from concrete properties of polymorphisms to their abstract properties. Systems of minor identities satisfied by polymorphisms provide a useful measure of how much symmetry a problem has. This measure gives a new tool to compare/relate the complexity of PCSPs far beyond what was available before. We envisage that our article will bring a step change in the study of PCSPs, similar to what Bulatov et al. [33, 34] did for the CSP. Let us explain this in some more detail.

To be slightly more technical, a *minor identity* is a formal expression of the form

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

where f, g are function symbols (of arity n and m, respectively), $x_1, \dots, x_n$ are variables, and $\pi : [m] \to [n]$ (we use notation $[n] = \{1, \dots, n\}$ throughout). A minor identity can be seen as an equation where the function symbols are the unknowns, and if some specific functions f and g satisfy such an identity, then f is called a *minor* of g. We use the symbol $\approx$ instead of $=$ to stress the difference between a formal identity (i.e., equation involving function symbols) and equality of two specific functions. A *minor condition* is a finite system of minor identities (where the same function symbol can appear in several identities). A bipartite minor condition is one where sets of function symbols appearing on the left- and right-hand sides of the identities are disjoint. Such a condition is said to be satisfied in a set $\mathscr{F}$ of functions if it is possible to assign a function from $\mathscr{F}$ of the corresponding arity to each of the function symbols in such a way that all the identities are simultaneously satisfied (as equalities of functions, i.e., for all possible values of the x_i's).

Informally, the main results of our new general theory state that

(A) If every bipartite minor condition satisfied in the polymorphisms of (the template of) one PCSP Π_1 is also satisfied in the polymorphisms of another PCSP Π_2, then Π_2 is log-space reducible to Π_1 (see Theorem 3.1 for a formal statement). Moreover, we characterise the premise of the above claim in many equivalent ways (see Theorem 4.12 and Corollary 9.5).

(B) Every PCSP Π is log-space equivalent to the problem deciding whether a given bipartite minor condition is satisfiable by projections/dictators or not satisfiable even by polymorphisms of Π (see Theorem 3.12).

The first of the above results establishes the key role of bipartite minor conditions satisfied in polymorphisms—in particular, the hardness or tractability of a PCSP can always be explained on this abstract level, since two PCSPs have the same complexity if their polymorphisms satisfy the same bipartite minor conditions. Moreover, this abstract level allows one to compare two PCSPs, even when it does not makes sense to compare their sets of polymorphisms inclusion-wise (say, because the functions involved are defined on different sets). The second result establishes that every PCSP is equivalent to what can be viewed as an algebraic version of the Gap Label Cover problem, which is the most common starting point of PCP-based hardness proofs in the inapproximability context. Our result uses the fact the Label Cover can be naturally interpreted as the problem, which we call MC, of checking triviality of a system of minor identities. The gap version of MC has an algebraic component in place of the quantitative gap of Gap Label Cover. In particular, result (B) can provide a general approach to proving NP-hardness of PCSPs—via analysis of bipartite minor conditions satisfied by polymorphisms.

Our general algebraic reductions use constructions that resemble dictatorship tests, which are also present in many inapproximability proofs. All known hardness results for PCSPs (possibly except References [59, 65]) can be proved using our view. We show how our theory can be used to obtain general algebraic sufficient conditions for NP-hardness of a PCSP—we give an example of this (Theorem 5.9) that covers several cases considered in the literature before. Our theory can also be used to translate specific hardness results into general hardness results (which of course can then be applied in new specific cases). We demonstrate how this works and answer specific questions from References [26, 27] by showing, among other things, the following:

(C) For any $k \geq 3$, it is NP-hard to distinguish k-colourable graphs from those that are not $(2k-1)$-colourable (see Theorem 6.5).

In particular, it follows that it is NP-hard to 5-colour a 3-colourable graph. This might seem a small step towards closing the big gap in our understanding of approximate graph colouring, but we believe that it is important methodologically and that further development of our general theory and further analysis of polymorphisms for graph colouring will eventually lead to a proof of NP-hardness for any constant number of colours.

We prove that the approximate graph colouring problem in result (C) has less symmetry (in the sense of bipartite minor conditions) than approximate hypergraph colouring, which is known to be NP-hard. Then our theory implies the required reduction. Our theory also allows one to rule out the existence of certain reductions—for example, we can explain exactly how k vs. $c = 2k-1$ colouring differs from the cases $c = 2k-2$ and $c = 2k$, and hence why we are able to improve the result from $c = 2k-2$ [26] to $c = 2k-1$ and why moving to larger c requires further analysis of polymorphisms and bipartite minor conditions.

On the tractability side, practically all conditions describing tractable cases of CSP (as well as conditions describing the power of specific important algorithms) have the form of minor identities. It follows from our results that efficient algorithms for PCSPs should be based on such conditions, too. We provide such characterisation of the power of three algorithms for PCSPs. The first two of these algorithms bring a tractability result from a CSP to PCSPs. The second and the third algorithm are related to recent tractability results from Reference [30]. While the algorithmic novelty of these results is limited, they demonstrate how the applicability of certain algorithms is characterised by minor conditions satisfied by polymorphisms.

Interestingly, all known tractability results for finite-domain PCSPs (such as those above or those from References [29, 30]) follow the same scheme: The PCSP is shown to be a subproblem (inclusion-wise) of a tractable CSP, possibly with an infinite domain. We show that using infinite domain CSPs in this scheme can be necessary—namely, a simple specific PCSP, that is known to be a subproblem of several tractable infinite-domain CSPs [29, 30], is *not* a subproblem of any tractable finite-domain CSP (see Theorem 8.1).

Subsequent work. Three new recent results are based on the general theory presented in this article. A dichotomy for symmetric Boolean PCSPs (generalising a classification from Reference [29]) was proved in Reference [49]. It was shown in Reference [68] that, for any fixed 3-colourable non-bipartite graph **H**, it is NP-hard to find a 3-colouring for a given graph that admits a homomorphism to **H**. Wrochna and Živný [91] generalised this result of Reference [68] to NP-hardness of finding a **G**-colouring an **H**-colourable graphs for several classes of **G** including "cyclic cliques" and "square-free graphs," these results can be seen as hardness of colouring such graphs with $4 - \varepsilon$ colours (i.e., they fall just short of $\mathbf{G} = \mathbf{K}_4$). They also greatly improved the result of Huang [59], providing NP-hardness of colouring a k-colourable graph with $\binom{k}{\lfloor k/2 \rfloor} - 1$ colours for any $k \geq 4$, this also improves our result for any $k > 5$. The papers [68, 91] and their merged and extended version [69] also suggest further directions that the general theory of PCSPs can take: They uncover a surprising connection to algebraic topology and describe a new abstract reduction that generalises reductions given by the theory of the present article. A new conditional hardness of approximate graph colouring was given in Reference [55]; they showed that c-colouring k-colourable graphs is NP-hard assuming the (standard) d-to-1 conjecture is true for some $d \geq 2$.

Further discussion. Let us now discuss how the complexity classification quest for PCSPs compares with that for CSPs. As we said above, the gist of the algebraic approach is that lack or presence of high-dimensional symmetries determines the complexity. For (finite-domain) CSPs, there is a sharp algebraic dichotomy: Having only trivial symmetries (i.e., satisfying only those systems of minor identities that are satisfied in polymorphisms of every CSP) leads to NP-hardness, while any non-trivial symmetry implies rather strong symmetry and thus leads to tractability. Moreover, the algorithms for tractable cases are (rather involved) combinations of only two basic algorithms—one is based on local propagation [13] and the other can be seen as a very general form of Gaussian elimination [60]. It is already clear that the situation is more complicated for PCSPs: There are hard PCSPs with non-trivial (but limited in some sense) symmetries, and tractable cases are more varied [7, 29, 30, 47]. This calls for more advanced methods, and we hope that our article will provide the basis for such methods. There is an obvious question whether PCSPs exhibit a dichotomy as CSPs do, but there is not enough evidence yet to conjecture an answer. More specifically, it is not clear whether there is any PCSP whose polymorphisms are not limited enough (in terms of satisfying systems of minor identities) to give NP-hardness, but also not strong enough to ensure tractability. Classifications for special cases such as Boolean PCSPs and graph homomorphisms would help to obtain more intuition about the general complexity landscape of PCSPs, but these special cases are currently open. A more detailed discussion of possible directions of further research can be found in Section 11.

Organisation of this article. The following section contains formal definitions of the concepts introduced in the introduction as well as several concrete examples of PCSPs.

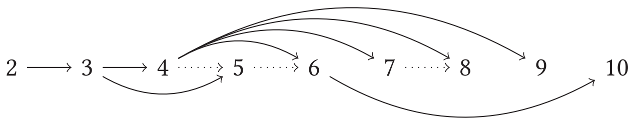

Fig. 1. Graph of dependencies of sections. Dotted edges express non-essential references between sections.

Sections 3 and 4 are the core of our general theory. Section 3 introduces reductions via algebraic conditions and proves the main results of our general theory. In Section 4, we describe several relational constructions behind the reduction from Section 3 and provide many equivalent ways to characterise the applicability of our main reduction.

Sections 5 and 6 are focused on the hardness results. In particular, Section 5 contains algebraic sufficient conditions leading to hardness by reduction from several versions of Label Cover and also a sketch of a simplified proof of NP-hardness of approximate hypergraph colouring. Section 6 contains our result on NP-hardness of approximate (k vs. $2k-1$) graph colouring.

Section 7 characterises the power of three polynomial-time algorithms for PCSPs by means of minor conditions.

In Section 8, we use a specific tractable PCSP to show that the only currently known approach for proving tractability of a PCSP—by a natural reduction to a tractable CSP—must in general involve CSPs with an infinite domain.

Section 9 describes algebraic counterparts of the constructions introduced in Section 4 and gives additional information that can be useful for further developing our general algebraic theory.

Section 10 further discusses the approximate graph colouring problem and provides technical results about distinguishing various special cases of this problem by means of minor conditions.

Finally, Section 11 summarises the results of the article and discusses possible directions of further research. For the convenience of the reader, we include a graph of dependencies of sections in Figure 1.

2 PRELIMINARIES

This section contains formal definitions of the notions introduced above. For comparison, the algebraic theory behind fixed-template CSPs can be found in a recent survey [15].

2.1 CSPs and PCSPs

We use the notation $[n] = \{1, \dots, n\}$ throughout the article.

Definition 2.1. A *constraint language* Γ on a set A is finite set of relations on A, possibly of different arity. Then A is called the *domain* of Γ.

To work with several constraint languages (possibly on different domains), it is often convenient to fix an indexing of the relations in Γ. This is formalised as follows:

Definition 2.2. A *(relational) structure* is a tuple $\mathbf{A} = (A; R_1^{\mathbf{A}}, \dots, R_l^{\mathbf{A}})$ where each $R_i^{\mathbf{A}} \subseteq A^{\mathrm{ar}(R_i)}$ is a relation on A of a well-defined arity $\mathrm{ar}(R_i) \geq 1$. We say that $\mathbf{A}$ is finite if A is finite. We will assume that all structures in this article are finite unless specified otherwise.

Two structures $\mathbf{A} = (A; R_1^{\mathbf{A}}, \dots, R_l^{\mathbf{A}})$ and $\mathbf{B} = (B; R_1^{\mathbf{B}}, \dots, R_l^{\mathbf{B}})$ are called *similar* if they have the same number of relations and $\mathrm{ar}(R_i^{\mathbf{A}}) = \mathrm{ar}(R_i^{\mathbf{B}})$ for each $i \in [l]$.

For example, a (directed) graph is relational structure with one binary relation. Any two graphs are similar structures.

We often use a single letter instead of R_i to denote a relation of a structure, e.g., $S^{\mathbf{A}}$ would denote a relation of $\mathbf{A}$, the corresponding relation in a similar structure $\mathbf{B}$, would be denoted by $S^{\mathbf{B}}$. Also, throughout the article, we denote the domains of structures $\mathbf{A}$, $\mathbf{B}$, $\mathbf{K}_n$ and so on by A, B, K_n, and so on, respectively.

Definition 2.3. For two similar relational structures $\mathbf{A}$ and $\mathbf{B}$, a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a map $h : A \to B$ such that, for each i,

$$\text{if } (a_1, \ldots, a_{\mathrm{ar}(R_i)}) \in R_i^{\mathbf{A}} \text{ then } (h(a_1), \ldots, h(a_{\mathrm{ar}(R_i)})) \in R_i^{\mathbf{B}}.$$

We write $h : \mathbf{A} \to \mathbf{B}$ to denote this, and simply $\mathbf{A} \to \mathbf{B}$ to denote that a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ exists. In the latter case, we also say that $\mathbf{A}$ maps homomorphically to $\mathbf{B}$.

Definition 2.4. For a fixed structure $\mathbf{B}$, CSP($\mathbf{B}$) is the problem of deciding whether a given input structure $\mathbf{I}$, similar to $\mathbf{B}$, admits a homomorphism to $\mathbf{B}$. In this case $\mathbf{B}$ is called the *template* for CSP($\mathbf{B}$).

For example, when $\mathbf{I}$ and $\mathbf{B}$ are graphs and $\mathbf{B} = \mathbf{K}_k$ is a k-clique, a homomorphism from $\mathbf{I}$ to $\mathbf{B}$ is simply a k-colouring of $\mathbf{I}$. Then CSP($\mathbf{B}$) is the standard k-colouring problem for graphs.

To see how the above definition corresponds to the definition of CSP with variables and constraints, view the domain of the structure $\mathbf{I}$ as consisting of variables, relations in $\mathbf{I}$ specifying which tuples of variables the constraints should be applied to, and (the corresponding) relations in $\mathbf{B}$ as sets of allowed tuples of values.

Definition 2.5. A *PCSP template* is a pair of similar structures $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{A} \to \mathbf{B}$. The problem PCSP($\mathbf{A}, \mathbf{B}$) is, given an input structure $\mathbf{I}$ similar to $\mathbf{A}$ and $\mathbf{B}$, output YES if $\mathbf{I} \to \mathbf{A}$, and NO if $\mathbf{I} \not\to \mathbf{B}$.

Note that PCSP($\mathbf{A}, \mathbf{A}$) is simply CSP($\mathbf{A}$). The promise in the PCSP is that it is never the case that $\mathbf{I} \not\to \mathbf{A}$ and $\mathbf{I} \to \mathbf{B}$. Note also that the assumption $\mathbf{A} \to \mathbf{B}$ is necessary for the problem to make sense—otherwise, the YES- and NO-cases would not be disjoint. We define PCSP($\mathbf{A}, \mathbf{B}$) as a decision problem, but it can also be defined as a search problem:

Definition 2.6. Given two relational structures $\mathbf{A}$, $\mathbf{B}$ as above, the search version of PCSP($\mathbf{A}, \mathbf{B}$) is, given an input structure $\mathbf{I}$ that maps homomorphically to $\mathbf{A}$, *find* a homomorphism $h : \mathbf{I} \to \mathbf{B}$.

There is an obvious reduction from the decision variant of every PCSP to its search variant. Nevertheless, it is open whether these two problems are equivalent for all PCSP templates. Note that, for problems CSP($\mathbf{A}$), these two versions are always equivalent [34]. We would also like to note that all of our results are formulated and proved for the decision version. However, all of them can be generalised to the corresponding search versions of the problems.

Let us give some examples of the problems of the form PCSP($\mathbf{A}, \mathbf{B}$), which are proper promise problems, i.e., not of the form CSP($\mathbf{A}$). More examples (of both tractable and intractable PCSPs) can be found in References [26, 29, 30].

Example 2.7 ((2 + ε)-SAT). For a tuple $\mathbf{t} \in \{0,1\}^n$, let $\mathrm{Ham}(\mathbf{t})$ be the Hamming weight of $\mathbf{t}$. Fix an integer $k \geq 1$. Let $\neq_2$ be the relation $\{(0,1),(1,0)\}$. Let

$$\mathbf{A} = (\{0,1\}; \{\mathbf{t} \in \{0,1\}^{2k+1} \mid \mathrm{Ham}(\mathbf{t}) \geq k\}, \neq_2),$$
$$\mathbf{B} = (\{0,1\}; \{\mathbf{t} \in \{0,1\}^{2k+1} \mid \mathrm{Ham}(\mathbf{t}) \geq 1\}, \neq_2).$$

The problem PCSP($\mathbf{A}, \mathbf{B}$) is then (equivalent to) the following variant of $(2k+1)$-SAT: Given an instance of $(2k+1)$-SAT such that some assignment satisfies at least k literals in each clause, find a normal satisfying assignment. This problem, called $(2+\varepsilon)$-SAT, was proved to be NP-hard in Reference [7].

Example 2.8 (1-in-3- vs. Not-All-Equal-Sat). Let

$$\mathbf{T} = (\{0,1\}; \{(1,0,0),(0,1,0),(0,0,1)\}),$$
$$\mathbf{H}_2 = (\{0,1\}; \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}).$$

Even though CSP($\mathbf{T}$) and CSP($\mathbf{H}_2$) are (well-known) NP-hard problems, the problem PCSP($\mathbf{T}, \mathbf{H}_2$) was shown to be in P in References [29, 30], along with a range of similar problems. All these results are obtained by using the same general scheme—a natural reduction to a tractable CSP, possibly with an infinite domain. We give several general results related to this scheme in Section 7 and then show in Section 8 that any proof of tractability of PCSP($\mathbf{T}, \mathbf{H}_2$) via this scheme must use a CSP with an infinite domain.

Example 2.9 (Approximate Graph Colouring). For $k \geq 2$, let $\neq_k$ denote the relation $\{(a,b) \in \{0,\ldots,k-1\}^2 \mid a \neq b\}$. For $k \leq c$, let

$$\mathbf{K}_k = (\{0,\ldots,k-1\}; \neq_k),$$
$$\mathbf{K}_c = (\{0,\ldots,c-1\}; \neq_c).$$

Then PCSP($\mathbf{K}_k, \mathbf{K}_c$) is the well-known *approximate graph colouring* problem: Given a k-colourable graph, find a c-colouring. The decision version of this asks to distinguish between k-colourable graphs and those that are not even c-colourable. The complexity of this problem has been studied since 1976 [50]—the problem has been conjectured to be NP-hard for any fixed $3 \leq k \leq c$, but this is still open in many cases (see References [26, 53, 59, 64, 65]) if we assume only P $\neq$ NP. For $k = 3$, the case $c = 4$ was shown to be NP-hard [53, 64], but even the case $c = 5$ was still open, and we settle it in this article (see Section 6 for more details). It was shown (using polymorphisms) in Reference [26] that, for any $k \geq 3$, it is NP-hard to distinguish k-colourable graphs from those that are not $(2k-2)$-colourable. This gives the best-known NP-hardness results for small enough k, but we further improve this result in this article. For large enough k, the best-known NP-hardness result is for k vs. $2^{\Omega(k^{1/3})}$ colouring [59]. By additionally assuming somewhat non-standard variants of the Unique Games Conjecture (with perfect completeness), NP-hardness of all approximate graph colouring problems (with $k \geq 3$) was proved in Reference [46].

Example 2.10 (Approximate Graph Homomorphism). The standard graph k-colouring problem has a natural generalisation to graph homomorphisms. For a fixed (undirected) graph $\mathbf{H}$, the problem CSP($\mathbf{H}$) is, given a graph $\mathbf{I}$ (which can be assumed to be undirected), decide whether $\mathbf{I} \to \mathbf{H}$. This problem is often called $\mathbf{H}$-colouring. It has been extensively studied, see, e.g., References [57, 58, 73]. A well-known result by Hell and Nešetřil [57] states that $\mathbf{H}$-colouring is solvable in polynomial time if $\mathbf{H}$ is bipartite or has a loop, and it is NP-complete otherwise. A natural generalisation to the promise setting, i.e., problems PCSP($\mathbf{G}, \mathbf{H}$), was suggested in Reference [30]. It was conjectured there that the problem is NP-hard for any non-bipartite loopless undirected $\mathbf{G}, \mathbf{H}$ with $\mathbf{G} \to \mathbf{H}$. Even the case $\mathbf{G} = \mathbf{C}_5$ (5-cycle), $\mathbf{H} = \mathbf{K}_3$ was mentioned in Reference [30] as an intriguing open problem, and we settle it in this article. See Section 6 for more details.

Example 2.11 (Approximate Hypergraph Colouring). This problem is similar to Example 2.9, but uses the "not-all-equal" relation $\mathrm{NAE}_k = \{0,\ldots,k-1\}^3 \setminus \{(a,a,a) \mid a \in \{0,\ldots,k-1\}\}$ instead of $\neq_k$, and similarly for c, i.e., we are talking about PCSP($\mathbf{H}_k, \mathbf{H}_c$) where

$$\mathbf{H}_k = (\{0,\ldots,k-1\}; \mathrm{NAE}_k),$$
$$\mathbf{H}_c = (\{0,\ldots,c-1\}; \mathrm{NAE}_c).$$

A colouring of a hypergraph is an assignment of colours to its vertices that leaves no hyperedge monochromatic. Thus, in (the search variant of) this problem one needs to find a c-colouring for a given k-colourable 3-uniform hypergraph. This problem has been proved to be NP-hard for any fixed $2 \leq k \leq c$ [47].

Example 2.12 (Strong vs. Normal Hypergraph Colouring). Let $k, c \geq 2$, and let

$$\mathbf{A} = (\{0,\dots,k\}; \{(a_1,\dots,a_k) \in \{0,\dots,k\}^k \mid a_i \neq a_j \text{ for all } i,j\}),$$
$$\mathbf{B} = (\{0,\dots,c-1\}; \{(a_1,\dots,a_k) \in \{0,\dots,c-1\}^k \mid a_i \neq a_j \text{ for some } i,j\}).$$

Then PCSP($\mathbf{A},\mathbf{B}$) is the problem of distinguishing k-uniform hypergraphs that admit a strong $(k+1)$-colouring (i.e., one without repetition of colours in any hyperedge) from those that do not admit a normal c-colouring. It was conjectured in Reference [26] that this problem is NP-hard for all $(k,c) \neq (2,2)$, and some special cases ($k = 3,4$ and $c = 2$) were settled in that paper, but this conjecture remains wide open.

Example 2.13 (Rainbow vs. Normal Hypergraph Colouring). Let k, q, c be positive integers with $k \geq q \geq 2$ and $c \geq 2$, and let

$$\mathbf{R}_{k,q} = (\{0,\dots,q-1\}; \{(a_1,\dots,a_k) \in \{0,\dots,q-1\}^k \mid \{a_1,\dots,a_k\} = \{0,\dots,q-1\}\}),$$
$$\mathbf{H}_{k,c} = (\{0,\dots,c-1\}; \{(a_1,\dots,a_k) \in \{0,\dots,c-1\}^k \mid a_i \neq a_j \text{ for some } i,j\}).$$

In PCSP($\mathbf{R}_{k,q},\mathbf{H}_{k,c}$), one is given a k-uniform hypergraph that has a q-colouring such that all colours appear in each hyperedge, and one needs to find a normal c-colouring. This problem is known to be in P for $k = q$ and $c = 2$; a randomised algorithm for it can be found in Reference [76], and a deterministic algorithm due to Alon (unpublished) is mentioned in Reference [29]. PCSP($\mathbf{R}_{k,q},\mathbf{H}_{k,c}$) is NP-hard if $2 \leq q \leq \lfloor k/2 \rfloor$ [54] or if $2 \leq q \leq k - 2\lfloor\sqrt{k}\rfloor$ and $c = 2$ [6]. Further variations of such PCSPs were considered in References [6, 54].

2.2 Polymorphisms

We now proceed to define polymorphisms, which are the main algebraic technical tools used in the analysis of CSPs.

Definition 2.14. The *nth power* of $\mathbf{A}$ is the structure $\mathbf{A}^n = (A^n; R_1^{\mathbf{A}^n},\dots,R_l^{\mathbf{A}^n})$ whose relations are defined as follows: For every $\mathrm{ar}(R_i) \times n$ matrix M such that all columns of M are in $R_i^{\mathbf{A}}$, consider the rows of M as elements of A^n and put this tuple of rows in $R_i^{\mathbf{A}^n}$.

Definition 2.15. Given two similar relational structures $\mathbf{A}$ and $\mathbf{B}$, an n-ary *polymorphism*[1] *from* $\mathbf{A}$ *to* $\mathbf{B}$ is a homomorphism from $\mathbf{A}^n$ to $\mathbf{B}$. To spell this out, a polymorphism is a mapping f from A^n to B such that, for each $i \leq l$ and all tuples $(a_{11},\dots,a_{\mathrm{ar}(R_i)1})$, …, $(a_{1n},\dots,a_{\mathrm{ar}(R_i)n}) \in R_i^{\mathbf{A}}$, we have

$$(f(a_{11},\dots,a_{1n}),\dots,f(a_{\mathrm{ar}(R_i)1},\dots,a_{\mathrm{ar}(R_i)n})) \in R_i^{\mathbf{B}}.$$

We denote the set of all polymorphisms from $\mathbf{A}$ to $\mathbf{B}$ by Pol($\mathbf{A},\mathbf{B}$), and we write simply Pol($\mathbf{A}$) for Pol($\mathbf{A},\mathbf{A}$).

For the case $\mathbf{A} = \mathbf{B}$, this definition coincides with the standard definition of a polymorphism of a relational structure $\mathbf{A}$ (see, e.g., Reference [15, Section 4]).

Definition 2.16. A *projection* (also known as *dictator*[2]) on a set A is an operation $p_i^{(n)} : A^n \to A$ of the form $p_i^{(n)}(x_1,\dots,x_n) = x_i$.

[1] Called *weak polymorphism* in References [7, 27].

[2] Projection is the standard name for these objects in universal algebra, while dictator is the standard name in approximability literature.

It is well-known and easy to see that every projection on A is a polymorphism of every relational structure on A. We will sometimes write simply p_i when the arity n of a projection is clear from the context.

Example 2.17. Consider the structures $\mathbf{T}$ and $\mathbf{H}_2$ from Example 2.8. It is well-known and not hard to verify that

- $\mathrm{Pol}(\mathbf{T})$ consists of all projections on $\{0,1\}$, and
- $\mathrm{Pol}(\mathbf{H}_2)$ consists of all operations of the form $\pi(p_i^{(n)})$ where π is a permutation on $\{0,1\}$ and $p_i^{(n)}$ is a projection.

However, $\mathrm{Pol}(\mathbf{T},\mathbf{H}_2)$ contains many functions that are not like projections. For example, for any $k \geq 1$, define $f_k : \{0,1\}^{3k-1} \to \{0,1\}$ so $f_k(\mathbf{t}) = 1$ if $\mathrm{Ham}(\mathbf{t}) \geq k$ and $f_k(\mathbf{t}) = 0$ otherwise. It is easy to see that $f_k \in \mathrm{Pol}(\mathbf{T},\mathbf{H}_2)$. Indeed, if M is a $3 \times (3k-1)$ matrix whose columns come from the set $\{(1,0,0),(0,1,0),(0,0,1)\}$, then some row of M contains strictly fewer than k 1's and some other row contains at least k 1's. Therefore, by applying f_k to the rows of M, one obtains a tuple in $\{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$, as required.

Example 2.18. Recall Example 2.9. It is easy to check that, for any $n \geq 1$, the n-ary functions from $\mathrm{Pol}(\mathbf{K}_k,\mathbf{K}_c)$ are simply the c-colourings of $\mathbf{K}_k^n$, the nth power of $\mathbf{K}_k$ (in the sense of Definition 2.14).

Unlike polymorphisms of a single relational structure $\mathbf{A}$, the set $\mathrm{Pol}(\mathbf{A},\mathbf{B})$ is not closed under composition—for example, if $f(x,y,z)$ and $g(x,y)$ are in $\mathrm{Pol}(\mathbf{A},\mathbf{B})$, then $f(g(x,w),y,z)$ is not necessarily there. In general, the composition is not always well-defined, and even when it is (e.g., when $\mathbf{A}$ and $\mathbf{B}$ have the same domain), $f(g(x,w),y,z)$ may not be in $\mathrm{Pol}(\mathbf{A},\mathbf{B})$. However, it is always closed under taking minors.

Definition 2.19. An n-ary function $f : A^n \to B$ is called a *minor* of an m-ary function $g : A^m \to B$ given by a map $\pi : [m] \to [n]$ if

$$f(x_1,\dots,x_n) = g(x_{\pi(1)},\dots,x_{\pi(m)})$$

for all $x_1,\dots,x_n \in A$.

Alternatively, one can say that f is a minor of g if it is obtained from g by identifying variables, permuting variables, and introducing dummy variables.

Definition 2.20. Let $\mathscr{O}(A,B) = \{f : A^n \to B \mid n \geq 1\}$. A *(function) minion*[3] $\mathscr{M}$ on a pair of sets (A,B) is a non-empty subset of $\mathscr{O}(A,B)$ that is closed under taking minors. For fixed $n \geq 1$, let $\mathscr{M}^{(n)}$ denote the set of n-ary functions from $\mathscr{M}$. Unless stated otherwise, we assume that all minions are defined on finite sets.

We remark that clones have been used extensively in the algebraic theory of CSP—a clone is a minion on (A,A) that is closed under composition and contains all projections. For any structure $\mathbf{A}$, $\mathrm{Pol}(\mathbf{A})$ is a clone. For more details, see Reference [15].

We now introduce one of the central notions of this article; it generalises the notion of *h1 clone homomorphisms* [16, Definition 5.1].

Definition 2.21. Let $\mathscr{M}$ and $\mathscr{N}$ be two minions (not necessarily on the same pairs of sets). A mapping $\xi : \mathscr{M} \to \mathscr{N}$ is called a *minion homomorphism* if

[3] What we define as a function minion was called *minor closed class* in Reference [83], and *clonoid* in Reference [1]. Albeit clonoids are usually also required to be closed under composition with a given clone from the outside.

(1) it preserves arities, i.e., $\mathrm{ar}(g) = \mathrm{ar}(\xi(g))$ for all $g \in \mathscr{M}$, and
(2) it preserves taking minors, i.e., for each $\pi : [m] \to [n]$ and each $g \in \mathscr{M}^{(m)}$, we have

$$\xi(g)(x_{\pi(1)}, \dots, x_{\pi(m)}) = \xi(g(x_{\pi(1)}, \dots, x_{\pi(m)})).$$

Item (2) above can also be interpreted as "preserving satisfaction of minor identities," i.e., if $f(x_1, \dots, x_n) = g(x_{\pi(1)}, \dots, x_{\pi(m)})$ for some $f \in \mathscr{M}^{(n)}$, $g \in \mathscr{M}^{(m)}$, and $\pi : [m] \to [n]$, then

$$\xi(f)(x_1, \dots, x_n) = \xi(g)(x_{\pi(1)}, \dots, x_{\pi(m)}).$$

Example 2.22. We will construct a minion homomorphism from $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_4)$ to the minion $\mathscr{P}_2$ of all projections on a two-element set. We note that the minion of projections on any set with at least two elements can be used instead $\mathscr{P}_2$; we use the two-element set so we can work with a concrete function minion. This minion homomorphism is built on the following combinatorial statement proved in Reference [26, Lemma 3.4]: For each $f \in \mathrm{Pol}^{(n)}(\mathbf{K}_3, \mathbf{K}_4)$, there exist $t \in K_4$ (we will call any such t a *trash colour*), a coordinate $i \in [n]$, and a map $\alpha : K_3 \to K_4$ such that

$$f(a_1, \dots, a_n) \in \{t, \alpha(a_i)\}$$

for all $a_1, \dots, a_n \in K_3$. In other words, if we erase the value t from the table of f, then $f(x_1, \dots, x_n)$, which is now a partial function, depends only on x_i. Moreover, it is shown in Reference [26, Lemma 3.9], that while the trash colour t is not necessarily unique, the coordinate i is.

We define $\xi : \mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_4) \to \mathscr{P}_2$ by mapping each f to p_i (preserving the arity) for the i that satisfies the above. To prove that such ξ is a minion homomorphism, consider f, g such that f is a minor of g, i.e.,

$$f(x_1, \dots, x_n) = g(x_{\pi(1)}, \dots, x_{\pi(m)})$$

for some $\pi : [m] \to [n]$. We claim that if t is a trash colour for g, then it is also a trash colour for f. Indeed, if $g(a_1, \dots, a_m) \in \{t, \alpha(a_i)\}$ for all $a_1, \dots, a_n \in K_3$, then we get

$$f(a_1, \dots, a_n) = g(a_{\pi(1)}, \dots, a_{\pi(m)}) \in \{t, \alpha(a_{\pi(i)})\}.$$

This also shows that the coordinate j assigned to f is $\pi(i)$, therefore $\xi(f) = p_{\pi(i)}$. We conclude that ξ preserves taking minors. That is true, since

$$p_{\pi(i)}(x_1, \dots, x_n) = x_{\pi(i)} = p_i(x_{\pi(1)}, \dots, x_{\pi(m)}).$$

2.3 Primitive Positive Formulas

Jeavons' proof [61] that polymorphisms capture the complexity of CSPs was built on a Galois correspondence between relations and operations [24, 25, 51]. A Galois correspondence appropriate for PCSP was found by Pippenger [83], and Brakensiek and Guruswami [27, Appendices D and E] used it to generalise Jeavons' result to PCSP. We build further on these results.

Primitive positive formulas (pp-formulas) have been used extensively in the theory of CSPs—see Reference [15] for many examples. We remind that, for a relational structure $\mathbf{A}$, with a relation $R^{\mathbf{A}}$, we use R to denote the abstract relational symbol associated with this relation. These abstract symbols are used in pp-formulas, which makes them interpretable in any structure that is similar to $\mathbf{A}$.

Definition 2.23. For a relational structure $\mathbf{A} = (A; R_1^{\mathbf{A}}, \dots, R_l^{\mathbf{A}})$, a primitive positive formula (pp-formula) over $\mathbf{A}$ is an existentially quantified conjunction of predicates of the form $(v_{j_1}, \dots, v_{j_{k_i}}) \in R_i$ or $v_{j_1} = v_{j_2}$ where v_i are variables and $k_i = \mathrm{ar}(R_i)$.

Note that any instance of $\mathrm{CSP}(\mathbf{A})$ can be interpreted as a conjunction of constraints, i.e., as a pp-formula Ψ without quantifiers and the equality predicate. Similarly, since an instance of $\mathrm{CSP}(\mathbf{A})$

is also an instance of CSP($\mathbf{B}$) for any two similar structures $\mathbf{A}$ and $\mathbf{B}$, and any pp-formula over $\mathbf{A}$ is also a pp-formula over $\mathbf{B}$, we can talk about pp-formulas over a PCSP template $(\mathbf{A}, \mathbf{B})$.

To ease readability, we will write $\Psi^{\mathbf{A}}$ when we interpret each relational symbol R_i appearing in Ψ as the relation $R_i^{\mathbf{A}}$. Therefore, while a pp-formula Ψ contains formal expressions such as $(x_1, x_2) \in R_i$, $\Psi^{\mathbf{A}}$ contains expressions of the form $(x_1, x_2) \in R_i^{\mathbf{A}}$, and therefore satisfying assignments of $\Psi^{\mathbf{A}}$ form a subset of A^k where k is the number of free variables of Ψ.

Example 2.24. The following is an example of a pp-formula over any digraph. This formula expresses "u and v are connected by a directed walk of length 2."

$$\Psi(u, v) = \exists w (u, w) \in E \wedge (w, v) \in E$$

It defines a binary relation $\Psi^{\mathbf{G}}$ on vertices of a digraph $\mathbf{G}$. For example, the relation $\Psi^{\mathbf{K}_3}$ is the full binary relation on K_3, i.e., $\Psi^{\mathbf{K}_3} = K_3^2$.

Definition 2.25. Let $(\mathbf{A}, \mathbf{B})$ be a PCSP template. A template $(\mathbf{A}', \mathbf{B}')$ such that $A' = A$ and $B' = B$ is said to be

(1) *primitive positive definable* (*pp-definable*) in $(\mathbf{A}, \mathbf{B})$ if for each relational symbol R of $(\mathbf{A}', \mathbf{B}')$ there is a pp-formula Ψ_R over $(\mathbf{A}, \mathbf{B})$ such that

$$R^{\mathbf{A}'} = \{(a_1, \dots, a_{\mathrm{ar}(R)}) \in A^{\mathrm{ar}(R)} \mid \Psi_R^{\mathbf{A}}(a_1, \dots, a_{\mathrm{ar}(R)})\}, \text{ and}$$
$$R^{\mathbf{B}'} = \{(b_1, \dots, b_{\mathrm{ar}(R)}) \in B^{\mathrm{ar}(R)} \mid \Psi_R^{\mathbf{B}}(b_1, \dots, b_{\mathrm{ar}(R)})\};$$

(2) a *simple relaxation* of $(\mathbf{A}, \mathbf{B})$ if $\mathbf{A}, \mathbf{A}', \mathbf{B}, \mathbf{B}'$ are all similar, and for each relational symbol R, we have $R^{\mathbf{A}'} \subseteq R^{\mathbf{A}}$ and $R^{\mathbf{B}} \subseteq R^{\mathbf{B}'}$.

We say that a template is ppp-definable in $(\mathbf{A}, \mathbf{B})$, if it is it is obtained from $(\mathbf{A}, \mathbf{B})$ by a sequence of pp-definitions and simple relaxations.

We remark that ppp-definability is closely related to polymorphisms and the reduction between PCSPs described in Reference [27].

Theorem 2.26 [27, 83]. *The following are equivalent for a pair of PCSP templates* $(\mathbf{A}, \mathbf{B})$ *and* $(\mathbf{A}', \mathbf{B}')$ *over the same pair of finite domains:*

(1) $(\mathbf{A}', \mathbf{B}')$ *is ppp-definable in* $(\mathbf{A}, \mathbf{B})$.
(2) $(\mathbf{A}', \mathbf{B}')$ *is a simple relaxation of a template pp-definable in* $(\mathbf{A}, \mathbf{B})$.
(3) $\mathrm{Pol}(\mathbf{A}, \mathbf{B}) \subseteq \mathrm{Pol}(\mathbf{A}', \mathbf{B}')$.

Moreover, if these conditions hold, then PCSP$(\mathbf{A}', \mathbf{B}')$ *is polynomial-time reducible to PCSP*$(\mathbf{A}, \mathbf{B})$.

3 ALGEBRAIC REDUCTIONS

In this section, we prove the two main results of our general theory. The first one can be formulated right away.

Theorem 3.1. *Let* $(\mathbf{A}_1, \mathbf{B}_1)$ *and* $(\mathbf{A}_2, \mathbf{B}_2)$ *be two finite PCSP templates, and let* $\mathscr{M}_i = \mathrm{Pol}(\mathbf{A}_i, \mathbf{B}_i)$ *for* $i = 1, 2$. *If there is a minion homomorphism* $\xi : \mathscr{M}_1 \to \mathscr{M}_2$, *then PCSP*$(\mathbf{A}_2, \mathbf{B}_2)$ *is log-space reducible to PCSP*$(\mathbf{A}_1, \mathbf{B}_1)$.

The proof of this theorem is provided by the second of our results, but to formulate it, we need to first address a few formalities about the problem of deciding minor conditions. We remark that we characterise the existence of such a minion homomorphism in many equivalent ways (see Theorem 4.12 and Corollary 9.5).

3.1 Deciding Satisfiability of Bipartite Minor Conditions

The study of Maltsev conditions[4] [77, 89] has a long history in universal algebra. We will use only a special case of (strong) Maltsev conditions called minor conditions and their restricted "bipartite" form.

Definition 3.2. A *bipartite minor condition* is a finite set Σ of minor identities where the sets of function symbols used on the right- and left-hand sides are disjoint. More precisely, we say that a minor condition Σ is *bipartite* over two disjoint sets of function symbols $\mathcal{U}$ and $\mathcal{V}$ if it contains only identities of the form

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

where $f \in \mathcal{U}$ and $g \in \mathcal{V}$ are symbols of arity n and m, respectively, $x_1, \dots, x_n$ are variables, and $\pi : [m] \to [n]$.

Such a condition is said to be *satisfied* in a minion $\mathscr{M}$ on (A, B) if there is an assignment $\zeta : \mathcal{U} \cup \mathcal{V} \to \mathscr{M}$ that assigns to each function symbol a function from $\mathscr{M}$ of the corresponding arity so

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

for each identity $f(x_1, \dots, x_n) \approx g(x_{\pi(1)}, \dots, x_{\pi(m)})$ in Σ and all $a_1, \dots, a_n \in A$. We say that a minor condition is *trivial* if it is satisfied in every minion, in particular, in the minion $\mathscr{P}_A$ consisting of all projections (dictators) on a set A that contains at least two elements.

We note that as long as a bipartite minor condition is satisfied in some $\mathscr{P}_A$ with $|A| \geq 2$, it is satisfied in every minion: Recall that by definition every minion $\mathscr{M}$ is non-empty, and therefore it contains a unary function f (obtained by identifying all variables in a function from $\mathscr{M}$). Consequently, $\mathscr{M}$ contains functions defined by $(x_1, \dots, x_n) \mapsto f(x_i)$ for each i. These functions then satisfy all identities satisfied by projections in $\mathscr{P}_A$. In fact, if f is not constant, then they satisfy exactly the same identities as projections.

The symbols f, g, and so on, in a minor condition are abstract function symbols. Nevertheless, we sometimes (in particular, when working with specific simple minor conditions) use the same symbols to denote concrete functions that satisfy this condition. When we want to stress the assignment of concrete functions to symbols, we use $\zeta(f)$ for the concrete function assigned to the abstract symbol f.

Example 3.3. Consider the following bipartite minor condition. We set $\mathcal{U}$ to contain a single binary symbol f, and $\mathcal{V}$ a single quaternary symbol g. The set Σ then consists of identities:

$$\begin{aligned} f(x, y) &\approx g(y, x, x, x), \\ f(x, y) &\approx g(x, y, x, x), \\ f(x, y) &\approx g(x, x, y, x), \\ f(x, y) &\approx g(x, x, x, y). \end{aligned}$$

This condition is not trivial, since if $f = p_1$ and $g = p_i$ for some i, then the ith identity is not satisfied, and if $f = p_2$, then the first identity forces that $g = p_1$, which contradicts any of the other identities.

The above bipartite minor condition is satisfied in the minion $\mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_k)$ (recall Example 2.11) for each $k \geq 4$. We define a function $\zeta(g)$ by the following:

$$\zeta(g)(x, y, z, u) = \begin{cases} a & \text{if at least 3 arguments are equal to } a; \\ x + 2 & \text{otherwise.} \end{cases}$$

[4] Not to be confused with Maltsev identities $f(x, y, y) \approx f(y, y, x) \approx x$, which are a specific case of a Maltsev condition.

The function $\zeta(f)$ is defined by $\zeta(f)(x,y) = x$. Clearly, $\zeta(f)$ and $\zeta(g)$ satisfy the required identities, also $\zeta(f)$ is in $\mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_k)$. We now show that also $\zeta(g)$ is. Consider a 3×4 matrix $M = (a_{ij})$ such that each column of M is a triple in NAE_2. We need to show that applying $\zeta(g)$ to the rows of M gives a triple in NAE_k. For contradiction, assume that we get a triple of the form (a,a,a). If a is 0 or 1, then in each row of M at least three entries are equal to a. In this case, one of the columns of M is $(a,a,a) \notin \mathrm{NAE}_2$, a contradiction. Otherwise, a is 2 or 3, which implies that the first column of M is $(a-2, a-2, a-2)$, a contradiction again.

Example 3.4. The condition Σ from the previous example is also satisfied in the minion $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_5)$ (recall Example 2.9). To see this, define $\zeta(f)(x,y) = x$ again and define $\zeta(g)$ as follows:

$$\zeta(g)(x,y,z,u) = \begin{cases} a & \text{if at least three arguments are equal to } a \text{, else} \\ 0 & \text{if } x = 0 \text{ and at least one of } y,z,u \text{ is } 0 \text{, else} \\ 1 & \text{if } x = 0 \text{ and at least two of } y,z,u \text{ are } 1 \text{, else} \\ 2 & \text{if } x = 0 \text{ and at least two of } y,z,u \text{ are } 2 \text{, else} \\ x+2 & \text{otherwise.} \end{cases}$$

It is straightforward to check that $\zeta(g)$ is a polymorphism and that $\zeta(f)$ and $\zeta(g)$ satisfy Σ.

Example 3.5. We now present a simple minor condition that is *not* satisfied in $\mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_k)$ for any $k \geq 2$:

$$\begin{aligned} f(x,y) &\approx g(x,x,y,y,y,x), \\ f(x,y) &\approx g(x,y,x,y,x,y), \\ f(x,y) &\approx g(y,x,x,x,y,y). \end{aligned}$$

Note that the columns of x's and y's on the right above correspond to the triples in NAE_2. Now assume that this condition is satisfied by some functions $\zeta(f), \zeta(g)$ in $\mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_k)$, so the identities above become equalities. Then substitute 0 for x and 1 for y in these equalities. The triple (column) on the right-hand side of the system is obtained by applying $\zeta(g)$ to the six triples in NAE_2, so it must be in NAE_k. However, this triple is equal to (b,b,b) where $b = \zeta(f)(0,1)$, which is not in NAE_k.

In Section 10, we prove that, for any $k \geq 3$, this minor condition is not satisfied in $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k-1})$ either—this is the key part in our proof that $\mathrm{PCSP}(\mathbf{K}_k, \mathbf{K}_{2k-1})$ is NP-hard.

Remark 3.6. An identity of *height 1* is an expression of the form

$$f(x_{\pi(1)}, \dots, x_{\pi(n)}) \approx g(x_{\sigma(1)}, \dots, x_{\sigma(m)}),$$

where f, g are function symbols, $\pi : [n] \to [k]$ and $\sigma : [m] \to [k]$. Systems of such identities were considered in Reference [16] (see also Reference [15]). Clearly, any minor identity has height 1. Moreover, any height 1 condition, i.e., a finite set of height 1 identities, can be turned into a (bipartite) minor condition by replacing each height 1 identity by two minor identities, e.g., the identity above would be replaced by

$$\begin{aligned} e(x_1, \dots, x_k) &\approx f(x_{\pi(1)}, \dots, x_{\pi(n)}), \\ e(x_1, \dots, x_k) &\approx g(x_{\sigma(1)}, \dots, x_{\sigma(m)}), \end{aligned}$$

where e is a newly introduced symbol. It is obvious that any minion $\mathscr{M}$ satisfies the obtained bipartite minor condition if and only if it satisfies the original height 1 condition.

Definition 3.7. We define the problem $\mathrm{MC}(N)$ (*triviality of a bipartite minor condition*) as the problem where the input is a triple $(\Sigma, \mathcal{U}, \mathcal{V})$, with Σ a bipartite minor condition over $\mathcal{U}$

and $\mathcal{V}$ that involves function symbols of maximal arity N, and the goal is to decide whether the condition Σ is trivial.

Deciding triviality of bipartite minor conditions is essentially just a different interpretation of the Label Cover problem that was introduced in Reference [2]. To compare these two problems, we use a formulation of Label Cover that is closer to the one that appeared in, e.g., References [7] and [26]. In addition, we bound the size of the label sets by a constant N. Some bounded version often appears in the literature as it is well-known that if $N \geq 3$, then it is a NP-complete problem (see, e.g., Reference [26, Lemma 4.2]).

Definition 3.8 (Label cover). Fix a positive integer N. We define $\mathrm{LC}(N)$ as the following decision problem: The input is a tuple (U, V, E, l, r, Π) where

- $G = (U, V; E)$ is a bipartite graph,
- $l, r \leq N$ are positive integers, and
- Π is a family of maps $\pi_e : [r] \to [l]$, one for each $e \in E$.

The goal is to decide whether there is a labelling of vertices from U and V with labels from $[r]$ and $[l]$, respectively, such that if $(u, v) \in E$, then the label of v is mapped by $\pi_{(u,v)}$ to the label of u.

Remark 3.9. Note that $\mathrm{LC}(N)$ is a CSP where vertices are variables and edges correspond to constraints: a constraint corresponding to an edge $e \in E$ is given by π_e.

We interpret a label cover instance (U, V, E, l, r, Π) with $l, r \leq N$ as a bipartite minor condition Σ, an input to $\mathrm{MC}(N)$, as follows:

- Each vertex $u \in U$ is interpreted as an l-ary function symbol f_u, and each vertex $v \in V$ as an r-ary function symbol g_v.
- For each edge $e = (u, v)$, we add to Σ the identity
$$f_u(x_1, \dots, x_l) \approx g_v(x_{\pi_e(1)}, \dots, x_{\pi_e(r)}). \tag{$\heartsuit$}$$

Observe that Σ is indeed a bipartite minor condition over $\mathcal{U} = \{f_u \mid u \in U\}$ and $\mathcal{V} = \{g_v \mid v \in V\}$. We claim that the minor condition obtained in this way is trivial if and only if the original Label Cover instance has a solution. The main difference is that a solution to Label Cover is a labelling, while a solution (a witness) to triviality of minor conditions is an assignment of projections to the function symbols. Nevertheless, there is a clear bijection between the labels and the projections: label i corresponds to projection p_i. Clearly, a constraint $((u, v), \pi)$ of the Label Cover is satisfied by a pair of labels (i, j) if and only if assigning p_i and p_j to f_u and g_v, respectively, satisfies ($\heartsuit$).

The *long code* is an error-correcting code that can be defined as the longest code over the Boolean alphabet that does not repeat bits. Precisely, it encodes a value $i \in [n]$ as the string of bits of length 2^n corresponding to the table of the function $p_i \in \mathscr{P}_2$.

Therefore, it is possible to see the problem MC as just a conjunction of Label Cover with the Long code. This conjunction has often been used before, e.g., References [18, 56]. Our insight is that satisfaction of a constraint can be extended to functions that are not projections (words that are not code words of the Long code), we can simply say that the constraint corresponding to the edge (u, v) is satisfied if f_u and g_v satisfy ($\heartsuit$). This approach circumvents some combinatorial difficulties of using Label Cover and the Long code, and is essential for our reduction to work.

The second (and main) advantage of using identities instead of Label Cover is that it allows us to define the following promise version. We will use this promise problem as an intermediate problem for our reduction, and it is interesting in its own right.

Definition 3.10. Fix a minion $\mathscr{M}$ and a positive integer N. We define $\mathrm{PMC}_{\mathscr{M}}(N)$ (*promise satisfaction of a bipartite minor condition*) as the promise problem in which, given a bipartite minor condition Σ that involves symbols of arity at most N, one needs to output YES if Σ is trivial and NO if Σ is not satisfiable in $\mathscr{M}$.

The promise in the above problem is that it is never the case that Σ is non-trivial, but satisfied in $\mathscr{M}$.

Remark 3.11. Let $\mathscr{M}_1, \mathscr{M}_2$ be two minions such that there is a minion homomorphism $\xi : \mathscr{M}_1 \to \mathscr{M}_2$. Then, for any N, $\mathrm{PMC}_{\mathscr{M}_2}(N)$ is obtained from $\mathrm{PMC}_{\mathscr{M}_1}(N)$ simply by strengthening the promise. To see this, observe that if some Σ is not satisfied in $\mathscr{M}_2$ then it cannot be satisfied in $\mathscr{M}_1$. Indeed, suppose the contrary, say that some f_i's and g_j's from $\mathscr{M}_1$ satisfy Σ. Since ξ is a minion homomorphism from $\mathscr{M}_1$ to $\mathscr{M}_2$, it follows that $\xi(f_i)$'s and $\xi(g_j)$'s satisfy Σ in $\mathscr{M}_2$.

We can finally formulate our second main result.

THEOREM 3.12. *Let $(\mathbf{A}, \mathbf{B})$ be a template and let $\mathscr{M}$ denote its polymorphism minion.*

(1) If N is an upper bound on the size of any relation $R^{\mathbf{A}}$ of $\mathbf{A}$ as well as on $|A|$, then PCSP$(\mathbf{A}, \mathbf{B})$ can be reduced to $\mathrm{PMC}_{\mathscr{M}}(N)$ in log-space.
(2) For each $N > 0$, $\mathrm{PMC}_{\mathscr{M}}(N)$ can be reduced to PCSP$(\mathbf{A}, \mathbf{B})$ in log-space.

Before we get to the proof, let us comment on how Theorem 3.1 follows from this result.

PROOF OF THEOREM 3.1 GIVEN THEOREM 3.12. We recall that we have two PCSP templates $(\mathbf{A}_1, \mathbf{B}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2)$, and the corresponding polymorphism minions $\mathscr{M}_i = \mathrm{Pol}(\mathbf{A}_i, \mathbf{B}_i)$, $i = 1, 2$. Our goal is to find a log-space reduction from $\mathrm{PCSP}(\mathbf{A}_2, \mathbf{B}_2)$ to $\mathrm{PCSP}(\mathbf{A}_1, \mathbf{B}_1)$ given that there is a minion homomorphism $\xi : \mathscr{M}_1 \to \mathscr{M}_2$.

By Remark 3.11, we have that, for any N, $\mathrm{PMC}_{\mathscr{M}_2}(N)$ is obtained from $\mathrm{PMC}_{\mathscr{M}_1}(N)$ by strengthening the promise. Clearly, this gives a (trivial) log-space reduction from $\mathrm{PMC}_{\mathscr{M}_2}(N)$ to $\mathrm{PMC}_{\mathscr{M}_1}(N)$.

To conclude the proof, we connect this reduction with the two reductions from Theorem 3.12. Starting with $\mathrm{PCSP}(\mathbf{A}_2, \mathbf{B}_2)$, we reduce it to $\mathrm{PMC}_{\mathscr{M}_2}(N)$ where N is given by the first item of Theorem 3.12. The above paragraph then gives us a reduction to $\mathrm{PMC}_{\mathscr{M}_1}(N)$. Finally, the second item of Theorem 3.12 provides a reduction to $\mathrm{PCSP}(\mathbf{A}_1, \mathbf{B}_1)$. □

The proof of Theorem 3.12 is given in the following two subsections.

3.2 From PCSP to Minor Conditions

We now prove Theorem 3.12(1). For that, we need to provide a reduction from PCSP to PMC for a given PCSP template and its polymorphism minion. This reduction follows a standard way of proving hardness of Label Cover [2]. It is built on a two-prover protocol introduced by Reference [19]. Our presentation of this reduction is a generalisation of Reference [26, Lemma 4.2].

Even though we start with a PCSP with template $(\mathbf{A}, \mathbf{B})$, we only use the structure $\mathbf{A}$ for the construction of a bipartite minor condition from a given instance $\mathbf{I}$ of PCSP$(\mathbf{A}, \mathbf{B})$. The structure $\mathbf{B}$ will influence soundness of the reduction. Fix an enumeration $A = \{a_1, \dots, a_n\}$ of the domain of $\mathbf{A}$, and consider an instance $\mathbf{I}$, i.e., a structure similar to $\mathbf{A}$. We construct a bipartite minor condition $\Sigma = \Sigma(\mathbf{A}, \mathbf{I})$ over $\mathcal{U}$ and $\mathcal{V}$ in the following way:

(1) Define $\mathcal{U}$ to be the set of symbols f_v for $v \in I$, each of arity $n = |A|$.
(2) For each relation R do the following: Let $k = \mathrm{ar}(R)$, $m = |R^{\mathbf{A}}|$, and let $\{(a_{\pi_1(1)}, \dots, a_{\pi_k(1)}), \dots, (a_{\pi_1(m)}, \dots, a_{\pi_k(m)})\}$ be the list of all tuples from $R^{\mathbf{A}}$; for each constraint $C = ((v_1, \dots, v_k), R)$, i.e., each tuple $(v_1, \dots, v_k) \in R^{\mathbf{I}}$, introduce into $\mathcal{V}$ a new symbol g_C of

arity m and add to Σ the following identities:

$$f_{v_1}(x_1, \dots, x_n) \approx g_C(x_{\pi_1(1)}, \dots, x_{\pi_1(m)})$$
$$\vdots$$
$$f_{v_k}(x_1, \dots, x_n) \approx g_C(x_{\pi_k(1)}, \dots, x_{\pi_k(m)}).$$

This assigns, to each $\mathbf{I}$ and $\mathbf{A}$, an instance $(\Sigma, \mathcal{U}, \mathcal{V})$ of PMC(N). The bound N is the larger of $|A|$ and the maximum of $|R^{\mathbf{A}}|$, for all relations of $\mathbf{A}$. Clearly, if $\mathbf{A}$ is fixed, then the condition Σ is computable from $\mathbf{I}$ in log-space.

Example 3.13. We show a reduction from NAE-Sat to MC(6). NAE-Sat is the same as CSP$(\mathbf{H}_2)$ (see Example 2.8). Starting with an instance $\mathbf{I}$ of NAE-Sat, for each variable $v \in I$, we add a binary function symbol f_v, and for each constraint C involving (not necessarily different) v_1, v_2, v_3, we add a 6-ary symbol g_C and the following identities:

$$f_{v_1}(x, y) \approx g_C(x, x, y, y, y, x),$$
$$f_{v_2}(x, y) \approx g_C(x, y, x, y, x, y),$$
$$f_{v_3}(x, y) \approx g_C(y, x, x, x, y, y).$$

The arity of f_v was chosen to be two so the assignment of a projection would correspond to an assignment of a value to v. In particular, each variable in the above identities corresponds to one of the elements from the domain. The function g_C has arity 6, since each constraint in NAE-Sat has exactly 6 satisfying assignments; the columns of variables on the right-hand side of these identities correspond to these satisfying assignments (three x's or three y's never align).

To conclude the proof of Theorem 3.12(1), it is enough to prove the following lemma:

Lemma 3.14. *Let* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{I}$ *be similar relational structures, and let* $\Sigma = \Sigma(\mathbf{A}, \mathbf{I})$ *be constructed as above. Then*

(1) if there is a homomorphism $h : \mathbf{I} \to \mathbf{A}$, *then* Σ *is trivial; and*
(2) if $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *satisfies* Σ, *then there is a homomorphism from* $\mathbf{I}$ *to* $\mathbf{B}$.

Proof. To prove item (1), assume that $h : \mathbf{I} \to \mathbf{A}$ is a homomorphism, and define $\zeta : \mathcal{U} \to \mathscr{P}_A$ by $\zeta(f_v) = p_i$ where i is chosen so $h(v) = a_i$. We extend this assignment to symbols from $\mathcal{V}$: Let $C = ((v_1, \dots, v_k), R)$ be a constraint of $\mathbf{I}$. Since $(h(v_1), \dots, h(v_k)) \in R^{\mathbf{A}}$, we can find a unique j such that $(a_{\pi_1(j)}, \dots, a_{\pi_k(j)})$ is equal to $(h(v_1), \dots, h(v_k))$. We set $\zeta(g_C) = p_j$. Clearly, this assignment satisfies all identities of Σ involving g_C. Thus, we found an assignment from $\mathcal{U} \cup \mathcal{V}$ to projections that satisfies Σ, proving that Σ is trivial.

For item (2), let us first suppose that Σ is satisfiable in projections and fix a satisfying assignment (of projections to symbols in $\mathcal{U} \cup \mathcal{V}$). In that case, we can define a map from I, equivalently $\mathcal{U}$, into A by assigning to v the a_i corresponding to the projection assigned to f_v. One easy way to identify the projection is to interpret it as a projection on the set A, i.e., suppose that $\zeta : \mathcal{U} \to \mathscr{P}_A$ is the assignment to projections, and define

$$h(v) = \zeta(f_v)(a_1, \dots, a_n). \tag{$\clubsuit$}$$

This would give a homomorphism to $\mathbf{A}$ by reversing the above argument. We only need a homomorphism to $\mathbf{B}$, but we also only have $\zeta : \mathcal{U} \cup \mathcal{V} \to \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. Still, we define $h : I \to B$ by the rule ($\clubsuit$). Clearly, this is a well-defined assignment; we only need to prove that h is a homomorphism, i.e., that for each constraint $C = ((v_1, \dots, v_k), R)$, we have $(h(v_1), \dots, h(v_k)) \in R^{\mathbf{B}}$. Consider the symbol g_C and its image under ζ. We know that $\zeta(g_C)$ is a polymorphism from $\mathbf{A}$ to $\mathbf{B}$ that satisfies the corresponding identities in Σ. Let us therefore substitute a_i for x_i, for $i \in [n]$, in those identities. We obtain the following:

$$\zeta(f_{v_1})(a_1, \dots, a_n) = \zeta(g_C)(a_{\pi_1(1)}, \dots, a_{\pi_1(m)})$$
$$\vdots$$
$$\zeta(f_{v_k})(a_1, \dots, a_n) = \zeta(g_C)(a_{\pi_k(1)}, \dots, a_{\pi_k(m)}).$$

But, since all the tuples $(a_{\pi_1(j)}, \dots, a_{\pi_k(j)})$, for $j \in [m]$, are in $R^{\mathbf{A}}$ and $\zeta(g_C)$ is a polymorphism from $\mathbf{A}$ to $\mathbf{B}$, we get that the resulting tuple $(h(v_1), \dots, h(v_k))$ is in $R^{\mathbf{B}}$. □

Remark 3.15. Note that if we take $\mathbf{A} = \mathbf{B}$ in the previous lemma, then we obtain that $\Sigma(\mathbf{A}, \mathbf{I})$ is trivial if and only if $\Sigma(\mathbf{A}, \mathbf{I})$ is satisfied in $\mathrm{Pol}(\mathbf{A})$ if and only if $\mathbf{I}$ maps homomorphically to $\mathbf{A}$.

3.3 From Minor Conditions to PCSP

Our proof of Theorem 3.12(2) follows another standard reduction in approximation: a conjunction of Long code testing and Label Cover [18]. If both $\mathbf{A}$ and $\mathbf{B}$ are Boolean ($A = B = \{0, 1\}$), then the construction can be viewed as a certain Long code test. Such analogy fails when A and B have a different size. Nevertheless, projections on the set A (as opposed to $\{0, 1\}$ for long codes) still play a crucial role in the completeness of this reduction. It has also appeared many times in the presented algebraic form and seems to be folklore (see, e.g., Reference [39, Lemma 3.8]).

The key idea of the construction is that the question "Is this bipartite minor condition satisfied by polymorphisms of $\mathbf{A}$?" can be interpreted as an instance of $\mathrm{CSP}(\mathbf{A})$. The main ingredient is that a polymorphism is a homomorphism from $\mathbf{A}^n$; this gives an instance whose solutions are exactly n-ary polymorphisms. Such instances are called *indicator problems* in [62]. By considering the union of several such instances (one for each function symbol appearing in Σ) and then introducing equality constraints that reflect the identities, we get that a solution to the obtained instance corresponds to polymorphisms satisfying the identities. We call the resulting instance an *indicator instance* (or an *indicator structure*) for Σ.

In detail, let us fix a template $(\mathbf{A}, \mathbf{B})$ and a bound on arity N. We start with a bipartite minor condition $(\Sigma, \mathcal{U}, \mathcal{V})$ with arity bounded by N, and construct an instance $\mathbf{I} = \mathbf{I}_\Sigma(\mathbf{A})$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ in three steps:

(1) for each n-ary symbol f in $\mathcal{U} \cup \mathcal{V}$, introduce into $\mathbf{I}$ a fresh copy of $\mathbf{A}^n$ where each element $(a_1, \dots, a_n) \in A^n$ is renamed to $v_{f(a_1,\dots,a_n)}$. To spell this out, for each relation $R^{\mathbf{A}}$, say of arity k, the relation $R^{\mathbf{I}}$ contains a tuple $(v_{f(a_{11},\dots,a_{1n})}, \dots, v_{f(a_{k1},\dots,a_{kn})})$ if and only if, for each $1 \le i \le n$, the tuple $(a_{1i}, \dots, a_{ki})$ is in $R^{\mathbf{A}}$;

(2) for each identity $f(x_1, \dots, x_l) \approx g(x_{\pi(1)}, \dots, x_{\pi(r)})$ in Σ, and all $a_1, \dots, a_l \in A$, add an equality constraint ensuring $v_{f(a_1,\dots,a_l)} = v_{g(a_{\pi(1)},\dots,a_{\pi(r)})}$;

(3) identify all pairs of variables connected by (a path of) equality constraints and then remove the equality constraints.

Clearly, the first and the second step can be done in log-space (note that the arity n is bounded by the constant N). The third step can be done in log-space by Reference [85]. We can therefore conclude that the indicator instance of a condition involving symbols of bounded arity can be constructed in log-space.

Lemma 3.16. *Let* $(\mathbf{A}, \mathbf{B})$ *be a template,* $N > 0$*, and let* $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. *The above construction gives a log-space reduction from* $\mathrm{PMC}_{\mathscr{M}}(N)$ *to* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$.

Proof. To prove completeness, suppose that a bipartite minor condition $(\Sigma, \mathcal{U}, \mathcal{V})$ is trivial and let $\zeta : \mathcal{U} \cup \mathcal{V} \to \mathscr{P}_A$ be a witness to that. We define a homomorphism from $\mathbf{I}$ to $\mathbf{A}$ by setting

$$h(v_{f(a_1,\dots,a_n)}) = \zeta(f)(a_1,\dots,a_n).$$

Note that h is well-defined due to step (2) in the above construction, i.e., because ζ-images of function symbols satisfy identities in Σ. Further, since every projection is a polymorphism of $\mathbf{A}$, h is a homomorphism.

To prove soundness, assume that there is a homomorphism $h : \mathbf{I} \to \mathbf{B}$. We reverse the above argument and define ζ by $\zeta(f)(a_1,\dots,a_n) = h(v_{f(a_1,\dots,a_n)})$ for all $a_1,\dots,a_n \in A$. Now, $\zeta(f) \in \mathscr{M}$ follows from the first step of the construction of $\mathbf{I}$, and satisfaction of the identities from Σ follows from the second and the third steps of the construction. □

This concludes the proof of Theorem 3.12.

Remark 3.17. In the above proof, we showed that if $\mathbf{I}_\Sigma(\mathbf{A}) \to \mathbf{B}$, then $\mathrm{Pol}(\mathbf{A},\mathbf{B})$ satisfies Σ. The converse is also true, which can be shown by generalising the completeness argument in the above proof. The fact that $\mathbf{I}_\Sigma(\mathbf{A}) \to \mathbf{B}$ if and only if $\mathrm{Pol}(\mathbf{A},\mathbf{B})$ satisfies Σ gives a useful intuition about the indicator structure.

4 RELATIONAL CONSTRUCTIONS

In the previous section, we have shown that the existence of a minion homomorphism between polymorphism minions of two templates gives us a log-space reduction between the corresponding PCSPs. The presented proof is both self-contained and succinct. In fact, we proved a generalisation of Reference [29, Theorem D.1], without referencing this special case, or even the Galois correspondence that underlines it [83]. This section describes several concepts that give understanding of how our result relates to these two results and the theory developed for CSPs in its current form [15, 16].

Our goal is to present a deeper understanding of when and how our reduction works. Namely, Theorem 3.1 shows that $\mathrm{PCSP}(\mathbf{A}_2,\mathbf{B}_2)$ can be reduced to $\mathrm{PCSP}(\mathbf{A}_1,\mathbf{B}_1)$ under the assumption that there is a minion homomorphism from $\mathrm{Pol}(\mathbf{A}_1,\mathbf{B}_1)$ to $\mathrm{Pol}(\mathbf{A}_2,\mathbf{B}_2)$. In this section, we show that this assumption is satisfied if and only if $(\mathbf{A}_2,\mathbf{B}_2)$ can be obtained from $(\mathbf{A}_1,\mathbf{B}_1)$ using certain relational constructions that we call *pp-constructions*, and that happens if and only if there is a homomorphism from a certain relational structure constructed using $\mathrm{Pol}(\mathbf{A}_1,\mathbf{B}_1)$ and $\mathbf{A}_2$, which we call the *free structure*, to $\mathbf{B}_2$.

4.1 Free Structures

It follows from Theorem 3.1 that if we have an NP-hard $\mathrm{PCSP}(\mathbf{A}_2,\mathbf{B}_2)$, then, for any PCSP template $(\mathbf{A}_1,\mathbf{B}_1)$ such that there is a minion homomorphism ξ from $\mathrm{Pol}(\mathbf{A}_1,\mathbf{B}_1)$ to $\mathrm{Pol}(\mathbf{A}_2,\mathbf{B}_2)$, the problem $\mathrm{PCSP}(\mathbf{A}_1,\mathbf{B}_1)$ is also NP-hard. The existence of such a minion homomorphism seems to be a global property of a minion $\mathrm{Pol}(\mathbf{A}_1,\mathbf{B}_1)$. Nevertheless, given that the target template is finite, it turns out that it is not the case, as we show in this section. This allows us to obtain a reduction from knowing only the structure of polymorphisms of arity bounded by a fixed parameter that depends on the target template—we will use this to obtain new hardness results in Section 6.

One way to get a bound on the arity of polymorphisms is to compose the two reductions used to obtain Theorem 3.1 (with the trivial reduction between the corresponding PMC-problems) and analyse the soundness of the composite reduction. Here, we will describe another way of getting such bound, generalising Reference [16, Section 7]. Our proof is based on two facts: First, a minion homomorphism from a minion $\mathscr{M}$ to a minion $\mathscr{A} \subseteq \mathscr{O}(A,B)$ is fully determined by its image on functions of arity $|A|$. Second, for a minion $\mathscr{M}$ and a relational structure $\mathbf{A}$ there is a most general structure **F** such that there is a minion homomorphism from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{A}, \mathbf{F})$. The structure **F** is the most general in the sense that it maps homomorphically to any other structure **B** similar to **A** for which there is a minion homomorphism from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$.[5]

To simplify the notation, we assume that $A = [n]$. The general case, as well as the case when **A** is an infinite structure, can be dealt with as in References [16, 81].

Definition 4.1. Let **A** be a finite relational structure on the set $A = [n]$, and $\mathscr{M}$ a minion (not necessarily related to **A**). The *free structure of* $\mathscr{M}$ *generated by* **A** is a relational structure $\mathbf{F}_{\mathscr{M}}(\mathbf{A})$ similar to **A**. Its universe consists of n-ary functions of $\mathscr{M}$, i.e., $F_{\mathscr{M}}(A) = \mathscr{M}^{(n)}$. Let $R^{\mathbf{A}}$ be a relation of **A** of arity k, and let $m = |R^{\mathbf{A}}|$ and $R^{\mathbf{A}} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$. The relation $R^{\mathbf{F}_{\mathscr{M}}(\mathbf{A})}$ is defined as the set of all k-tuples $(f_1, \dots, f_k) \in F_{\mathscr{M}}(A)$ such that there exists an m-ary function $g \in \mathscr{M}$ that satisfies

$$f_1(x_1, \dots, x_n) = g(x_{\mathbf{r}_1(1)}, \dots, x_{\mathbf{r}_m(1)})$$
$$\vdots$$
$$f_k(x_1, \dots, x_n) = g(x_{\mathbf{r}_1(k)}, \dots, x_{\mathbf{r}_m(k)}).$$

(Note that the indices of variables x_i on the right-hand side of the equalities are distributed so the columns correspond to the tuples in the relation $R^{\mathbf{A}}$.)

There is a natural minion homomorphism ϕ from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{A}, \mathbf{F}_{\mathscr{M}}(\mathbf{A}))$. The image of $g \in \mathscr{M}^{(k)}$ under ϕ is defined by

$$\phi(g) : (a_1, \dots, a_k) \mapsto f(x_1, \dots, x_n),$$

where $a_1, \dots, a_k \in [n]$ and $f(x_1, \dots, x_n) = g(x_{a_1}, \dots, x_{a_k})$. The relation $R^{\mathbf{F}_{\mathscr{M}}(\mathbf{A})}$ is then the smallest relation S such that each function from $\phi(\mathscr{M})$ is a polymorphism from $(A; R^{\mathbf{A}})$ to $(F_{\mathscr{M}}(A); S)$.

Example 4.2. As an example let us describe the free structures of the minion $\mathscr{H}$ of all Boolean functions of the form $x_{i_1} \wedge \dots \wedge x_{i_k}$. In fact, $\mathscr{H}$ is the set of polymorphisms of the CSP template for Horn 3-Sat, i.e., the structure

$$\mathbf{H} = (\{0, 1\}; x \wedge y \to z, x \wedge y \to \neg z, \{0\}, \{1\})$$

(see Reference [15, Example 5]). Note that the n-ary functions $f \in \mathscr{H}$ can be identified with non-empty subsets $[n]$: We identify $\emptyset \neq I \subseteq [n]$ with $f_I = \bigwedge_{i \in I} x_i$. Clearly, any function from $\mathscr{H}$ can be expressed in this way. Also $f_I = f_J$ if and only if $I = J$.

Now, fix a relational structure **A** and assume $A = [n]$. The free structure $\mathbf{F} = \mathbf{F}_{\mathscr{H}}(\mathbf{A})$ of $\mathscr{H}$ generated by **A** is then defined as follows: The elements of **F** are the n-ary functions from $\mathscr{H}$, i.e., the non-empty subsets of A. For a relation $R^{\mathbf{A}} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$ of arity k, the relation $R^{\mathbf{F}}$ consists of all k-tuples $(f_{I_1}, \dots, f_{I_k})$ for which there exists a function $g_J \in \mathscr{H}^{(m)}$ such that

$$f_{I_i}(x_1, \dots, x_n) = g_J(x_{\mathbf{r}_1(i)}, \dots, x_{\mathbf{r}_m(i)})$$

for all i. Note that this identity is satisfied if and only if $I_i = \{\mathbf{r}_j(i) \mid j \in J\}$, since the left-hand side is $\bigwedge_{a \in I_i} x_a$ and similarly, the right-hand side is $\bigwedge_{j \in J} x_{\mathbf{r}_j(i)}$. In other words, the elements of $R^{\mathbf{F}}$ can be viewed as those k-tuples $(I_1, \dots, I_k)$ of non-empty subsets of A for which there exists a subset $\{\mathbf{r}_j \mid j \in J\} \subseteq R^{\mathbf{A}}$ such that $I_i = \{\mathbf{r}_j(i) \mid j \in J\}$ for all $i = 1, \dots, k$. The resulting structure is isomorphic to the structure introduced in Reference [48, Section 6.1.1] (called U there) to characterise the so-called CSPs of width 1—now it is often referred to as the "power structure" of **A**.

The following lemma provides a useful connection between the free structure and the condition $\Sigma(\mathbf{A}, \mathbf{I})$ constructed in Section 3.2:

[5] In the case $\mathscr{M}$ is a clone, the structure **F** is obtained by considering the free algebra generated by **A** in the variety of actions of $\mathscr{M}$ (see Reference [81, Section 3.2] for more detailed description of this case).

Lemma 4.3. *Let $\mathscr{M}$ be a minion and* $\mathbf{A}$ *a relational structure. Then $\mathscr{M}$ satisfies the condition* $\Sigma(\mathbf{A}, \mathbf{F}_{\mathscr{M}}(\mathbf{A}))$.

Proof. Let $A = [n]$ and $\mathbf{F} = \mathbf{F}_{\mathscr{M}}(\mathbf{A})$. We claim that $\Sigma(\mathbf{A}, \mathbf{F})$ is satisfied in $\mathscr{M}$. Note that $\mathcal{U}$ consists of symbols f_v where $v \in F = \mathscr{M}^{(n)}$. Therefore, we can define $\zeta : \mathcal{U} \to \mathscr{M}$ by $\zeta(f_v) = v$. To extend this map to $\mathcal{V}$, pick a constraint C corresponding to $(v_1, \dots, v_k) \in R^{\mathbf{F}}$ and let $R^{\mathbf{A}} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$. We need to find $\zeta(g_C) \in \mathscr{M}^{(m)}$ such that

$$\zeta(f_{v_i})(x_1, \dots, x_n) \approx \zeta(g_C)(x_{\mathbf{r}_1(i)}, \dots, x_{\mathbf{r}_m(i)})$$

for all $i = 1, \dots, k$. Existence of such $\zeta(g_C)$ is guaranteed by the definition of $R^{\mathbf{F}}$ and the fact that the left-hand side is equal to $v_i(x_1, \dots, x_n)$. □

The following lemma relates free structures with minion homomorphisms between minions. This gives a key correspondence between homomorphisms of certain relational structures and minion homomorphisms.

Lemma 4.4. *Let $\mathscr{M}$ be a minion, and* $(\mathbf{A}, \mathbf{B})$ *a PCSP-template. There is a 1-to-1 correspondence between homomorphisms from* $\mathbf{F}_{\mathscr{M}}(\mathbf{A})$ *to* $\mathbf{B}$ *and minion homomorphisms from $\mathscr{M}$ to* $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$.

Proof. Assume that $A = [n]$. A homomorphism $c : \mathbf{F}_{\mathscr{M}}(\mathbf{A}) \to \mathbf{B}$ is then a mapping from $\mathscr{M}^{(n)}$ to B. We would like to connect it to the restriction of a minion homomorphism ξ from $\mathscr{M}$ to $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$ on the n-ary functions. The only obstacle is that an image of an n-ary function under ξ is a mapping $A^n \to B$, not an element of B. Following the decoding used in Lemma 3.14, we identify such a mapping with its image of the tuple $(1, \dots, n)$. Formally, to a minion homomorphism $\xi : \mathscr{M} \to \operatorname{Pol}(\mathbf{A}, \mathbf{B})$, we assign the map $c_\xi : f \mapsto \xi(f)(1, \dots, n)$. The fact that this map is a homomorphism from $\mathbf{F}_{\mathscr{M}}(\mathbf{A})$ to $\mathbf{B}$ follows by argument similar to the proof of Lemma 3.14(2).

It remains to prove that any such restriction can be extended in a unique way. Let ϕ denote the natural minion homomorphism from $\mathscr{M}$ to $\operatorname{Pol}(\mathbf{A}, \mathbf{F}_{\mathscr{M}}(\mathbf{A}))$, i.e., for $f \in \mathscr{M}$, say of arity m, $\phi(f)$ is the m-ary function from $\operatorname{Pol}(\mathbf{A}, \mathbf{F}_{\mathscr{M}}(\mathbf{A}))$ defined as follows: $\phi(f)(a_1, \dots, a_m)$ is the element $g \in \mathscr{M}^{(n)}$ such that

$$f(x_{a_1}, \dots, x_{a_m}) = g(x_1, \dots, x_n). \tag{$\blacklozenge$}$$

Given $c : \mathbf{F}_{\mathscr{M}}(\mathbf{A}) \to \mathbf{B}$, we define $\xi_c : \mathscr{M} \to \operatorname{Pol}(\mathbf{A}, \mathbf{B})$ by

$$\xi_c(f) : (a_1, \dots, a_m) \mapsto c(\phi(f)(a_1, \dots, a_m)).$$

It is easy to see that the mapping ξ_c preserves taking minors. We need to prove that $\xi_c(f) \in \operatorname{Pol}(\mathbf{A}, \mathbf{B})$ for each f, but that follows from the fact that $\phi(f) \in \operatorname{Pol}(\mathbf{A}, \mathbf{F}_{\mathscr{M}}(\mathbf{A}))$, i.e., it is a homomorphism from $\mathbf{A}^m$ to $\mathbf{F}_{\mathscr{M}}(\mathbf{A})$ and c is a homomorphism from $\mathbf{F}_{\mathscr{M}}(\mathbf{A})$ to $\mathbf{B}$.

For the uniqueness, suppose that $\chi : \mathscr{M} \to \operatorname{Pol}(\mathbf{A}, \mathbf{B})$ is a minion homomorphism such that $c_\chi = c$, i.e., $\chi(g)(1, \dots, n) = c(g)$ for all $g \in \mathscr{M}^{(n)}$. Since χ preserves minors, we get that for any $(a_1, \dots, a_m) \in A^m$ and f satisfying ($\blacklozenge$), we have $\chi(f)(a_1, \dots, a_m) = \chi(g)(1, \dots, n) = c(g)$ concluding $\chi = \xi_c$. □

We remark in passing that the correspondence in the above lemma is *natural* in the categorical sense. More precisely, the mapping $- \mapsto \mathbf{F}_{-}(\mathbf{A})$, assigning to a minion the free structure generated by $\mathbf{A}$ is an adjoint to the functor $\operatorname{Pol}(\mathbf{A}, -)$.

Remark 4.5. The previous lemma holds also for minions $\mathscr{M}$ over infinite sets A' and B' and for infinite relational structures $\mathbf{A}$ and $\mathbf{B}$. The proof applies to the case when $\mathbf{A}$ is finite. It can be generalised to an infinite $\mathbf{A}$ using a more general definition of the free structure along the lines of Reference [81, Definition 3.1].

4.2 Pp-constructions

We introduce two relational constructions: *pp-powers*, which generalise the analogous notion for CSP templates [16], and *homomorphic relaxations*, which generalise both simple relaxations (Definition 2.25) and homomorphic equivalence for CSP templates. Relaxation is in fact a very natural notion for promise problems: The idea is that any problem that has a stronger promise has to be at least as easy as the original problem.

Definition 4.6. Assume that $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}', \mathbf{B}')$ are similar PCSP templates. We say that $(\mathbf{A}', \mathbf{B}')$ is a *homomorphic relaxation*[6] of $(\mathbf{A}, \mathbf{B})$ if there are homomorphisms $h_A : \mathbf{A}' \to \mathbf{A}$ and $h_B : \mathbf{B} \to \mathbf{B}'$.

All the relaxations that appear in this article are homomorphic relaxations, therefore, we will usually omit the word "homomorphic." In particular, the simple relaxation defined in Definition 2.25 is a special case of homomorphic relaxation, namely, one where both h_A and h_B are identity maps on the corresponding domains.

Clearly, if $(\mathbf{A}', \mathbf{B}')$ is a relaxation of $(\mathbf{A}, \mathbf{B})$, then the trivial reduction (which does not change the input) is a reduction from $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}')$ to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, since it is a strengthening of the promise (and weakening of the objective).

The following is a generalisation of the definition of a pp-power for CSP templates (see Reference [15, Definition 14]):

Definition 4.7. Let $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}', \mathbf{B}')$ be two PCSP templates. We say that $(\mathbf{A}', \mathbf{B}')$ is an (nth) *pp-power* of $(\mathbf{A}, \mathbf{B})$ if $A' = A^n$, $B' = B^n$, and, if we view k-ary relations on $\mathbf{A}'$ and $\mathbf{B}'$ as kn-ary relations on A and B, respectively, then $(\mathbf{A}', \mathbf{B}')$ is pp-definable in $(\mathbf{A}, \mathbf{B})$ in the sense of Definition 2.25.

Lemma 4.8. *Let $(\mathbf{A}_1, \mathbf{B}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2)$ be two PCSP templates. If*

(1) $(\mathbf{A}_2, \mathbf{B}_2)$ is a relaxation of $(\mathbf{A}_1, \mathbf{B}_1)$, or
(2) $(\mathbf{A}_2, \mathbf{B}_2)$ is a pp-power of $(\mathbf{A}_1, \mathbf{B}_1)$,

then there is a minion homomorphism $\xi : \mathrm{Pol}(\mathbf{A}_1, \mathbf{B}_1) \to \mathrm{Pol}(\mathbf{A}_2, \mathbf{B}_2)$.

Proof. For $i = 1, 2$, let $\mathscr{M}_i = \mathrm{Pol}(\mathbf{A}_i, \mathbf{B}_i)$.

(1) Assume $h_A : \mathbf{A}_2 \to \mathbf{A}_1$ and $h_B : \mathbf{B}_1 \to \mathbf{B}_2$ are homomorphisms, and define $\xi : \mathscr{M}_1 \to \mathscr{M}_2$ by $\xi(f) : (a_1, \dots, a_n) \mapsto h_B(f(h_A(a_1), \dots, h_A(a_n)))$. It is easy to see that $\xi(f)$ is a polymorphism of $(\mathbf{A}_2, \mathbf{B}_2)$ (as it is a composition of homomorphisms), and also that ξ preserves minors.

(2) Assume that $(\mathbf{A}_2, \mathbf{B}_2)$ is nth pp-power of $(\mathbf{A}_1, \mathbf{B}_1)$. We define ξ to map each f to its component-wise action on $A_2 = A_1^n$, i.e.,

$$\xi(f) : (\mathbf{a}_1, \dots, \mathbf{a}_m) \mapsto \big(f(\mathbf{a}_1(1), \dots, \mathbf{a}_m(1)), \dots, f(\mathbf{a}_1(n), \dots, \mathbf{a}_m(n))\big).$$

Clearly, ξ is minor-preserving. Also $\xi(f)$ is a polymorphism, since f is a polymorphism from $\mathbf{A}_1$ to $\mathbf{B}_1$, and therefore it preserves all pp-definable relations. □

The above together with Theorem 3.1 prove that both the constructions yield a log-space reduction between the corresponding PCSPs. We remark that the same can also be proven directly. Since both constructions yield a reduction, we can combine them and still retain log-space reductions. This motivates the next definition, which is similar to the corresponding notion for CSPs (see Reference [15, Section 3.4]).

[6] Homomorphic relaxations of CSP templates (i.e., $\mathbf{A} = \mathbf{B}$, but $\mathbf{A}'$ and $\mathbf{B}'$ can be different) have been considered in Reference [30] where they are called *homomorphic sandwiches*.

Definition 4.9. We say that $(\mathbf{A}', \mathbf{B}')$ is *pp-constructible* from $(\mathbf{A}, \mathbf{B})$ if there exists a sequence

$$(\mathbf{A}, \mathbf{B}) = (\mathbf{A}_1, \mathbf{B}_1), \dots, (\mathbf{A}_k, \mathbf{B}_k) = (\mathbf{A}', \mathbf{B}')$$

of templates where each $(\mathbf{A}_{i+1}, \mathbf{B}_{i+1})$ is a pp-power or a homomorphic relaxation of $(\mathbf{A}_i, \mathbf{B}_i)$. We say that $(\mathbf{A}', \mathbf{B}')$ is pp-constructible from $\mathbf{A}$ if it is pp-constructible from $(\mathbf{A}, \mathbf{A})$.

Corollary 4.10. *If* $(\mathbf{A}', \mathbf{B}')$ *is pp-constructible from* $(\mathbf{A}, \mathbf{B})$*, then there is a minion homomorphism from* $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$ *to* $\operatorname{Pol}(\mathbf{A}', \mathbf{B}')$.

Proof. Assume that we have a sequence $(\mathbf{A}_1, \mathbf{B}_1)$, …, $(\mathbf{A}_k, \mathbf{B}_k)$ as in the definition of pp-constructibility, and $\mathscr{M}_i = \operatorname{Pol}(\mathbf{A}_i, \mathbf{B}_i)$ for $i \in [k]$. That means, that by Lemma 4.8, we have minion homomorphisms $\xi_i : \mathscr{M}_i \to \mathscr{M}_{i+1}$ for all $i < k$. The minion homomorphism from $\operatorname{Pol}(\mathbf{A}, \mathbf{B}) = \mathscr{M}_1$ to $\operatorname{Pol}(\mathbf{A}', \mathbf{B}') = \mathscr{M}_k$ is obtained by composing all ξ_i's. □

An example of a template that can be pp-constructed is a template obtained using the free structure.

Lemma 4.11. *Let* $(\mathbf{A}_1, \mathbf{B}_1)$ *be a PCSP template,* $\mathscr{M} = \operatorname{Pol}(\mathbf{A}_1, \mathbf{B}_1)$*, and* $\mathbf{A}_2$ *be a relational structure. Then the template* $(\mathbf{A}_2, \mathbf{F}_{\mathscr{M}}(\mathbf{A}_2))$ *is a relaxation of a pp-power of* $(\mathbf{A}_1, \mathbf{B}_1)$.

Proof. Let us first comment on some ideas underlying the proof. We will argue that the whole reduction from $\operatorname{PCSP}(\mathbf{A}_2, \mathbf{F})$, where $\mathbf{F} = \mathbf{F}_{\mathscr{M}}(\mathbf{A}_2)$, to $\operatorname{PCSP}(\mathbf{A}_1, \mathbf{B}_1)$ according to the proof of Theorem 3.1 is basically a pp-construction of $(\mathbf{A}_2, \mathbf{F})$ from $(\mathbf{A}_1, \mathbf{B}_1)$, and in particular it is a relaxation of a pp-power. Note that Theorem 3.1 applies, since we have a minion homomorphism $\phi : \mathscr{M} \to \operatorname{Pol}(\mathbf{A}_2, \mathbf{F})$, as defined in the proof of Lemma 4.4.

For the formal proof, assume that $A_2 = [n]$, and let $N = |A_1|^n$. We first describe an Nth pp-power of $(\mathbf{A}_1, \mathbf{B}_1)$ and then argue that $(\mathbf{A}_2, \mathbf{F})$ is a homomorphic relaxation of this power. Let $R^{\mathbf{A}_2} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$ be a relation of $\mathbf{A}_2$ of arity k. Recall that the relation $R^{\mathbf{F}}$ is defined as the set of all k-tuples $(f_1, \dots, f_k)$ of functions from A_1^n to B_1 for which there exists a polymorphism $g : \mathbf{A}_1^m \to \mathbf{B}_1$ such that

$$f_i(x_1, \dots, x_n) = g(x_{\mathbf{r}_1(i)}, \dots, x_{\mathbf{r}_m(i)}) \tag{$\spadesuit$}$$

for all $i \in [k]$. Note that we do not need to require that f_i's are polymorphisms, since the mentioned identities enforce that all f_i's are minors of the polymorphism g. We argue that the set of all such tuples is pp-definable in $\mathbf{B}_1$ as a kN-ary relation. In fact, since this is just a bipartite minor condition, we can use the construction from Section 3.3 and adapt it to provide a pp-formula: The free variables of the pp-formula are labelled by $f_i(a_1, \dots, a_n)$ where $i \in [k]$ and $a_1, \dots, a_n \in A_1$, the quantified variables are labelled by $g(a_1, \dots, a_m)$ where $a_1, \dots, a_m \in A_1$. Further, let $\mathcal{R}^{\mathbf{A}_1}$ denote the set of all relational symbols of $\mathbf{A}_1$. We use the following pp-formulas to define the pp-power (see Definition 2.25):

$$\Psi_R(v_{f_i(a_1,\dots,a_n)}, \dots) = \exists_{a_1,\dots,a_m \in A_1} v_{g(a_1,\dots,a_m)}$$
$$\bigwedge_{S \in \mathcal{R}^{\mathbf{A}_1}} \ \bigwedge_{\mathbf{s}_1,\dots,\mathbf{s}_m \in S^{\mathbf{A}_1}} (v_{g(s_1(1),\dots,s_m(1))}, \dots, v_{g(s_1(\operatorname{ar}(S)),\dots,s_m(\operatorname{ar}(S)))}) \in S \wedge$$
$$\bigwedge_{i \in [k]} \ \bigwedge_{a_1,\dots,a_n \in A_1} v_{f_i(a_1,\dots,a_n)} = v_{g(a_{\mathbf{r}_1(i)},\dots,a_{\mathbf{r}_m(i)})}.$$

The first conjunction ensures that the values assigned to $v_{g(a_1,\dots,a_m)}$'s give a valid polymorphism g, the second conjunction then ensures that this polymorphism will satisfy (♠). Let us denote the resulting pp-power by $(\mathbf{A}_1', \mathbf{B}_1')$.

We still need to find homomorphisms from $\mathbf{A}_2$ to $\mathbf{A}_1'$ and from $\mathbf{B}_1'$ to $\mathbf{F}$. The first one can be constructed following an argument from Lemma 3.16: We define $h_A : \mathbf{A}_2 \to \mathbf{A}_1'$ by $h_A(a) = p_a$ where p_a

denotes the projection on the ath coordinate (note that $\mathbf{A}_1'$ is an $|A_1|^{|A_2|}$-th pp-power of $\mathbf{A}_1$). Further, if $(a_1, \dots, a_k) \in R^{\mathbf{A}_2}$, i.e., $(a_1, \dots, a_k) = \mathbf{r}_i$ for some i, then choosing the value a_i for $v_{g(a_1,\dots,a_m)}$ will give a satisfying assignment of $\Psi_R^{\mathbf{A}_1}$. The homomorphism from $\mathbf{B}_1'$ to $\mathbf{F}$ is easier: Since $F \subseteq B_1'$, and $\Psi_R^{\mathbf{B}_1}(f_1, \dots, f_k)$ if and only if $(f_1, \dots, f_k) \in R^{\mathbf{F}}$ (this was the motivation behind the definition of Ψ_R), we can define $h_B : B_1' \to F$ as any extension of the identity mapping on F. □

Finally, we are ready to formulate and prove the main result of this section.

Theorem 4.12. *Let* $(\mathbf{A}_i, \mathbf{B}_i)$ *for* $i = 1, 2$ *be PCSP templates and* $\mathscr{M}_i = \operatorname{Pol}(\mathbf{A}_i, \mathbf{B}_i)$. *The following are equivalent:*

(1) *There exists a minion homomorphism* $\xi : \mathscr{M}_1 \to \mathscr{M}_2$.
(2) $\mathscr{M}_2$ *satisfies all bipartite minor conditions satisfied in* $\mathscr{M}_1$.
(3) $\mathscr{M}_2$ *satisfies the condition* $\Sigma(\mathbf{A}_2, \mathbf{F}_{\mathscr{M}_1}(\mathbf{A}_2))$.
(4) *There exists a homomorphism from* $\mathbf{F}_{\mathscr{M}_1}(\mathbf{A}_2)$ *to* $\mathbf{B}_2$.
(5) $(\mathbf{A}_2, \mathbf{B}_2)$ *is a homomorphic relaxation of a pp-power of* $(\mathbf{A}_1, \mathbf{B}_1)$.
(6) $(\mathbf{A}_2, \mathbf{B}_2)$ *is pp-constructible from* $(\mathbf{A}_1, \mathbf{B}_1)$.

Proof. The implication (5) → (6) is trivial, (6) → (1) follows from Corollary 4.10, (1) → (2) follows directly from the definition (see Definition 2.21 and the comment after it), (2) → (3) follows from Lemma 4.3 for $\mathscr{M} = \mathscr{M}_1$ and $\mathbf{A} = \mathbf{A}_2$, and (3) → (4) follows directly from Lemma 3.14(2).

Finally, we prove (4) → (5). Lemma 4.11 gives that $(\mathbf{A}_2, \mathbf{F}_{\mathscr{M}_1}(\mathbf{A}_2))$ is a relaxation of a pp-power of $(\mathbf{A}_1, \mathbf{B}_1)$, the homomorphism from $\mathbf{F}_{\mathscr{M}_1}(\mathbf{A}_2)$ to $\mathbf{B}_2$ then proves that $(\mathbf{A}_2, \mathbf{B}_2)$ is a relaxation of $(\mathbf{A}_2, \mathbf{F}_{\mathscr{M}_1}(\mathbf{A}_2))$. We obtain the desired claim by composing the two relaxations into one. □

5 HARDNESS FROM THE PCP THEOREM

The celebrated PCP theorem [3, 4] is a starting point for many proofs of inapproximability of many problems. As an example, we refer to the work of Håstad [56] that derives inapproximability of several CSPs from the PCP theorem. Also note that the PCP theorem itself can be formulated as a result on inapproximability of the CSP (see Reference [44, Theorem 1.3]). Concrete results on hardness of many PCSPs rely on the PCP theorem (e.g., References [7, 26, 47, 59, 65]). A common approach for using the PCP theorem is to first derive hardness of some approximation version of Label Cover, or some of its variants, and then using gadgets reduce from Label Cover to PCSP.

In the scope of this article, we described a reduction from MC, which is essentially Label Cover, to PCSP (see Section 3.1). The present section then derives some algebraic conditions for applicability of this reduction when applied to Label Cover itself. We note that there are many approximation versions of Label Cover and many variants of PCPs. Most can be used as a starting point for a reduction, and we do not provide an exhaustive description. We focus on a few versions of Label Cover, including a plain approximation version thereof and one that was used in Reference [47] to obtain NP-hardness of approximate hypergraph colouring.

As mentioned above, the general approach for a reduction from (some version of) Label Cover to some PCSP$(\mathbf{A}, \mathbf{B})$ is to interpret an instance of Label Cover as a minor condition (as described in the beginning of Section 3.1) and then reduce from PMC$_{\mathscr{M}}$, where $\mathscr{M} = \operatorname{Pol}(\mathbf{A}, \mathbf{B})$, to PCSP$(\mathbf{A}, \mathbf{B})$ using Theorem 3.12. To ensure that this composite reduction works, it is enough to relate the yes- and no-instances of the LC, or its variant, and the corresponding PMC. Proving completeness (i.e., that yes-instances are preserved) is usually straightforward, while proving soundness (i.e., preserving no-instances) require some extra work.

Before we get to more general cases, let us briefly focus on a reduction from the plain (exact) Label Cover.

5.1 Reduction from Label Cover

As noted before, exact Label Cover is essentially the same as deciding non-triviality of bipartite minor conditions. This also means that we can use Theorem 3.12(1) to give an immediate proof that MC is NP-hard. A similar reduction from some NP-hard CSP is a commonly used argument for NP-hardness of Label Cover.

Theorem 5.1. *MC(N) is NP-hard for each $N \geq 3$.*

Proof. We reduce from 1-in-3-Sat using Theorem 3.12(1): Let $\mathbf{T}$ denote the CSP template of 1-in-3-Sat and $R^{\mathbf{T}}$ its ternary relation (see Example 2.8). It is well-known that every polymorphism of $\mathbf{T}$ is a projection, i.e., $\mathrm{Pol}(\mathbf{T}) = \mathscr{P}_2$. The mentioned theorem then gives a reduction from CSP$(\mathbf{T})$ to $\mathrm{PMC}_{\mathscr{P}_2}(N)$ for each N that is larger than both the size of the domain of $\mathbf{T}$, which is 2, and the number of tuples in $R^{\mathbf{T}}$, which is 3. Therefore, $\mathrm{PMC}_{\mathscr{P}_2}(N)$ is NP-hard for each $N \geq 3$. Since $\mathrm{PMC}_{\mathscr{P}_2}(N)$ is the same as $\mathrm{MC}(N)$, we obtain the desired hardness. □

To align no-instances of $\mathrm{MC}(N)$ and $\mathrm{PMC}_{\mathscr{M}}(N)$, we need that $\mathscr{M}$ does not satisfy any non-trivial bipartite minor conditions involving symbols of arity at most N. Since the theorem above gives NP-hardness of $\mathrm{MC}(N)$ for any $N = 3$, this means that arity at most 3 is enough, as we state in the following direct corollary of Theorem 3.12(2) and Theorem 5.1:

Corollary 5.2. *If* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *does not satisfy any non-trivial bipartite minor condition of arity at most three, then PCSP*$(\mathbf{A}, \mathbf{B})$ *is NP-hard.*

The assumption of the above corollary can be also satisfied by constructing a minor-preserving map $\xi : \mathscr{M}^{(3)} \to \mathscr{P}_2^{(3)}$. The search for such a minion homomorphism can be easily automated, which, with a suitable implementation, can be useful for small enough structures. We also remark that, since it is enough to work with binary and ternary functions, this might simplify some combinatorial arguments.

We now show that a minion that does not satisfy any non-trivial bipartite minor condition of small arities cannot satisfy such a condition of large arity. This has been also observed in Reference [17, Section 5.3].

Proposition 5.3. *The following are equivalent for every minion $\mathscr{M}$:*

(1) $\mathscr{M}$ does not satisfy any non-trivial bipartite minor condition of arity at most three.
(2) There exists a minion homomorphism from $\mathscr{M}$ to $\mathscr{P}_2$.

Proof. The implication (2) → (1) is obvious; let us prove (1) → (2). We use the notation from the proof of Theorem 5.1 and follow the proof of Theorem 4.12(3) → (1). The key observation is that the condition $\Sigma(\mathbf{T}, \mathbf{F}_{\mathscr{M}}(\mathbf{T}))$ is of arity at most three, and therefore it is trivial by the assumption and Lemma 4.3. The bound on arity is clear from the construction: The condition is composed of identities of the form

$$f_{v_1}(x, y) \approx g_{(v_1, v_2, v_3), R}(x, x, y),$$
$$f_{v_2}(x, y) \approx g_{(v_1, v_2, v_3), R}(x, y, x),$$
$$f_{v_3}(x, y) \approx g_{(v_1, v_2, v_3), R}(y, x, x),$$

where $(v_1, v_2, v_3) \in R^{\mathbf{F}_{\mathscr{M}}(\mathbf{T})}$. Thus, by Lemma 3.14(2), we get that $\mathbf{F}_{\mathscr{M}}(\mathbf{T}) \to \mathbf{T}$, which implies that there is a minor homomorphism from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{T}) = \mathscr{P}_2$ by, e.g., Lemma 4.4. □

Remark 5.4. The above proposition can be easily generalised for minions on infinite sets: The finiteness was used only to ensure that $\mathscr{M}^{(2)}$, and consequently also the free structure $\mathbf{F}_{\mathscr{M}}(\mathbf{T})$, is finite. This can be circumvented by a standard compactness argument. See, e.g., Reference [23, Lemma III.5].

The significance of the above proposition and the considerations of this subsection is that it implies that a reduction from $\mathrm{LC}(N)$ using polymorphism gadgets (following Section 3.3) works for some N if and only if the same reduction works for $N = 3$.

Example 5.5 (See also Example 2.22). In Reference [26], the authors prove that $\mathrm{PCSP}(\mathbf{K}_3, \mathbf{K}_4)$ is NP-hard by a reduction from LC. This in fact means that there is a simple gadget reduction from, e.g., 1-in-3-SAT, or equivalently, from $\mathrm{CSP}(\mathbf{T})$. This reduction then uses only properties of binary and ternary polymorphisms from $\mathbf{K}_3$ to $\mathbf{K}_4$.

We use Section 3 for the reduction: Given an instance $\mathbf{I}$ of 1-in-3-SAT, we first obtain a bipartite condition $\Sigma = \Sigma(\mathbf{I}, \mathbf{T})$. This condition (similarly as in the proof of Proposition 5.3 involves only symbols of arity two and three. From Σ, we then obtain a graph $\mathbf{G} = \mathbf{I}_\Sigma(\mathbf{K}_3)$. The goal is to show that if $\mathbf{I} \to \mathbf{T}$, then $\mathbf{G}$ is 3-colourable (which is the easier part, and we refer to Section 3 for the proof), and that if $\mathbf{G}$ is 4-colourable, then $\mathbf{I} \to \mathbf{T}$. The latter can be observed by constructing a homomorphism from the free structure $\mathbf{F}_{\mathscr{M}}(\mathbf{T})$ to $\mathbf{T}$ where $\mathscr{M} = \mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_4)$. Let us sketch one such homomorphism $h : \mathbf{F}_{\mathscr{M}}(\mathbf{T}) \to \mathbf{T}$.

The elements of $\mathbf{F}_{\mathscr{M}}(\mathbf{T})$ are binary polymorphisms of $(\mathbf{K}_3, \mathbf{K}_4)$. Therefore, to construct a homomorphism, we need to choose a Boolean value for each of the colourings of $\mathbf{K}_3^2$ with 4 colours. Such colourings are easy to describe; in fact the "trash colour lemma" that we mentioned in Example 2.22 is much easier to prove for binary functions. Therefore, we can map a colouring f to 0 if, after removing the trash colour, f depends on the first variable, and 1 if it depends on the second. This defines the mapping h. The only hard part is to prove that h is a homomorphism; for that it is necessary to look at ternary polymorphisms of $(\mathbf{K}_3, \mathbf{K}_4)$, since they determine the relational structure $\mathbf{F}_{\mathscr{M}}(\mathbf{T})$.

5.2 Reduction from Gap Label Cover

Let us continue with a more general reduction from Gap Label Cover. The framework that we present generalises the approach of, e.g., References [7, 26], where the authors proved NP-hardness of various fixed-template PCSPs using polymorphisms by reducing from Gap Label Cover.

Definition 5.6. The *Gap Label Cover* problem with parameters δ (*completeness*), ε (*soundness*), and N, denoted by $\mathrm{GLC}_{\delta,\varepsilon}(N)$, is a promise problem that, given an instance of $\mathrm{LC}(N)$,

- accepts if there is an assignment that satisfies at least δ-fraction of the given constraints, or
- rejects if no assignment satisfies more than ε-fraction of the given constraints.

The hardness of Gap Label Cover with perfect completeness (i.e., $\delta = 1$) and some soundness $\varepsilon < 1$ can be directly obtained from the PCP theorem of References [3, 4]. The soundness parameter can then be brought down to arbitrarily small $\varepsilon > 0$ using the parallel repetition theorem of Raz [84] at the cost of increasing N.

THEOREM 5.7 ([3, 4, 84]). *There exist constants $C_1, C_2 > 0$ such that for every $\varepsilon > 0$ and every $N \geq C_1 \varepsilon^{-C_2}$, $GLC_{1,\varepsilon}(N)$ is NP-hard.*

Fix some $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ and let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. As before, we can reduce Gap Label Cover to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ via Theorem 3.12(2) if we can ensure that the standard transformation from $\mathrm{GLC}_{1,\varepsilon}(N)$ to $\mathrm{PMC}_{\mathscr{M}}(N)$ (i.e., interpreting a Label Cover instance as a bipartite minor condition) is a valid reduction. The completeness is immediate. To prove that NO-answers are preserved, one typically uses some limitations of the structure (which may be non-trivial to obtain) of bipartite minor conditions satisfied in $\mathscr{M}$ and shows the contrapositive. More specifically, one proves that if a bipartite minor condition Σ is satisfied in $\mathscr{M}$, then some part of Σ containing more than ε-fraction of identities can be satisfied in projections, and hence the corresponding Label Cover instance has an assignment satisfying more than ε-fraction of constraints. To formulate a theorem characterising this approach, we use the following definition that captures NO-instances of Gap Label Cover in an algebraic language:

Definition 5.8. Let $\varepsilon > 0$. We say that a bipartite minor condition Σ is *ε-robust* if no ε-fraction of identities from Σ is trivial.

THEOREM 5.9. *There exist constants $C_1, C_2 > 0$ such that the following holds: If there exists an $\varepsilon > 0$ and $N \geq C_1\varepsilon^{-C_2}$ such that* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *does not satisfy any ε-robust bipartite minor condition involving symbols of arity at most N, then PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.*

PROOF. Let C_1 and C_2 be the same as in Theorem 5.7, so $\mathrm{GLC}_{1,\varepsilon}(N)$ is NP-hard for any ε and N satisfying the assumptions, and let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. We transform an instance of $\mathrm{GLC}_{1,\varepsilon}(N)$ to an instance of $\mathrm{PMC}_{\mathscr{M}}(N)$ in the usual way. Since label cover constraints are in 1-to-1 correspondence with identities in the bipartite minor condition, any NO-instance of $\mathrm{GLC}_{1,\varepsilon}(N)$ is transformed into a bipartite minor condition that is ε-robust and therefore fails in $\mathscr{M}$, i.e., this condition is a NO instance of $\mathrm{PMC}_{\mathscr{M}}(N)$. This shows soundness of the reduction, and completeness is obvious. The statement now follows from Theorem 3.12(2). □

Since for every $\varepsilon > 0$ there exists a suitable N in the above theorem, we can formulate the following useful weaker version of this theorem:

COROLLARY 5.10. *Let $\varepsilon > 0$, if* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *does not satisfy any ε-robust bipartite minor condition, then PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.*

The only general approach (that we are currently aware of) to verify that a minion $\mathscr{M}$ satisfies no ε-robust bipartite condition is a probabilistic method applied as follows: Suppose we can find a probability distribution on the set of pairs of arity-preserving mappings $\mathscr{M} \to \mathscr{P}_2$ so, for each minor identity

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

the following happens: If the identity is satisfied by functions $\zeta(f)$ and $\zeta(g)$ in $\mathscr{M}$, and we select $\tau, \tau' : \mathscr{M} \to \mathscr{P}_2$ according to the probability distribution, then the probability that the identity is satisfied by the projections $p_i = \tau(\zeta(f))$ and $p_j = \tau'(\zeta(g))$ (i.e., that $i = \pi(j)$) is at least ε. In this case, $\mathscr{M}$ satisfies no ε-robust bipartite minor condition. Indeed, if a bipartite minor condition $(\Sigma, \mathcal{U}, \mathcal{V})$ is satisfied in $\mathscr{M}$ through a map $\zeta : \mathcal{U} \cup \mathcal{V} \to \mathscr{M}$ and we randomly assign a projection to each function h in the image of ζ as above, then the probability that a single minor identity in Σ is satisfied is at least ε and it follows that the expected fraction of satisfied identities in Σ is at least ε. Consequently, some ε-fraction of identities in Σ is trivial.

The random procedure of assigning projections to function in $\mathscr{M}$ can be thought of as a generalised minion homomorphisms from $\mathscr{M}$ to $\mathscr{P}_2$. We will not study this concept in full generality, since, in applications, a very simple version has been sufficient so far. Namely, projections $\tau(f)$ as well as $\tau'(f)$ are chosen independently for each $f \in \mathscr{M}$ and uniformly from a subset of projections (which corresponds to a subset of coordinates). In this case, the probabilistic argument gives us the following lemma:

LEMMA 5.11. *Let $\mathscr{M}$ be a minion and let $C : \mathbb{N} \to \mathbb{N}$. Assume that there exists a mapping I assigning to each $h \in \mathscr{M}^{(n)}$ a subset of $[n]$ of size at most $C(n)$ such that for each $\pi : [m] \to [n]$ and each $g \in \mathscr{M}^{(m)}$, we have*

$$\pi(I(g(x_1, \dots, x_m)) \cap I(g(x_{\pi(1)}, \dots, x_{\pi(m)})) \neq \emptyset.$$

Then $\mathscr{M}$ satisfies no $(1/C(N)^2)$-robust bipartite minor condition involving symbols of arity at most N.

Proof. Let $(\Sigma, \mathcal{U}, \mathcal{V})$ be a bipartite minor condition that involves symbols of arity at most N and is satisfied in $\mathscr{M}$, which is witnessed by $\zeta : \mathcal{U} \cup \mathcal{V} \to \mathscr{M}$. To ease readability, we write $f^{\mathscr{M}} = \zeta(f)$ for each f. Let $\varepsilon = 1/C(N)^2$. We want to show that Σ is not ε-robust, i.e., we want to find an assignment $\rho : \mathcal{U} \cup \mathcal{V} \to \mathscr{P}_2$ that satisfies at least ε-fraction of identities in Σ.

Let us choose such an assignment by choosing a coordinate $i \in I(f^{\mathscr{M}})$ uniformly at random, and setting $\rho(f) = p_i$. We claim that the probability that any single identity in Σ is satisfied is at least ε. Indeed, if we consider an identity

$$f(x_1, \ldots, x_n) \approx g(x_{\pi(1)}, \ldots, x_{\pi(m)}), \tag{5.1}$$

in Σ, then we get that $I(f^{\mathscr{M}}) \cap \pi(I(g^{\mathscr{M}})) \neq \emptyset$, and consequently, there is a choice of $i \in I(f^{\mathscr{M}})$ and $j \in I(g^{\mathscr{M}})$ such that $i = \pi(j)$, i.e., $f = p_i$ and $g = p_j$ satisfies Equation (5.1). This means that the probability is at least

$$1/\Big(|I(f^{\mathscr{M}})| \cdot |I(g^{\mathscr{M}})|\Big) \geq 1/C(n)C(m) \geq 1/C(N)^2.$$

It follows that the expected fraction of identities that gets satisfied by ρ is at least ε, which means that there is an assignment of projections that satisfies at least ε-fraction of identities in Σ. □

Directly from the above and Corollary 5.10, we get the following:

Corollary 5.12. *Let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ and let C be a constant such that there exists a mapping I assigning to each $h \in \mathscr{M}^{(n)}$ a subset of $[n]$ of size at most C such that for each $\pi : [m] \to [n]$ and each $g \in \mathscr{M}^{(m)}$, we have*

$$\pi(I(g(x_1, \ldots, x_m)) \cap I(g(x_{\pi(1)}, \ldots, x_{\pi(m)})) \neq \emptyset.$$

Then PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.

We will now explain some known hardness results that can be obtained using the above corollary. Possibly the simplest way to choose the set $I(f)$ of coordinates for a function f is to choose only coordinates that can influence the value of f. This is formalised in the following definition:

Definition 5.13. Let $f : A^n \to B$, a coordinate $i \in [n]$ is called *essential* if there exist $a_1, \ldots, a_n$ and b_i in A such that

$$f(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, b_i, a_{i+1}, \ldots, a_n).$$

A minion $\mathscr{N}$ on (A, B), where A and/or B can be infinite, is said to have *essential arity at most k*, if each function $f \in \mathscr{N}$ has at most k essential variables. We say that $\mathscr{N}$ has bounded essential arity if it has essential arity at most k for some k.

The following is a generalisation of Reference [7, Theorem 4.7]:

Proposition 5.14. *Let $(\mathbf{A}, \mathbf{B})$ be a PCSP template and let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. Assume that there exists a minion homomorphism $\xi : \mathscr{M} \to \mathscr{N}$ for some minion $\mathscr{N}$, possibly on infinite sets, which has bounded essential arity and does not contain a constant function (i.e., a function without essential variables). Then PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.*

Proof. Let C be the bound on the essential arity of $\mathscr{N}$. For $f \in \mathscr{M}$, we set $I(f)$ to be the set of all essential coordinates of $\xi(f)$. Clearly, $1 \leq |I(f)| \leq C$. To apply Corollary 5.12, we only need to show that if

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

for all $x_1, \dots, x_n \in A$, then $I(f) \cap \pi(I(g)) \neq \emptyset$. Indeed, if $i \in [m]$ is an essential coordinate of $\xi(f)$, then there are two tuples $a_1, \dots, a_n$ and $b_1, \dots, b_n$ such that $a_i \neq b_i$, $a_{i'} = b_{i'}$ for all $i' \neq i$ and $\xi(f)(a_1, \dots, a_n) \neq \xi(f)(b_1, \dots, b_n)$. The last disequality together with the above identity (which is preserved by ξ) gives that

$$\xi(g)(a_{\pi(1)}, \dots, a_{\pi(n)}) \neq \xi(g)(b_{\pi(1)}, \dots, b_{\pi(n)}).$$

The two tuples of arguments differ only on coordinates j' with $\pi(j') = i$, therefore $\xi(g)$ has to depend essentially on at least one coordinate from $\pi^{-1}(i)$. This shows that $i \in \pi(I(g))$, and therefore $I(f) \subseteq \pi(I(g))$. We get the claim, since $I(f) \neq \emptyset$ by assumption. □

The following technical notion is a slight strengthening of one that was used in Reference [29] as a sufficient condition for NP-hardness of some Boolean PCSPs:

Definition 5.15. Let $C > 0$ be a constant. A minion $\mathscr{M} \subseteq \mathscr{O}(\{0,1\})$ is said to be *strongly C-fixing* if for each $f \in \mathscr{M}$ there exists a (*fixing*) set $I_f \subseteq [\mathrm{ar}(f)]$, $|I_f| \leq C$ such that $f(x_1, \dots, x_{\mathrm{ar}(f)}) = 0$ whenever $x_i = 0$ for all $i \in I$, and similarly, $f(x_1, \dots, x_{\mathrm{ar}(f)}) = 1$ whenever $x_i = 1$ for all $i \in I$.

We remark that a strongly C-fixing minion does not need to have bounded essential arity. The following proposition is a generalisation of Reference [29, Theorem 5.1]:

Proposition 5.16. *Let* $(\mathbf{A}, \mathbf{B})$ *be a PCSP template and let* $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. *Assume that there exists a minion homomorphism* $\xi : \mathscr{M} \to \mathscr{N}$ *for some minion* $\mathscr{N} \subseteq \mathscr{O}(\{0,1\})$ *that is strongly* C*-fixing for some* $C > 0$. *Then PCSP*$(\mathbf{A}, \mathbf{B})$ *is NP-hard.*

Proof. We set $I(f)$ to be some fixing set of $\xi(f)$ of size at most C. Observe that no function $f \in \mathscr{N}$ can have two disjoint fixing sets: If I and J would be disjoint and fixing, then we would get that for a tuple $(x_1, \dots, x_n)$ such that $x_i = 0$ for all $i \in I$ and $x_j = 1$ for all $j \in J$, we would get that $f(x_1, \dots, x_n)$ is both 0 and 1. However, a π-image of a fixing set I of g is a fixing set for the minor $g(x_{\pi(1)}, \dots, x_{\pi(m)})$. This shows that if f and g satisfy

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

and consequently their ξ-images satisfy the same identity, then $I(f) \cap \pi(I(g)) \neq \emptyset$. Hence, Corollary 5.12 applies. □

5.3 Reduction from Multilayered Label Cover

NP-hardness of Gap Label Cover is just one of the consequences of the PCP theorem, although arguably the most prevalent one. Other variants of GLC can also be used in proving NP-hardness of some PCSP. In this section, we describe in algebraic terms a reduction from the so-called *Layered Label Cover* (or *Multilayered PCP*) that has been used in Reference [47] to prove hardness of approximate hypegraph colouring, i.e., PCSP$(\mathbf{H}_2, \mathbf{H}_k)$. We also briefly comment on how Layered Label Cover is applied in this case. Gap Layered Label Cover was introduced in Reference [45].

The **Layered Label Cover (LLC)** is a generalisation of LC from bipartite graphs to L-partite graphs. The partite sets are referred to as *layers.*

Definition 5.17 (Layered Label Cover). Fix positive integers N and $L \geq 2$. We define LLC(L, N) as the following decision problem. The input is a tuple

$$((V_i, r_i)_{1 \leq i \leq L}, (E_{ij}, \Pi_{ij})_{1 \leq i < j \leq L})$$

where

- Each $G_{ij} = (V_i, V_j; E_{ij})$ is a bipartite graph,
- $r_i \leq N$ are positive integers, and
- each Π_{ij} is a family of maps $\pi_{ij,e} : [r_j] \to [r_i]$, one for each $e \in E_{ij}$.

The goal is to decide whether there is a labelling of vertices from $V_1, \dots, V_L$ with labels from $[r_1], \dots, [r_L]$, respectively, such that if $(u, v) \in E_{ij}$, then the label of v is mapped by $\pi_{ij,(u,v)}$ to the label of u.

Note that the LLC is a CSP, where edges $(u, v) \in E_{ij}$ correspond to constraints between the layers i and j.

Just like $\mathrm{LC}(N) = \mathrm{LLC}(2, N)$ is essentially the same problem as $\mathrm{MC}(N)$ (recall Subsection 3.1), the L-layered version $\mathrm{LLC}(L, N)$ is essentially the same problem as $\mathrm{LMC}(L, N)$, the problem of deciding triviality of *L-layered minor conditions*. To be precise, for pairwise disjoint sets $\mathcal{V}_1, \dots, \mathcal{V}_L$ of function symbols, an L-layered minor condition is a tuple $(\Sigma, \mathcal{V}_1, \dots, \mathcal{V}_L)$ where $\mathcal{V}_i$ are disjoint sets of function symbols, and Σ is a set of identities of the form $f(x_1, \dots, x_{r_i}) = g(x_{\pi(1)}, \dots, x_{\pi(r_j)})$ where $i < j$, $f \in \mathcal{V}_i$ and $g \in \mathcal{V}_j$.

For a minion $\mathscr{M}$, one can naturally define $\mathrm{PLMC}_{\mathscr{M}}(L, N)$, the promise version of $\mathrm{LMC}(L, N)$, in the same way as $\mathrm{PMC}_{\mathscr{M}}(N)$ is obtained from $\mathrm{MC}(N)$, i.e., the YES-instances of $\mathrm{LMC}(L, N)$ are all the trivial L-layered minor conditions Σ involving symbols of arity at most N, and NO-instances those L-layered minor conditions Σ involving symbols of arity at most N that are not satisfied in $\mathscr{M}$. Moreover, if $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$, then, for any fixed L and N, $\mathrm{PLMC}_{\mathscr{M}}(L, N)$ can be reduced to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ in log-space in the same way as in the proof of Theorem 3.12(2) (see also Section 3.3).

We will exploit these observations in a similar way as in the previous subsection—by exploring when a known NP-hard gap version of LLC can be naturally reduced to $\mathrm{PLMC}_{\mathscr{M}}(L, N)$. An important gain of using more layers is a certain kind of density of LLC instances that can be required while preserving hardness.

Definition 5.18. An instance of $\mathrm{LLC}(L, N)$ is called *weakly dense* if for any $1 < m < L$, any m layers $i_1 < \cdots < i_m$, and any sets $S_j \subseteq V_{i_j}$ such that $|S_j| \geq 2|V_{i_j}|/m$ for every $1 \leq j \leq m$, there exist $1 \leq j < j' \leq m$ such that $|E_{i_j i_{j'}} \cap (S_j \times S_{j'})| \geq |E_{i_j i_{j'}}|/m^2$.

We are ready to state a gap version of Layered Label Cover from Reference [45].

Definition 5.19. The *Gap Layered Label Cover* problem with parameters ε, L, and N denoted by $\mathrm{GLLC}_\varepsilon(L, N)$, is a promise problem in which, given a weakly dense instance of $\mathrm{LLC}(L, N)$, one needs to

- accept if there is an assignment that satisfies all the constraints, or
- reject if for every $1 \leq i < j \leq L$ no assignment satisfies more than ε-fraction of the constraints between the layers i and j.

THEOREM 5.20 [45, THEOREM 4.2]. *There exist constants $C_1, C_2 > 0$ such that for every $1 \geq \varepsilon > 0$, every $L \geq 2$, and every $N \geq C_1 \varepsilon^{-C_2 L}$ the problem $\mathrm{GLLC}_\varepsilon(L, N)$ is NP-hard.*

An analogue of Theorem 5.9, which we state now, allows us to break a polymorphism minion into finitely many sets that do not need to be minions and check the "no ε-robust minor condition" property for each of the pieces separately (note that Definition 3.2 of satisfaction of a minor condition in $\mathscr{M}$ makes sense for an arbitrary subset of $\mathscr{O}(A, B)$).

THEOREM 5.21. *Let $(\mathbf{A}, \mathbf{B})$ be a PCSP template and let $\varepsilon : \mathbb{N} \to \mathbb{R}$ be a function such that $\varepsilon(N) \in \Omega(N^{-K})$ for each $K > 0$. Assume that $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is a union of finitely many sets $\mathscr{M}_1, \dots, \mathscr{M}_k$,* *none of which satisfies (for any N) any $\varepsilon(N)$-robust bipartite minor condition involving symbols of arity at most N. Then PCSP$(\mathbf{A},\mathbf{B})$ is NP-hard.*

Proof. We show that, with an appropriate choice of ε', L, and N, the usual transformation of a $\mathrm{GLLC}_{\varepsilon'}(L,N)$ to a $\mathrm{PLMC}_{\mathscr{M}}(L,N)$ is a reduction. Since $\mathrm{GLLC}_{\varepsilon'}(L,N)$ is NP-hard by Theorem 5.21 (for a large enough N) and $\mathrm{PLMC}_{\mathscr{M}}(L,N)$ reduces to $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$, the claim will follow.

It is straightforward that the yes-instances are mapped to yes-instances with any choice of parameters. The non-trivial part is to show that, with a suitable choice of the parameters, no-instances are mapped to no-instances. We set $L = 2k^2$ and choose N, ε' so

$$\varepsilon' = \varepsilon(N)/(4k^2) \text{ and } N \geq C_1(\varepsilon')^{-C_2 L},$$

which is possible as $\varepsilon(N) = \Omega(N^{-K})$ for $K < 1/C_2L$.

We verify the contrapositive. Consider a weakly dense instance $I = ((V_i, r_i)_{i\leq L}, (E_{ij}, \Pi_{ij})_{i<j})$ of $\mathrm{GLLC}_{\varepsilon'}(L,N)$ that is not mapped to a no-instance Σ (over the sets of symbols $(\mathcal{V}_i)_{1\leq i\leq L}$) of $\mathrm{PLMC}_{\mathscr{M}}(L,N)$, i.e., there exists a mapping ζ from $\cup_{i=1}^{L}\mathcal{V}_i$ to $\mathscr{M}$ that witnesses that $\mathscr{M}$ satisfies Σ.

We colour each symbol f in $\mathcal{V}_i$ by some index c such that $\zeta(f) \in \mathscr{M}_c$. Next, we colour each layer $\mathcal{V}_i$ by the most popular colour among its members. Since the number of colours is at most k and the number of layers is $2k^2$, at least $m = 2k$ layers $i_1, \dots, i_m$ received the same colour, say c. Finally, let S_j (where $j = 1, \dots, m$) denote the set of all elements of $\mathcal{V}_{i_j}$ with colour c.

We have $|S_j| \geq |V_{i_j}|/k = 2|V_{i_j}|/m$, therefore, as I is weakly dense, there exist $j < j'$ such that at least $1/m^2$-fraction of identities in Σ, which are between $\mathcal{V}_j$ and $\mathcal{V}_{j'}$, is between S_j and $S_{j'}$. The assignment ζ witnesses that the system of all identities between S_j and $S_{j'}$ is satisfied in $\mathscr{M}_c$. Since $\mathscr{M}_c$ does not satisfy any $\varepsilon(N)$-robust condition with symbols of arity at most N, some $\varepsilon(N)$-fraction of identities between S_j and $S_{j'}$ is trivial. Since $\varepsilon' = \varepsilon(N)/(4k^2) = \varepsilon(N)/m^2$, it follows that some ε'-fraction of identities between $\mathcal{V}_{i_j}$ and $\mathcal{V}_{i_{j'}}$ is trivial, which for the original instance I means that at least ε'-fraction of the constraints between V_{i_j} and $V_{i_{j'}}$ is satisfied, so I is not a no-instance of $\mathrm{GLLC}_{\varepsilon'}(L,N)$. □

The assumption that $\mathscr{M}_i$ does not satisfy any $\varepsilon(N)$-robust bipartite minor condition may be verified by a probabilistic argument as in the previous section. We remark that Lemma 5.11 applies to sets $\mathscr{M}$ of functions that are not minions, e.g., sets $\mathscr{M}_i$ from the previous theorem.

Hardness of approximate hypergraph colouring. In the rest of this subsection, we will sketch a proof of NP-hardness of approximate graph colouring (see Example 2.11). This was proved in Reference [47], and in our sketch, we will reuse some of the combinatorial arguments from that paper. The theorem can be stated as follows:

Theorem 5.22 ([47]). *For any $k \geq 2$, PCSP$(\mathbf{H}_2, \mathbf{H}_k)$ is NP-hard.*

Our proof loosely follows the one in Reference [47] with the main difference being the usage of our general theory. In particular, we use the above universal reduction from LLC to PLMC instead of an ad hoc reduction from LLC to hypergraph colouring used in Reference [47].

The combinatorial core of the proof is a strengthening Lovász's theorem on the chromatic number of Kneser graphs. To state the theorem in the algebraic language, recall that $\mathrm{Ham}(\mathbf{t})$ denotes the Hamming weight of a tuple $\mathbf{t} \in \{0,1\}^n$. We call two tuples $\mathbf{u}, \mathbf{v} \in \{0,1\}^n$ *disjoint* if $u_i = 0$ or $v_i = 0$ (or both) for every $i \in [n]$.

Theorem 5.23 ([75]). *Let $f : \{0,1\}^N \to \{0,\dots,k-1\}$ and $s = \lfloor (N+1-k)/2 \rfloor$. Then there exist disjoint tuples $\mathbf{u}^f, \mathbf{v}^f \in \{0,1\}^N$ such that $\mathrm{Ham}(\mathbf{u}^f) = \mathrm{Ham}(\mathbf{v}^f) = s$ and $f(\mathbf{u}^f) = f(\mathbf{v}^f)$.*

This theorem has an immediate consequence for polymorphisms of $(\mathbf{H}_2, \mathbf{H}_k)$. Namely, it implies that, for every polymorphism $f : \mathbf{H}_2^N \to \mathbf{H}_k$, there exists $c \in \{0,\dots,k-1\}$ and a *c-avoiding set A* of coordinates of size at most k, by which we mean that $f(\mathbf{w}) \neq c$ for every tuple $\mathbf{w} \in \{0,1\}^N$ such that $w_i = 1$ for every $i \in A$. Indeed, take $c = f(\mathbf{u}^f) = f(\mathbf{v}^f)$ and

$$A = \left\{ i \in [N] \mid u_i^f = v_i^f = 0 \right\}.$$

Since $\mathbf{w}$ is such that $w_i = 1$ for every $i \in A$, the matrix with rows $\mathbf{u}^f, \mathbf{v}^f, \mathbf{w}$ has all columns in NAE_2, and therefore f applied to the rows of this matrix gives a tuple in NAE_k. As $c = f(\mathbf{u}^f) = f(\mathbf{v}^f)$, we get $f(\mathbf{w}) \neq c$ for every such a tuple. The size of A is indeed at most $|N - 2\lfloor (N+1-k)/2 \rfloor| \leq k$.

These c-avoiding sets have similar properties as fixing sets (see Definition 5.15): The π-image of a c-avoiding set for a function is clearly c-avoiding for the corresponding minor. However, a function can have two disjoint c-avoiding sets (indeed, every set can be c-avoiding if this colour is never used by f), so exactly the same argument does not work.

This issue can be resolved by strengthening Theorem 5.23 from Kneser graphs to Schrijver graphs [87]. A counting argument then gives the following:

Lemma 5.24 [47, Lemma 2.2]. *Let $f : \{0,1\}^N \to \{0,\dots,k-1\}$ and $s = \lfloor (N+1-k)/2 \rfloor$. There exists $c^f \in \{0,\dots,k-1\}$ such that*

- *there exist disjoint tuples $\mathbf{u}^f, \mathbf{v}^f \in \{0,1\}^N$ such that $\mathrm{Ham}(\mathbf{u}^f) = \mathrm{Ham}(\mathbf{v}^f) = s$ and $f(\mathbf{u}^f) = f(\mathbf{v}^f) = c^f$ and*
- *the fraction of elements of $\{\mathbf{u} : \mathrm{Ham}(\mathbf{u}) = s\}$ such that $f(\mathbf{u}) = c^f$ is $\Omega(N^{-k-1})$ (where the constant hidden in Ω depends only on k).*

The first item gives us a small (of size at most k) c^f-avoiding set for every function f. The second item together with another counting argument (see Reference [47, Claim 4.5]) implies that there is no collection of pairwise disjoint small c^f-avoiding sets of size bigger than $C_3 \log N$, where the constant C_3 depends only on k. Now, we can finish the proof using Theorem 5.21 and Lemma 5.11. For $c = 1,\dots,k$, we define $\mathscr{M}_c = \{f \in \mathscr{M} \mid c^f = c\}$ and, for $f \in \mathscr{M}_c$, we define $I(f)$ as the union of a maximal collection of pairwise disjoint small c^f-avoiding sets. For any $\pi : [m] \to [n]$ and an m-ary polymorphism g such that both $g(x_1,\dots,x_m)$ and $f(x_1,\dots,x_n) = g(x_{\pi(1)},\dots,x_{\pi(m)})$ are in $\mathscr{M}_c$, we have that $\pi(I(g)) \cap I(f) \neq \emptyset$, since the π-image of a c-avoiding set (and hence of $I(g)$) is a c-avoiding set for f. Moreover, $|I(h)| \leq C_3 \log N$ if $\mathrm{ar}(h) \leq N$. By Lemma 5.11, no $\mathscr{M}_c$ satisfies any $(1/C_3^2 \log^2 N)$-robust bipartite minor condition, and then $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ is NP-hard by Theorem 5.21.

6 HARDNESS FROM OTHER PCSPS

In this section, we derive hardness results by reductions from other PCSPs by directly applying results from Sections 3 and 4. We give a concise algebraic characterisation of PCSPs that admit a reduction from approximate hypergraph colouring, i.e., from $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ for some k (which is NP-hard by Theorem 5.22), and we apply this characterisation to special cases of approximate graph colouring and graph homomorphism. We give similar characterisation results for the existence of a reduction from approximate graph colouring or graph homomorphism problems (which are currently not known to be NP-hard in full generality).

6.1 Hardness from Approximate Hypergraph Colouring

As a starting point for our reductions, we can use Theorem 5.22. Following our approach, the first step in a reduction from $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is a reduction from $\mathrm{PCSP}(\mathbf{H}_2, \mathbf{H}_k)$ to $\mathrm{PMC}_{\mathscr{H}_k}(6)$ where $\mathscr{H}_k = \mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_k)$, given by Theorem 3.12(2). To analyse when we can continue further, it is useful to understand which bipartite minor conditions are not satisfied in $\mathscr{H}_k$.

We explained in Example 3.5 that the following condition is not satisfied in $\mathscr{H}_k$ for any $k \geq 2$:

$$\begin{aligned} t(x,y) &\approx o(x,x,y,y,y,x), \\ t(x,y) &\approx o(x,y,x,y,x,y), \\ t(x,y) &\approx o(y,x,x,x,y,y). \end{aligned} \tag{$\spadesuit$}$$

We prove below that having polymorphisms satisfying this condition is the only obstacle for a reduction from approximate hypergraph colouring to a given PCSP template.

Definition 6.1. An *Olšák function* is a 6-ary function o that satisfies

$$o(x,x,y,y,y,x) \approx o(x,y,x,y,x,y) \approx o(y,x,x,x,y,y).$$

The above identities appeared in Olšák's paper [78]. The algebraic significance of these identities is that they give the weakest non-trivial Maltsev condition for all idempotent algebras (also infinite ones), though we will not use this fact in our article. As it is usual in the literature, we call a polymorphism of some given template an *Olšák polymorphism* if it is an Olšák function.

Theorem 6.2. *Let $\mathscr{M}$ be a minion. The following are equivalent:*

(1) There exists $K \geq 2$ and a minion homomorphism $\xi : \mathscr{M} \to \mathscr{H}_K$;
(2) $\mathscr{M}$ does not contain an Olšák function.

Proof. Clearly, if $\mathscr{M}$ contains an Olšák function, then its image under a minion homomorphism would also be an Olšák function. Since $\mathscr{H}_K$ does not contain such a function, this proves the implication (1) → (2).

For the other implication assume that $\mathscr{M}$ does not contain an Olšák function. Consider the 3-uniform hypergraph $\mathbf{F} = \mathbf{F}_{\mathscr{M}}(\mathbf{H}_2)$ obtained as the free structure of $\mathscr{M}$ generated by $\mathbf{H}_2$. Note that the vertices of $\mathbf{F}$ are binary functions in $\mathscr{M}$, and three such vertices f, g, h are connected by a hyperedge if there is a 6-ary function o' such that

$$\begin{aligned} f(x,y) &\approx o'(x,x,y,y,y,x), \\ g(x,y) &\approx o'(x,y,x,y,x,y), \\ h(x,y) &\approx o'(y,x,x,x,y,y). \end{aligned}$$

Since $\mathscr{M}$ does not contain an Olšák function, we get that $\mathbf{F}$ does not contain hyperedges (f,g,h) with $f = g = h$, which is enough to guarantee that it is colourable with $K = |F| = |\mathscr{M}^{(2)}|$ colours (note that $|\mathscr{M}^{(2)}|$ is finite). In other words, there is a homomorphism from $\mathbf{F}$ to $\mathbf{H}_K$, and we have a minion homomorphism from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{H}_2, \mathbf{H}_K) = \mathscr{H}_K$ by Lemma 4.4. □

Corollary 6.3. *For every finite template $(\mathbf{A}, \mathbf{B})$ that does not have an Olšák polymorphism, PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.*

Proof. Let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. From the previous theorem, we know that there is a minion homomorphism from $\mathscr{M}$ to $\mathscr{H}_K$ for some K. Therefore, the claim follows from Theorem 3.1 and Theorem 5.22. □

We give two applications of the above corollary in the next subsection. Note that the absence of Olšák polymorphism is a useful sufficient condition for hardness of PCSPs, but by no means

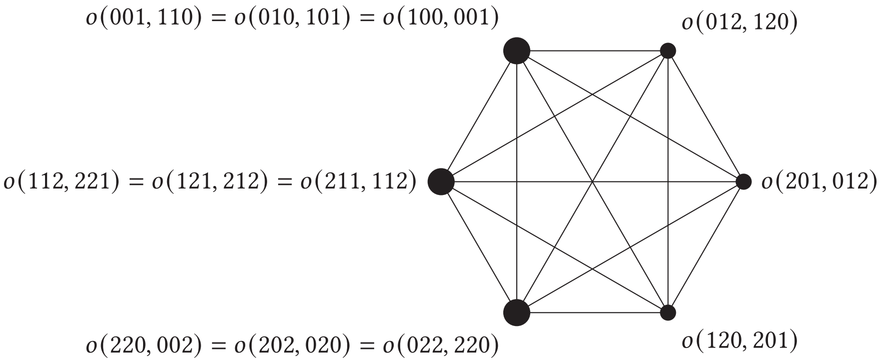
$o(001, 110) = o(010, 101) = o(100, 001)$
$o(012, 120)$
$o(112, 221) = o(121, 212) = o(211, 112)$
$o(201, 012)$
$o(220, 002) = o(202, 020) = o(022, 220)$
$o(120, 201)$

Fig. 2. A 6-clique in the indicator graph $\mathbf{I}_3$ of an Olšák function on $\mathbf{K}_3$.

a universal tool for this, as witnessed by, e.g., the hardness results of Reference [68] (see also Proposition 10.1).

6.2 Hardness of Approximate Graph Colouring and Homomorphism

In this section, we settle some of the open problems mentioned in Examples 2.9 and 2.10. We start by considering approximate graph colouring $\mathrm{PCSP}(\mathbf{K}_k, \mathbf{K}_c)$. Recall that the strongest known NP-hardness results (without additional assumptions) for this problem are the cases when $k \geq 3$ is arbitrary and $c \leq 2k - 2$ [26] and when $c \leq 2^{\Omega(k^{1/3})}$ and k is large enough [59].

We prove that distinguishing between k-colourable graphs and those not even $(2k-1)$-colourable is NP-hard for all $k \geq 3$. In particular, we prove that colouring a 3-colourable graph with 5 colours is NP-hard. This improves the results of References [26, 64], which are the best-known bounds for small k.

By Corollary 6.3, it is enough to show that no polymorphism from $\mathbf{K}_k$ to $\mathbf{K}_{2k-1}$ is an Olšák function.

Lemma 6.4. $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k-1})$ *does not contain an Olšák function.*

Proof. To show that $(\mathbf{K}_k, \mathbf{K}_{2k-1})$ does not have an Olšák polymorphism, we consider the indicator structure for the Olšák minor condition; more precisely, the structure $\mathbf{I}_\Sigma(\mathbf{K}_k)$ where Σ is the bipartite condition (♤). For brevity, we write $\mathbf{I}_k$ instead of $\mathbf{I}_\Sigma(\mathbf{K}_k)$. The structure $\mathbf{I}_k$ is constructed by taking a union of the 6th power of $\mathbf{K}_k$ (whose colourings correspond to the Olšák polymorphism o) and the 2nd power of $\mathbf{K}_k$ (whose colourings correspond to the polymorphism t, which is defined as a minor of o) while identifying vertices of $\mathbf{K}_k^2$ with vertices of $\mathbf{K}_k^6$ according to the identities (♤). This construction identitfies a vertex (x, y) of $\mathbf{K}_2^2$ with the vertices (x, x, y, y, y, x), (x, y, x, y, x, y), and (y, x, x, x, y, y) of $\mathbf{K}^6$ (note that these are the tuples that get the same colour under an Olšák polymorphism).

We claim that $\mathbf{I}_k$ contains a $2k$-clique. Namely, for all $i \in K_k$, we consider:

$$a_i = (i, i+1, i+2, i+1, i+2, i) \text{ and } b_i = (i, i+1),$$

where addition is modulo k. Note that the vertices b_i are involved in the gluing that produced the graph, therefore they can be also represented by tuples $(i, i+1, i, i+1, i, i+1)$, $(i, i, i+1, i+1, i+1, i)$, and $(i+1, i, i, i, i+1, i+1)$. Figure 2 depicts this clique in $\mathbf{I}_3$. We claim that all pairs of these vertices are connected by an edge. Clearly, for any $i \neq j$, there is an edge between a_i and a_j, as well as between b_i and b_j.

For edges between a_i and b_j, we consider the following cases:

- $j \notin \{i, i+1\}$. Then also $j+1 \notin \{i+1, i+2\}$, so there is an edge between b_j and a_i, since b_j is represented by the tuple $(j, j, j+1, j+1, j+1, j)$.
- $j = i$. There is an edge between a_i and b_j, since b_j is represented by the tuple $(j+1, j, j, j, j+1, j+1)$.
- $j = i+1$. There is an edge between a_i and b_j, since b_j is represented by the tuple $(j, j+1, j, j+1, j, j+1)$.

Altogether, we get that $\{a_i, b_i \mid i \in K_k\}$ is a clique of size $2k$, and therefore $\mathbf{I}_k$ has no $(2k-1)$-colouring. Finally, we get that $\mathbf{I}_k \not\to \mathbf{K}_{2k-1}$ implies that $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k-1})$ does not contain an Olšák function, since an Olšák polymorphism $\mathbf{K}_k^6 \to \mathbf{K}_{2k-1}$ would give a well-defined colouring of $\mathbf{I}_k$ (see also Remark 3.17). □

As a direct corollary of the above lemma and Corollary 6.3, we get the following:

Theorem 6.5. *Deciding whether a given graph is k-colourable or not even $(2k-1)$-colourable is NP-hard for any $k \geq 3$.*

We remark that Reference [38] contains a more direct proof of Theorem 6.5, which mimics the proofs from Section 3 without using the theory that we developed in Section 4.

We also remark that the presented method does not work for $c \geq 2k$ because $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k})$, and therefore also $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$, contains an Olšák polymorphism. For a detailed discussion, see Proposition 10.1.

We now consider the problem of distinguishing whether a given graph homomorphically maps to $\mathbf{C}_5$ (a 5-cycle) or it is not even 3-colourable. This problem (mentioned above in Example 2.10) was suggested as an intriguing open case in Reference [29]. This problem can be understood as the "other" relaxation of 3-colourability of graphs. For example, in approximate graph colouring, we may be interested whether a graph is 3-colourable or not 4-colourable, i.e., we relax $(\mathbf{K}_3, \mathbf{K}_3)$ to $(\mathbf{K}_3, \mathbf{K}_4)$. If we want a different relaxation, then we change the first $\mathbf{K}_3$ into some graph $\mathbf{G}$ that is 3-colourable. The odd cycles are then a natural choice, since any graph that is not 2-colourable contains an odd cycle. Therefore, if we could prove that $\mathrm{PCSP}(\mathbf{C}_{2k+1}, \mathbf{K}_3)$ is NP-hard, then we would have a complete picture for problems of the form $\mathrm{PCSP}(\mathbf{G}, \mathbf{K}_3)$. We further discuss problems of the form $\mathrm{PCSP}(\mathbf{G}, \mathbf{H})$ for two graphs $\mathbf{G}$ and $\mathbf{H}$ in Section 6.3.

Lemma 6.6. $\mathrm{Pol}(\mathbf{C}_5, \mathbf{K}_3)$ *does not contain an Olšák function.*

Proof. Suppose that $\mathrm{Pol}(\mathbf{C}_5, \mathbf{K}_3)$ contains an Olšák function o, which can be viewed as a 3-colouring of $\mathbf{C}_5^6$. Assume that the vertices of $\mathbf{C}_5$ are 0,1,2,3,4, in this cyclic order. Consider the 6-tuples $b_i = (i+1, i, i, i, i+1, i+1), 0 \leq i \leq 4$, where the addition is modulo 5. These tuples form a 5-cycle in $\mathbf{C}_5^6$. Since there is only one 3-colouring of $\mathbf{C}_5$ up to automorphisms of $\mathbf{C}_5$ and $\mathbf{K}_3$, we can assume without loss of generality that $o(b_0) = 1, o(b_1) = 0, o(b_2) = 1, o(b_3) = 2, o(b_4) = 0$.

It is easy to check that the graph in Figure 3 is a subgraph of $\mathbf{C}_5^6$, with each node assigned a set of colours, i.e., elements of $\mathbf{K}_3$. Unique colours (in columns 1 and 4) correspond to the values of the assumed operation o on these tuples; note that these colours are correct, because o is assumed to be an Olšák function. The two-element lists of colours simply indicate the possible values for the corresponding tuples on the basis that each such tuple has a neighbour whose colour is known. Now, the two middle vertices in the middle row must be assigned different colours by o, but neither option extends to a full proper colouring of this graph. □

Theorem 6.7. *PCSP$(\mathbf{C}_5, \mathbf{K}_3)$ is NP-hard.*

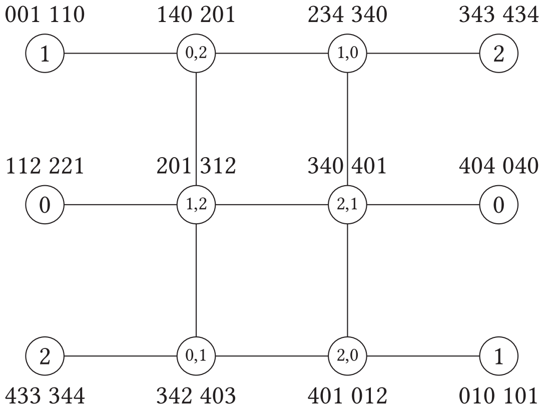

Fig. 3. No Olšák polymorphism for $(\mathbf{C}_5, \mathbf{K}_3)$.

6.3 Implications of Hardness of Approximate Graph Colouring

We proved above that the hardness of approximate hypergraph colouring implies hardness of any PCSPs satisfying an algebraic condition (namely, no Olšák polymorphism). In fact, this is one case of a more general pattern for such implications. To illustrate this, we now describe similar implications for the (yet unproved) hardness of approximate graph colouring and graph homomorphism problems. Recall Examples 2.9 and 2.10 and the conjectures mentioned there.

For the case of approximate graph colouring, the so-called Siggers functions play a role similar to that played by Olšák functions in relation to hypergraphs. The original Siggers functions appeared in Reference [88], but now there are several related versions of such functions (which are usually all called Siggers functions). We will use the one using 3-variable identities that appeared, e.g., in Reference [11].

Definition 6.8. A *Siggers function* is a 6-ary function s that satisfies

$$s(x, y, x, z, y, z) \approx s(y, x, z, x, z, y).$$

Again, a polymorphism of some given template that is a Siggers function is called *Siggers polymorphism.*

We remark that, for problems CSP$(\mathbf{A})$ with finite $\mathbf{A}$, the different versions of a Siggers function are equivalent in the sense that if Pol$(\mathbf{A})$ has one of them, then it has all of them. Moreover, for CSPs, this is also equivalent to having an Olšák polymorphism [78] and this property exactly characterises tractable CSPs [36, 92, 93] (under P ≠ NP). However, Olšák and Siggers can be shown not to be equivalent for PCSPs.

Just like the definition of an Olšák function relates to the six tuples in $\mathbf{H}_2$, the definition of a Siggers polymorphism relates to the six edges of $\mathbf{K}_3$ (viewed as a directed graph). This motivates the following theorem:

Theorem 6.9. *The following are equivalent:*

(1) For every finite template $(\mathbf{A}, \mathbf{B})$ without a Siggers polymorphism, PCSP$(\mathbf{A}, \mathbf{B})$ is NP-hard.
(2) PCSP$(\mathbf{K}_k, \mathbf{K}_c)$ is NP-hard for all $c \geq k \geq 3$.
(3) PCSP$(\mathbf{K}_3, \mathbf{K}_c)$ is NP-hard for each $c \geq 3$.

Proof. To prove (1) → (2), it is enough to show that $(\mathbf{K}_k, \mathbf{K}_c)$ has no Siggers polymorphisms for any $k \geq 3$. That is true, since any such polymorphism would force a loop in $\mathbf{K}_c$ on the vertex $s(0,1,0,2,1,2) = s(1,0,2,0,2,1)$.

The implication (2) → (3) is trivial.

For (3) → (1), the proof is similar to that of Theorem 6.2. Assume that a template $(\mathbf{A}, \mathbf{B})$ has no Siggers polymorphism, and let $\mathscr{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$. Consider the free structure $\mathbf{F} = \mathbf{F}_{\mathscr{M}}(\mathbf{K}_3)$. Then $\mathbf{F}$ is a graph whose vertices are the ternary polymorphisms of $(\mathbf{A}, \mathbf{B})$, and (f, g) is an edge if there is a 6-ary $e \in \mathscr{M}$ such that

$$f(x,y,z) = e(x,y,x,z,y,z),$$
$$g(x,y,z) = e(y,x,z,x,z,y)$$

for all $x, y, z \in A$. Clearly, this graph is loopless, since a loop would correspond to a Siggers polymorphism. It is then c-colourable where $c = |F|$. So, we have a homomorphism from $\mathbf{F}$ to $\mathbf{K}_c$, which by Lemma 4.4 implies that there is a minion homomorphism from $\mathscr{M}$ to $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_c)$. The result now follows from Theorem 3.1. □

We now generalise the above to the case of approximate graph homomorphism.

Definition 6.10 [80]. Fix a loopless graph $\mathbf{G}$, with vertices $v_1, \ldots, v_n$ and edges $(a_1, b_1), \ldots, (a_m, b_m)$ where each edge (u, v) is listed as both (u, v) and (v, u). The $\mathbf{G}$-*loop condition* is the bipartite minor condition:

$$f(x_{v_1}, \ldots, x_{v_n}) \approx e(x_{a_1}, \ldots, x_{a_m}),$$
$$f(x_{v_1}, \ldots, x_{v_n}) \approx e(x_{b_1}, \ldots, x_{b_m}).$$

We remark that the $\mathbf{G}$-loop condition can be also constructed as $\Sigma(\mathbf{G}, \mathbf{L})$ (recall Section 3.2) where the graph $\mathbf{L}$ is "the loop," i.e., the graph with a single vertex with a loop.

Example 6.11. Consider the $\mathbf{K}_3$-loop condition with edges of $\mathbf{K}_3$ listed as $(0,1)$, $(1,0)$, $(0,2)$, $(2,0)$, $(1,2)$, $(2,1)$. Then the functions e satisfying this condition (with some f) are exactly the Siggers functions.

Theorem 6.12. *The following are equivalent:*

(1) *PCSP*$(\mathbf{C}_{2k+1}, \mathbf{K}_c)$ *is NP-hard for all* $c \geq 3$, $k \geq 1$.
(2) *PCSP*$(\mathbf{G}, \mathbf{H})$ *is NP-hard for any two non-bipartite loopless graphs* $\mathbf{G}, \mathbf{H}$ *with* $\mathbf{G} \to \mathbf{H}$.
(3) *For any finite template* $(\mathbf{A}, \mathbf{B})$ *that does not satisfy the* $\mathbf{G}$*-loop condition for some non-bipartite loopless graph* $\mathbf{G}$, *PCSP*$(\mathbf{A}, \mathbf{B})$ *is NP-hard.*

Proof. (2) → (3) is proven similarly to Theorem 6.9, and (3) → (1) is given by the fact that $(\mathbf{C}_{2k+1}, \mathbf{K}_c)$ does not satisfy the $\mathbf{C}_{2k+1}$-loop condition.

(1) → (2). Assume that PCSP$(\mathbf{C}_{2k+1}, \mathbf{K}_c)$ is NP-hard, and let $\mathbf{G}$, $\mathbf{H}$ be loopless non-bipartite graphs. Since $\mathbf{G}$ is not bipartite, it contains an odd cycle, i.e., $\mathbf{C}_{2k+1} \to \mathbf{G}$ for some k, and, since $\mathbf{H}$ is loopless, it is colourable by some finite number of colours, i.e., $\mathbf{H} \to \mathbf{K}_c$ for some c. Therefore, $(\mathbf{C}_{2k+1}, \mathbf{K}_c)$ is a relaxation of $(\mathbf{G}, \mathbf{H})$, and the claim follows, e.g., by Lemma 4.8 and Theorem 3.1. □

7 TRACTABILITY FROM SOME CSPS

In the previous section, we described some limitations on bipartite minor conditions satisfied in the polymorphism minion (such as the absence of an Olšák function) that imply hardness of the corresponding PCSP. The goal of this section is to describe some minor conditions that imply tractability of the corresponding PCSPs. Observe first that, after excluding the trivial cases that imply a constant operation (the condition $f(x) \approx f(y)$), no other finite set of identities can imply tractability, since it would involve functions of bounded arity and hence is satisfied in the minion of functions of some bounded essential arity whose corresponding PCSP is NP-hard (see Proposition 5.14). Therefore, the conditions implying tractability always involve an infinite set of (minor) identities.

We do not provide any new concrete tractability results here; our contribution is an algebraic characterisation of the power of several known algorithms. Each of the conditions that we present describes applicability of a certain algorithm. Some of these algorithms are based on known algorithms for CSPs, e.g., References [43, 71], and some on recent tractability results for PCSPs [30]. In general, these algorithms are obtained from algorithms for a fixed tractable CSP with template $\mathbf{D}$ (possibly with infinite domain), and we apply them to all PCSP templates that are pp-constructible from $\mathbf{D}$ (recall Definition 4.9). This can be viewed as a (slight) generalisation of the "homomorphic sandwiching" method described by Brakensiek and Guruswami [30]. We show in the next section that there exist a tractable finite fixed-template PCSP whose tractability cannot be explained by pp-constructibility from (or sandwiching) a finite CSP template.

We consider three algorithms. The first one is based on a special case of local consistency, the second on the basic LP relaxation, and the last one on an affine integer relaxation. The last two types are closely related to the algorithms considered in Reference [30], and we refer to that paper for the latest algorithmic results for PCSPs that are based on relaxations over various numerical domains. We remark that the complexity of (infinite-domain) CSP with numerical domains is receiving a good amount of attention (see Reference [22]), but the connection with PCSPs has not been well-studied yet.

7.1 Local Consistency

A large family of algorithms for solving CSPs is based on some type of local propagation. Such an algorithm runs a propagation procedure that either refutes a given instance or transforms it to a locally consistent instance, without changing the set of solutions. This algorithm is said to solve $\mathrm{CSP}(\mathbf{A})$ if every locally consistent instance has a solution. Such algorithms naturally generalise to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$: The algorithm interprets the input $\mathbf{I}$ as an instance of $\mathrm{CSP}(\mathbf{A})$ and runs the appropriate consistency checking. A negative answer means that there is no homomorphism from $\mathbf{I}$ to $\mathbf{A}$, and, for correctness, we only require that every instance that is consistent as an instance of $\mathrm{CSP}(\mathbf{A})$ admits a homomorphism to $\mathbf{B}$. The most common type of such algorithms, the *bounded width* algorithm [13], works by inferring as much information about a solution as possible from considering fixed-size subsets of the instance, one at a time.

Definition 7.1. For a structure $\mathbf{I}$ and a subset $X \subseteq I$, let $\mathbf{I}[X]$ denote the structure *induced by* X in $\mathbf{I}$—its domain is X, and each relation $R_i^{\mathbf{I}}$ is replaced by $R_i^{\mathbf{I}} \cap X^{\mathrm{ar}(R_i)}$. A *partial homomorphism* from $\mathbf{I}$ to $\mathbf{A}$ with domain X is any homomorphism $\mathbf{I}[X] \to \mathbf{A}$.

Given an instance $\mathbf{I}$ of $\mathrm{CSP}(\mathbf{A})$ and $k \leq l$, the (k, l)*-consistency algorithm* constructs the largest family $\mathcal{F}$ of partial homomorphisms from $\mathbf{I}$ to $\mathbf{A}$ with at most l-element domains satisfying the following two conditions:

- for any $f \in \mathcal{F}$, all restrictions of f to smaller domains are also in $\mathcal{F}$, and
- for any $f \in \mathcal{F}$ with at most k-element domain, there is an extension $g \in \mathcal{F}$ of f to any l-element domain containing the domain of f.

On input $\mathbf{I}$, the algorithm starts with the set of all mappings from at most l-element subsets of I to A and repeatedly removes, until stable, mappings from $\mathcal{F}$ that are not partial homomorphisms or violate (at least) one of the two conditions above. We say that an instance $\mathbf{I}$ is a (k, l)*-consistent* *instance* of CSP($\mathbf{A}$) if the output of the (k, l)-consistency algorithm is a non-empty family of partial homomorphisms.

Definition 7.2. A PCSP template $(\mathbf{A}, \mathbf{B})$ has *width* (k, l) if every instance $\mathbf{I}$ that is (k, l)-consistent as an instance of CSP($\mathbf{A}$) maps homomorphically to $\mathbf{B}$. A template has *bounded width* if it has width (k, l) for some $k \leq l$, and *width 1* if it has width $(1, l)$ for some l.

Note that the (k, l)-consistency can be tested efficiently, which provides a polynomial time algorithm for PCSPs of bounded width. CSP templates of bounded width have been fully characterised by Barto and Kozik [13]. For PCSPs, we can characterise width 1, which corresponds to solvability by (generalised) arc consistency, one of the most prevalent algorithms used in constraint programming.

Definition 7.3. A function f is *totally symmetric* if its output depends only on the set of input elements (not accounting for multiplicity), i.e.,

$$f(x_{\pi(1)}, \dots, x_{\pi(n)}) \approx f(x_{\sigma(1)}, \dots, x_{\sigma(n)})$$

for all pairs of surjective maps $\pi, \sigma : [n] \to [m]$.

Note that if we replace "surjective maps" with "bijections" in the definition we get the definition of a *symmetric function* (see Definition 7.8 below).

The following theorem generalises the characterisation of width 1 for CSP templates from References [43, 48]. We add descriptions using minion homomorphisms and pp-constructibility. Recall the template $\mathbf{H}$ (of Horn 3-Sat) as defined in Example 4.2 and let $\mathscr{H} = \mathrm{Pol}(\mathbf{H})$.

Theorem 7.4. *Let $(\mathbf{A}, \mathbf{B})$ be a PCSP template. The following are equivalent:*

(1) $(\mathbf{A}, \mathbf{B})$ has width 1.
(2) $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ contains totally symmetric functions of all arities.
(3) There exists a minion homomorphism from $\mathscr{H}$ to $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$.
(4) $(\mathbf{A}, \mathbf{B})$ is pp-constructible from $\mathbf{H}$.

Proof sketch Let $\mathbf{F} = \mathbf{F}_{\mathscr{H}}(\mathbf{A})$ be the free structure of $\mathscr{H}$ generated by $\mathbf{A}$—see Example 4.2 for detailed information about this structure. By Theorem 4.12, items (3) and (4) are equivalent between themselves, and also equivalent to the condition $\mathbf{F} \to \mathbf{B}$. This last condition can be shown to be equivalent to both (1) and (2) essentially in the same way as the corresponding result for CSP [43] (where notation $C(\mathbf{A})$ is used for the free structure). □

Let us move towards the general bounded width algorithm. The following lemma shows that the class of PCSP templates of bounded width is closed under minion homomorphisms. The proof builds on an analogous result for CSPs obtained by Larose and Zádori [74]. To establish the hardness part of their then-conjectured characterisation of CSP templates of bounded width, they proved that bounded width is preserved under so-called "pp-interpretations" (pp-power is a special case of pp-interpretation) of CSP templates. The generalisation to PCSPs suggests that it may be possible to characterise bounded width for PCSPs by minor conditions as well, although we are not able to formulate such a conjecture yet.

Lemma 7.5. *Let $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}', \mathbf{B}')$ be templates such that there exists a minion homomorphism from $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ to $\mathrm{Pol}(\mathbf{A}', \mathbf{B}')$. If $(\mathbf{A}, \mathbf{B})$ has bounded width, then so does $(\mathbf{A}', \mathbf{B}')$.*

Proof sketch Using Theorem 4.12, it is enough to show that bounded width is preserved under homomorphic relaxations and pp-powers. We present a complete proof for relaxations and sketch a proof for pp-powers, since the latter proof follows Reference [74].

Suppose that $(\mathbf{A}', \mathbf{B}')$ is a homomorphic relaxation of a template $(\mathbf{A}, \mathbf{B})$, i.e., there exist homomorphisms $h_A : \mathbf{A}' \to \mathbf{A}$ and $h_B : \mathbf{B} \to \mathbf{B}'$, and that $(\mathbf{A}, \mathbf{B})$ has width (k, l). We will prove that in this case, $(\mathbf{A}', \mathbf{B}')$ has width (k, l) as well. Let $\mathbf{I}'$ be an instance of $\mathrm{CSP}(\mathbf{A}')$ that is (k, l)-consistent and let this fact be witnessed by a nonempty family of partial homomorphisms $\mathcal{F}'$. It is easy to see that the set $\{h_A f' \mid f' \in \mathcal{F}'\}$ witnesses that $\mathbf{I}'$ is (k, l)-consistent as an instance of $\mathrm{CSP}(\mathbf{A})$, and thus there exists a homomorphism $s : \mathbf{I}' \to \mathbf{B}$. Consequently, $h_B s : \mathbf{I}' \to \mathbf{B}'$.

We only sketch a proof for pp-powers: Assume that $(\mathbf{A}', \mathbf{B}')$ is a pp-power of $(\mathbf{A}, \mathbf{B})$, and the latter has width (k, l). We claim that $(\mathbf{A}', \mathbf{B}')$ has width (kM, lM) where M is the maximal arity of a relation of $(\mathbf{A}', \mathbf{B}')$. The key point is to observe that if we start with an instance $\mathbf{I}'$ of $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}')$ that is (kM, lM)-consistent, and replace every constraint of $\mathbf{I}'$ by its pp-definition in $(\mathbf{A}, \mathbf{B})$, then we obtain an instance $\mathbf{I}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ that is (k, l)-consistent. Note that this construction follows the standard reduction from $\mathrm{CSP}(\mathbf{A}')$ to $\mathrm{CSP}(\mathbf{A})$, therefore the arguments of Reference [74, Lemma 3.3] apply. Since $\mathbf{I}$ is (k, l)-consistent and $(\mathbf{A}, \mathbf{B})$ has width (k, l), we get $\mathbf{I} \to \mathbf{B}$. This homomorphism witnesses that $\mathbf{I}' \to \mathbf{B}'$, since $\mathbf{B}'$ is defined from $\mathbf{B}$ in the same way as $\mathbf{A}'$ from $\mathbf{A}$. □

7.2 Linear Programming Relaxations

Every CSP instance can be expressed as a 0-1 integer program in a canonical way. When we allow the variables in this program to attain any values from $[0, 1]$, we obtain the so-called basic linear programming relaxation [71].

Definition 7.6. Given an instance $\mathbf{I}$ of $\mathrm{CSP}(\mathbf{A})$, let $C = \{(\mathbf{v}, R) \mid \mathbf{v} \in R^{\mathbf{I}}\}$. The *basic linear programming relaxation* of $\mathbf{I}$ is the following linear program: The variables are $\mu_v(a)$ for every $v \in I$ and $a \in A$, and $\mu_{\mathbf{v},R}(\mathbf{a})$ for every $(\mathbf{v}, R) \in C$, and every $\mathbf{a} \in A^{\mathrm{ar}(R)}$. Each of the variables is allowed to have values in the interval $[0, 1]$. The objective is to maximise

$$\frac{1}{|C|} \sum_{(\mathbf{v},R) \in C} \sum_{\mathbf{a} \in R^{\mathbf{A}}} \mu_{\mathbf{v},R}(\mathbf{a})$$

subject to:

$$\sum_{a \in A} \mu_v(a) = 1 \qquad v \in I, \tag{7.1}$$

$$\sum_{\mathbf{a} \in A^{\mathrm{ar}(R)}, \mathbf{a}(i) = a} \mu_{\mathbf{v},R}(\mathbf{a}) = \mu_{\mathbf{v}(i)}(a) \qquad a \in A, (\mathbf{v}, R) \in C, i \in [\mathrm{ar}(R)]. \tag{7.2}$$

We denote the maximum possible value of the objective function by $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I})$.

It is clear that the optimum value, $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I})$, of this LP is smaller than or equal to 1, since Equations (7.1) and (7.2) together imply that $\sum_{\mathbf{a} \in A^{\mathrm{ar}(R)}} \mu_{\mathbf{v},R}(\mathbf{a}) = 1$, and therefore $\sum_{\mathbf{a} \in R} \mu_{\mathbf{v},R}(\mathbf{a}) \leq 1$ for each of the constraints $(\mathbf{v}, R)$. Given that the instance $\mathbf{I}$ has a solution as an instance of $\mathbf{A}$, say a homomorphism $s : \mathbf{I} \to \mathbf{A}$, there is an integral solution to the above linear program that achieves this optimum value: $\mu_v(s(v)) = 1$, $\mu_v(a) = 0$ for all $a \neq s(v)$, $\mu_{(v_1,\dots,v_k),R}(s(v_1), \dots, s(v_k)) = 1$, and $\mu_{\mathbf{v},R}(\mathbf{a}) = 0$ for all other $\mathbf{a}$'s.

Definition 7.7. Let $(\mathbf{A}, \mathbf{B})$ be a *PCSP* template. We say that BLP *solves PCSP* $(\mathbf{A}, \mathbf{B})$ if every instance $\mathbf{I}$ with $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$ maps homomorphically to $\mathbf{B}$.

It is easy to see that $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$ if and only if $\mu_{\mathbf{v},R}(\mathbf{a}) = 0$ for each constraint $(\mathbf{v}, R)$ in $\mathbf{I}$ and $a \notin R^{\mathbf{A}}$. Therefore, if we add all such constraints to $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I})$, then testing the feasibility of the obtained LP is equivalent to testing whether $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$. Therefore, if BLP solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, then $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ reduces to an LP feasibility problem, where each LP constraint (except the nonnegativity inequalities) is a linear equation with ±1 coefficients and a bounded number of variables. Such a problem is expressible as $\mathrm{CSP}(\mathbf{D})$ for an appropriate structure $\mathbf{D}$ with domain $\mathbb{Q}$ and finitely many relations. Note that the definition of a CSP extends in a straightforward way to such structures. The template $\mathbf{D}$ is obtained from the structure $\mathbf{Q}_{\text{conv.}}$, whose domain is $\mathbb{Q}$ and whose (infinitely many) relations are all possible linear inequalities with rational coefficients (see Reference [22, Definition 4]) by dropping all but finitely many relations. If a structure $\mathbf{D}$ is obtained in this way, then we say that it is a *finite reduct* of $\mathbf{Q}_{\text{conv.}}$. Note that the relations pp-definable in $\mathbf{Q}_{\text{conv.}}$ are the convex polytopes in $\mathbb{Q}^k$, $k \geq 1$. It is easy to see that the polymorphisms of $\mathbf{Q}_{\text{conv.}}$ are exactly convex linear functions, i.e., functions $f : \mathbb{Q}^n \to \mathbb{Q}$ defined by $f(x_1, \dots, x_n) = \sum_{i \in [n]} \alpha_i x_i$ for some α_i's, $i \in [n]$, such that $\alpha_i \in [0, 1]$, $\sum_{i \in [n]} \alpha_i = 1$. We denote the set of all such operations by $\mathscr{Q}_{\text{conv.}}$.

Definition 7.8. We say that a function f is *symmetric* if the output is independent of the order of the input elements, i.e.,

$$f(x_1, \dots, x_n) \approx f(x_{\pi(1)}, \dots, x_{\pi(n)})$$

for all bijections $\pi : [n] \to [n]$.

While total symmetry means that the output is dependent only on the set of the input elements, symmetry can be formulated as “the output is dependent only on the *multiset* of the input elements.”

Note that $\mathscr{Q}_{\text{conv.}}$ contains symmetric operations of all arities: For arity n, take $f(x_1, \dots, x_n) = \sum_{i \in [n]} x_i/n$.

We provide a characterisation of the direct applicability of the basic linear programming relaxation that generalises Reference [71, Theorem 2(5–6)]. Note that Reference [71, Theorem 2] is partially false: There is a finite CSP template $\mathbf{A}$ such that CSP($\mathbf{A}$) is solvable by BLP and $\mathbf{A}$ does not have width 1 [72, Example 99], contrary to what Reference [71] claims.

Theorem 7.9. *Let* $(\mathbf{A}, \mathbf{B})$ *be a PCSP template. The following are equivalent:*

(1) BLP solves PCSP$(\mathbf{A}, \mathbf{B})$,
(2) $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$ *contains symmetric functions of all arities,*
(3) $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$ *admits a minion homomorphism from* $\mathscr{Q}_{\text{conv.}}$,
(4) $(\mathbf{A}, \mathbf{B})$ *is pp-constructible from (a finite reduct of)* $\mathbf{Q}_{\text{conv.}}$.

The theorem above is stated for the decision variant of the PCSP, and it needs a slight modification to be valid for the search variant. Namely, to produce a solution of the PCSP from a BLP solution, we need an efficient algorithm that computes values of the symmetric polymorphisms of the template.

The structure of the proof is the same as for Theorem 7.4. Although, there are several important issues: mostly that Theorem 4.12 holds for finite structures, but in general it does not hold for infinite structures. Nevertheless, we will show that it holds for the structure $\mathbf{Q}_{\text{conv.}}$. The proof uses an appropriate modification of the notion of a free structure for $\mathscr{Q}_{\text{conv.}}$, which we define next. This modification is similar to the instance $\mathcal{M}(\Gamma)$ defined in Reference [71, Definitions 10 and 11]. To keep some consistency in our notation, we will denote this structure by $\operatorname{LP}(\mathbf{A})$.

Definition 7.10. The structures $\operatorname{LP}(\mathbf{A})$ and $\mathbf{A}$ are similar. The universe $\operatorname{LP}(A)$ consists of rational probability distributions on $\mathbf{A}$, i.e., functions $\phi : \mathbf{A} \to \mathbb{Q} \cap [0, 1]$ such that $\sum_{a \in A} \phi(a) = 1$. For a k-ary relation $R^{\mathbf{A}}$, the corresponding relation $R^{\operatorname{LP}(\mathbf{A})}$ is defined as the set of all k-tuples $(\phi_1, \dots, \phi_k)$ of elements of $\operatorname{LP}(A)$ for which there exists a rational probability distribution γ on $R^{\mathbf{A}}$ such that

$$\sum_{\mathbf{a} \in R^{\mathbf{A}}, \mathbf{a}(i) = a} \gamma(\mathbf{a}) = \phi_i(a) \tag{7.3}$$

for all $i \in [k]$ and $a \in A$.

Note that the above structure relates closely to the BLP relaxation: It is easy to observe that $\mathbf{I}$ is an instance such that $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$ if and only if $\mathbf{I}$ maps homomorphically to $\mathrm{LP}(\mathbf{A})$. This has also been observed in Reference [71].

Remark 7.11. The structure $\mathrm{LP}(\mathbf{A})$ is isomorphic to the free structure $\mathbf{F}_{\mathscr{Q}}(\mathbf{A})$ of $\mathscr{Q}_{\text{conv.}}$ generated by $\mathbf{A}$. For a proof, assume that $A = [n]$. Recall that the elements of the free structure $\mathbf{F}_{\mathscr{Q}}(\mathbf{A})$ are n-ary convex linear functions. An isomorphism $h : \mathrm{LP}(\mathbf{A}) \to \mathbf{F}_{\mathscr{Q}}(\mathbf{A})$ is given by $h(\phi) = f_\phi$ where f_ϕ is defined by $f_\phi(x_1, \ldots, x_n) = \sum_{i\in[n]} \phi(i)x_i$. Note that f_ϕ is always a convex linear function, therefore an element of $F_{\mathscr{Q}}(A)$, and also that h is bijective, since every such function is uniquely determined by its coefficients. To prove that h preserves a relation R, assume $(\phi_1, \ldots, \phi_k) \in R^{\mathrm{LP}(\mathbf{A})}$, and let γ be the probability distribution on $R^{\mathbf{A}}$ witnessing this fact. Further, let $R^{\mathbf{A}} = \{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$, and define $g : \mathbb{Q}^m \to \mathbb{Q}$ by

$$g(x_1, \ldots, x_m) = \sum_{i\in[m]} \gamma(\mathbf{r}_i)x_i.$$

We claim that for each $j \in [k]$, we have

$$g(x_{\mathbf{r}_1(j)}, \ldots, x_{\mathbf{r}_m(j)}) = h(\phi_j)(x_1, \ldots, x_n),$$

which is easily observed by comparing coefficients of the left- and right-hand side. The fact that h^{-1} is also a homomorphism is obtained by reversing this argument, equating a convex linear function $g : \mathbb{Q}^m \to \mathbb{Q}$ with the probability distribution γ, which maps a tuple $\mathbf{r}_i$ to the ith coefficient of g.

In the light of the previous remark, the following can be understood as an infinite case of Lemma 4.11:

Lemma 7.12. *Let* $\mathbf{A}$ *be a finite relational structure, and let* $\mathrm{LP}(\mathbf{A})$ *be the free structure of* $\mathscr{Q}_{\mathit{conv.}}$ *generated by* $\mathbf{A}$*. Then* $(\mathbf{A}, \mathrm{LP}(\mathbf{A}))$ *is a relaxation of a pp-power of* $\mathbf{Q}_{\mathit{conv.}}$*.*

Proof. In this proof, we assume $A = [n]$ and equate a probability distribution ϕ on A with the tuple $(\phi(1), \ldots, \phi(n))$. Let us define an nth pp-power $\mathbf{P}$ of $\mathbf{Q}_{\text{conv.}}$ and its relaxation that will be isomorphic to $(\mathbf{A}, \mathrm{LP}(\mathbf{A}))$: A relation $R^{\mathbf{P}}$ is defined to contain all tuples $(\phi_1, \ldots, \phi_k)$ of n-tuples such that $\phi_j \geq 0$ and $\sum_{i\in[n]} \phi_j(i) = 1$ for all j, and so there exists $\gamma : R^{\mathbf{A}} \to \mathbf{Q}$ such that Equation (7.3) is satisfied. This is indeed a pp-definition, since each of the inequalities and identities define a relation of $\mathbf{Q}_{\text{conv.}}$. Finally, we argue as in Lemma 4.11: $\mathbf{A}$ maps homomorphically into $\mathbf{P}$ by $a \mapsto \chi_a$ where $\chi_a(b) = 1$ for $b = a$ and $\chi_a(b) = 0$ for $b \neq a$, and $\mathbf{P}$ maps homomorphically to $\mathrm{LP}(\mathbf{A})$ by $\phi \mapsto f_\phi$ whenever $\sum_{a\in A} \phi(a) = 1$ and $\phi(a) \geq 0$, and extending arbitrarily. □

Remark 7.13. In the proof below, we will need to find a homomorphism from a relational structure with an infinite universe, let us for now call it $\mathbf{I}$, to a similar finite relational structure $\mathbf{B}$. It is well-known that in that case it is enough to find a homomorphism from all finite substructures of $\mathbf{I}$. This fact is usually proven by a standard compactness argument, e.g., using Tychonoff's theorem. For completeness, we present one such argument that uses König's lemma. This approach works only for countable structures $\mathbf{I}$, which is enough in our case.

Assuming that $I = \{1, 2, \ldots\}$ and that every finite substructure of $\mathbf{I}$ maps to $\mathbf{B}$, we construct an infinite, finitely branching tree: The nodes of the tree are partial homomorphisms defined on sets $[n]$ (starting with the empty set, so the empty mapping is the root of the tree). We set that a partial homomorphism $r_{n+1} : [n + 1] \to \mathbf{B}$ is a child of $r_n : [n] \to \mathbf{B}$ if the map r_n is the restriction of r_{n+1} to $[n]$. Clearly, this tree is finitely branching and infinite. König's lemma states that such a tree has an infinite branch, which in our case gives us a sequence $r_1, r_2, \ldots$ such that r_{n+1} extends r_n. We define $r : \mathbf{I} \to \mathbf{B}$ as the union of these maps. It is a homomorphism, since all constraints are local, and therefore included in the domain of some r_n, which is a partial homomorphism.

Proof of Theorem 7.9 First, we show that item (1) is equivalent to the existence of a homomorphism from $\mathrm{LP}(\mathbf{A})$ to $\mathbf{B}$. The direct implication follows by the same argument as Reference [71, Proposition 12]. If BLP solves $\mathrm{CSP}(\mathbf{A})$ and $\mathbf{I}$ is a finite subinstance of $\mathrm{LP}(\mathbf{A})$, then it is easy to see that $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$, and therefore $\mathbf{I} \to \mathbf{B}$. Using a standard compactness argument (see the previous remark), this implies that $\mathrm{LP}(\mathbf{A})$ homomorphically maps to $\mathbf{B}$. The converse is straightforward: Indeed, any instance $\mathbf{I}$ with $\mathrm{BLP}_{\mathbf{A}}(\mathbf{I}) = 1$ maps homomorphically to $\mathrm{LP}(\mathbf{A})$, and therefore also $\mathbf{I} \to \mathbf{B}$ because $\mathrm{LP}(\mathbf{A}) \to \mathbf{B}$.

Given a homomorphism from $\mathrm{LP}(\mathbf{A})$ to $\mathbf{B}$, we get item (4), i.e., that $(\mathbf{A}, \mathbf{B})$ is pp-constructible from $\mathbf{Q}_{\text{conv.}}$, from Lemma 7.12. Further, (4) → (3) follows from Lemma 4.8 (note that the proof does not require the structures to be finite). We have (3) → (2), since $\mathscr{Q}_{\text{conv.}}$ contains symmetric operations of all arities and any minion homomorphism preserves this property. (Note that (1) → (2) can be also obtained by argument similar to Reference [71, Proposition 12].)

Let us prove that (2) implies that $\mathrm{LP}(\mathbf{A}) \to \mathbf{B}$, and hence item (1). To do that, we define homomorphisms from certain finite substructures of the structure $\mathrm{LP}(\mathbf{A})$: We define $\mathrm{LP}_\ell(\mathbf{A})$ to be a structure similar to $\mathbf{A}$ whose universe consists of rational probability distributions on $\mathbf{A}$ with denominators dividing ℓ, i.e., functions $\phi : A \to \mathbb{Q}$ where, for each $a \in A$, $\phi(a) = q/\ell$ for some $q \in \{0, 1, \dots, \ell\}$ and $\sum_{a \in A} \phi(a) = 1$. The relations are defined the same way as in $\mathrm{LP}(\mathbf{A})$ where we restrict to probability distributions γ with denominators dividing ℓ. (Note that, unlike instances $\mathcal{M}_\ell(\Gamma)$ defined in Reference [71, Definition 10], the structure $\mathrm{LP}_\ell(\mathbf{A})$ is not an induced substructure of $\mathrm{LP}(\mathbf{A})$.)

We define a homomorphism $h_\ell : \mathrm{LP}_\ell(\mathbf{A}) \to \mathbf{B}$ by fixing a symmetric function $s_\ell \in \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ of arity ℓ, and setting

$$h_\ell(\phi) = s_\ell(a_1, \dots, a_\ell)$$

where $a_1, \dots, a_\ell \in A$ are chosen so each a appears exactly $\phi(a)\ell$ times. Note that the order of a_i's does not matter, since s_ℓ is symmetric. To show that h_ℓ is a homomorphism, consider a tuple $(\phi_1, \dots, \phi_k) \in R^{\mathrm{LP}_\ell \mathbf{A}}$ and let $\gamma : R^{\mathbf{A}} \to \mathbb{Q}$ be the witnessing probability distribution. We pick tuples $\mathbf{r}_1, \dots, \mathbf{r}_\ell \in R^{\mathbf{A}}$ such that each $\mathbf{r} \in R^{\mathbf{A}}$ appears exactly $\gamma(\mathbf{r})\ell$ times. Now,

$$(h_\ell(\phi_1), \dots, h_\ell(\phi_k)) = s_\ell(\mathbf{r}_1, \dots, \mathbf{r}_\ell) \in R^{\mathbf{B}},$$

where the equality follows, since, for each $i \in [k]$, each $a \in A$ appears exactly $\phi_i(a)\ell = \sum_{r \in \mathrm{R}^{\mathrm{A}}, \mathbf{r}(i)=a} \gamma(\mathbf{r})\ell$ times among $\mathbf{r}_1(i), \dots, \mathbf{r}_\ell(i)$ (again, we use symmetry of s_ℓ). This establishes that every $\mathrm{LP}_\ell(\mathbf{A})$ maps homomorphically to $\mathbf{B}$. Note that every finite substructure of $\mathrm{LP}(\mathbf{A})$ is a substructure of $\mathrm{LP}_\ell(\mathbf{A})$ for a big enough ℓ, and hence every finite substructure of $\mathrm{LP}(\mathbf{A})$ maps homomorphically to $\mathbf{B}$. Thus, we have a homomorphism from $\mathrm{LP}(\mathbf{A})$ to $\mathbf{B}$ by a standard compactness argument. □

7.3 Affine Diophantine Relaxations

Another way to relax the natural 0-1 integer program expressing a CSP instance is to allow the variables to attain any integer values, and relax constraints to linear equations. We get a relaxation with very similar properties to the basic LP relaxation described in the previous section.

Definition 7.14. Given an instance $\mathbf{I}$ of $\mathrm{CSP}(\mathbf{A})$, let $C = \{(\mathbf{v}, R) \mid v \in R^{\mathbf{I}}\}$. The *basic affine integer relaxation* of $\mathbf{I}$, denoted $\mathrm{AIP}_{\mathbf{A}}(\mathbf{I})$, is the following affine program: The variables are $\mu_v(a)$ for every $v \in I$ and $a \in A$, and $\mu_{\mathbf{v},R}(\mathbf{a})$ for every $(\mathbf{v}, R) \in C$ and $\mathbf{a} \in R^{\mathbf{A}}$. The objective is to solve the following system over $\mathbb{Z}$:

$$\sum_{a \in A} \mu_v(a) = 1 \qquad v \in I, \tag{7.4}$$

$$\sum_{\mathbf{a} \in R^{\mathbf{A}}, \mathbf{a}(i)=a} \mu_{\mathbf{v},R}(\mathbf{a}) = \mu_{\mathbf{v}(i)}(a) \qquad a \in A, (\mathbf{v}, R) \in C, i \in [\mathrm{ar}(R)]. \tag{7.5}$$

Note that this system is an instance of the infinite CSP with template $\mathbf{Z}_{\text{aff.}}$ with domain $\mathbb{Z}$ whose relations are all affine equations over $\mathbb{Z}$. Let $\mathscr{Z}_{\text{aff.}} = \operatorname{Pol}(\mathbf{Z}_{\text{aff.}})$. We claim that $\mathscr{Z}_{\text{aff.}}$ consists of all affine functions over $\mathbb{Z}$, i.e., all functions $g : \mathbb{Z}^n \to \mathbb{Z}$ described as $g(x_1, \dots, x_n) = \sum_{i \in [n]} \gamma(i) x_i$ where $\gamma : [n] \to \mathbb{Z}$ is such that $\sum_{i \in [n]} \gamma(i) = 1$. Clearly, any such function g is a polymorphism. For the other inclusion assume that $f : \mathbb{Z}^n \to \mathbb{Z}$ is a polymorphism. Then, in particular, it preserves the relation $\{(x, y, z) \mid x + y = z\}$, which implies that it is linear. Also, it preserves the unary singleton relation $\{1\}$ (the solution to the equation $x = 1$), which implies that the sum of its coefficients is 1.

This situation is similar to the one with convex linear functions in the previous subsection. And in fact, using similar methods, we obtain similar results. Let us first show an example of the use of this relaxation.

Example 7.15. Let us describe the AIP relaxation of 1-in-3- vs. Not-All-Equal-Sat and compare it with an algorithm for this PCSP described in Reference [29]. Recall that the PCSP template of this problem is $(\mathbf{T}, \mathbf{H}_2)$ as defined in Example 2.8, we denote the single ternary relation of these structures by R.

The basic affine integer relaxation of an instance $\mathbf{I}$ of $\operatorname{PCSP}(\mathbf{T}, \mathbf{H}_2)$ is a system of equations using variables $\mu_v(0)$, $\mu_v(1)$ for $v \in I$ bound by $\mu_v(0) + \mu_v(1) = 1$, and $\mu_{(v_1, v_2, v_3)}(\mathbf{a})$ for $(v_1, v_2, v_3) \in R^{\mathbf{I}}$ and $\mathbf{a} \in R^{\mathbf{T}}$:

$$\begin{aligned}
\mu_{v_1, v_2, v_3}(0,0,1) + \mu_{v_1, v_2, v_3}(0,1,0) &= \mu_{v_1}(0), \\
\mu_{v_1, v_2, v_3}(1,0,0) &= \mu_{v_1}(1), \\
\mu_{v_1, v_2, v_3}(0,0,1) + \mu_{v_1, v_2, v_3}(1,0,0) &= \mu_{v_2}(0), \\
\mu_{v_1, v_2, v_3}(0,1,0) &= \mu_{v_2}(1), \\
\mu_{v_1, v_2, v_3}(0,1,0) + \mu_{v_1, v_2, v_3}(1,0,0) &= \mu_{v_3}(0), \\
\mu_{v_1, v_2, v_3}(0,0,1) &= \mu_{v_3}(1),
\end{aligned}$$

for each $(v_1, v_2, v_3) \in R^{\mathbf{I}}$. Since the value of $\mu_v(0)$ is determined by the value of $\mu_v(1)$, and moreover $\sum_{\mathbf{a} \in R^{\mathbf{T}}} \mu_{v_1, v_2, v_3}(\mathbf{a}) = \mu_{v_1}(0) + \mu_{v_1}(1) = 1$, we can simplify this system by dropping variables $\mu_v(0)$ and $\mu_{v_1, v_2, v_3}(i)$ and replacing the six equations above with

$$\mu_{v_1}(1) + \mu_{v_2}(1) + \mu_{v_3}(1) = 1.$$

Note that each satisfying tuple of $R^{\mathbf{T}}$ satisfies this constraint. The resulting system is the same as suggested by Reference [29, Remark 3.3].

Definition 7.16. A function a of arity $2n + 1$ is called *alternating*, if

$$a(x_1, \dots, x_{2n+1}) \approx a(x_{\pi(1)}, \dots, x_{\pi(2n+1)})$$

for all permutations π that preserve parity, i.e., $\pi(i) \bmod 2 = i \bmod 2$, and

$$a(x_1, \dots, x_{2n-1}, y, y) \approx a(x_1, \dots, x_{2n-1}, z, z)$$

assuming $n \geq 1$. (A function of arity 1 is alternating.)

Even though the definition makes sense for functions of even arities, we refrain from calling such functions alternating. The main reason is that such an "even arity alternating function" always has a minor that is a constant function and is therefore as useful for classifying PCSPs as a constant function is.

The property of being alternating can be also expressed as a variant of symmetry: The value is independent of the order of its inputs on odd positions, and also of the order of its inputs on even positions. Putting this together with the second identity, which expresses some form of cancellation, we get that the output is dependent only on the multiset (where we allow negative coefficients) of inputs where the odd inputs are counted positively and even inputs negatively.

Example 7.17. An example of an alternating function is the function $a : \mathbb{Z}^{2n+1} \to \mathbb{Z}$ defined as the alternating sum, i.e.,

$$a(x_1, \dots, x_n) = x_1 - x_2 + x_3 - \cdots + x_{2n+1}.$$

Clearly, permuting x_i's with odd indices as well as permuting those with even indices does not change the value. Also

$$a(x_1, \dots, x_{2n-1}, y, y) = x_1 - \cdots + x_{2n-1} - y + y = x_1 - \cdots + x_{2n-1},$$

which concludes that this value does not depend on y. Also note that the function defined by

$$a'(x_1, \dots, x_{2n-1}) = a(x_1, \dots, x_{2n-1}, x_1, x_1)$$

is an alternating function of arity $2n - 1$. This is always the case, as can be easily derived from the two defining identities.

Note that such alternating sum can be generalised from $\mathbb{Z}$ to any abelian group. One such example, though degenerate, would be the Boolean *parity function* (see Reference [27, p. 9]) of odd arity; it is defined as $p(x_1, \dots, x_{2n+1}) = \sum_{i \in [2n+1]} x_i \bmod 2$.

Example 7.18. Another important example is the *alternating threshold* defined in Reference [27, p. 9] as the Boolean function t of arity $2n + 1$ satisfying:

$$t(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } x_1 - x_2 + x_3 - \cdots + x_{2n+1} > 0, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

This function is obtained from the alternating sum by composing with the natural inclusion $e : \{0, 1\} \to \mathbb{Z}$ and the function $r : \mathbb{Z} \to \{0, 1\}$, which maps positive integers to 1 and non-positive to 0 in the following way:

$$t(x_1, \dots, x_{2n-1}) = r(a(e(x_1), \dots, e(x_{2n-1}))).$$

This composition is a *reflection* that we define later in Definition 9.1.

Let us now formulate the main result of this section.

Theorem 7.19. *Let* $(\mathbf{A}, \mathbf{B})$ *be a PCSP template. The following are equivalent:*

1. *AIP solves PCSP*$(\mathbf{A}, \mathbf{B})$*,*
2. $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *contains alternating functions of all odd arities,*
3. $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *admits a minion homomorphism from* $\mathscr{Z}_{\mathit{aff.}}$*,*
4. $(\mathbf{A}, \mathbf{B})$ *is pp-constructible from (a finite reduct of)* $\mathbf{Z}_{\mathit{aff.}}$*.*

This theorem is stated for the decision variant of the PCSP, and similarly to Theorem 7.9, it needs a modification for the search variant: To produce a solution of the PCSP from an AIP solution, we need an efficient algorithm that computes values of the alternating polymorphisms of the template.

The proof of this theorem follows closely the proof of Theorem 7.9. As in the mentioned proof, we rely on an appropriate modification of the free structure of $\mathscr{Z}_{\text{aff.}}$. In this case, there is no straightforward interpretation as probability distributions, but the core idea remains the same.

Definition 7.20. For a structure $\mathbf{A}$, we define an infinite structure $\mathrm{IP}(\mathbf{A})$ similar to $\mathbf{A}$ in the following way: The universe $\mathrm{IP}(A)$ is the set of all mappings (tuples) $\phi : A \to \mathbb{Z}$ such that $\sum_{a \in A} \phi(a) = 1$.

For a k-ary relation $R^{\mathbf{A}}$, we define the corresponding relation $R^{\mathrm{IP}(\mathbf{A})}$ as the set of all k-tuples $(\phi_1, \dots, \phi_k)$ for which there exists a mapping $\gamma : R^{\mathbf{A}} \to \mathbb{Z}$ such that $\sum_{\mathbf{a} \in R^{\mathbf{A}}} \gamma(\mathbf{a}) = 1$ and

$$\sum_{\mathbf{a} \in R^{\mathbf{A}}, \mathbf{a}(i) = a} \gamma(\mathbf{a}) = \phi_i(a)$$

for all $a \in A$ and $i \in [k]$.

Remark 7.21. It can be proven in very similar way as in Remark 7.11 that the structure $\mathrm{IP}(\mathbf{A})$ is isomorphic to the free structure of $\mathscr{Z}_{\text{aff.}}$ generated by $\mathbf{A}$.

Similarly to Lemma 7.12, the following can be viewed as an infinite case of Lemma 4.11:

Lemma 7.22. *Let* $\mathbf{A}$ *be a finite relational structure, and let* $\mathrm{IP}(\mathbf{A})$ *denote the free structure of* $\mathscr{Z}_{\text{aff.}}$ *generated by* $\mathbf{A}$. *Then* $(\mathbf{A}, \mathrm{IP}(\mathbf{A}))$ *is a relaxation of a pp-power of* $\mathbf{Z}_{\text{aff.}}$.

Proof. The lemma is proven the same way as Lemma 7.12. □

Proof of Theorem 7.19 We claim that (1) is equivalent to the existence of a homomorphism from $\mathrm{IP}(\mathbf{A})$ to $\mathbf{B}$. The key is that an instance $\mathbf{I}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ maps to $\mathrm{IP}(\mathbf{A})$ if and only if the AIP relaxation of $\mathbf{I}$ is a solvable system of equations—a solution μ defines such a homomorphism by $v \mapsto f_{\mu_v}$ where $f_{\mu_v}(x_1, \dots, x_n) = \sum_{a \in A} \mu_v(a) x_a$ (here, we assume $A = [n]$) and vice versa. This immediately implies that if $\mathrm{IP}(\mathbf{A})$ maps homomorphically to $\mathbf{B}$, then AIP solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. In the other direction, we know that every finite substructure of $\mathrm{IP}(\mathbf{A})$ maps to $\mathbf{B}$, a global homomorphism then follows by the standard compactness argument.

As in the proof of Theorem 7.9, the combination of the previous lemma with the proof of Theorem 4.12 gives us that the existence of a homomorphism from $\mathrm{IP}(\mathbf{A})$ to $\mathbf{B}$ implies item (4), and (4) → (3). Further, (3) → (2), since $\mathscr{Z}_{\text{aff.}}$ has alternating functions of all odd arities (the alternating sums).

We finish the proof by showing that item (2) implies the existence of a homomorphism from $\mathrm{IP}(\mathbf{A})$ to $\mathbf{B}$. This again follows the proof of Theorem 7.9. We first define some substructures of $\mathrm{IP}(\mathbf{A})$: $\mathrm{IP}_\ell(\mathbf{A})$ is a structure similar to $\mathbf{A}$ whose universe is the set of all functions $\phi : A \to \mathbb{Z}$ such that $\sum_{a \in A} |\phi(a)| \leq 2\ell + 1$ and $\sum_{a \in A} \phi(a) = 1$. The relations of $\mathrm{IP}_\ell(\mathbf{A})$ are defined in the same way as those of $\mathrm{IP}(\mathbf{A})$ with the only difference that we require that the witnessing function $\gamma : R^{\mathbf{A}} \to \mathbb{Z}$ also satisfies $\sum_{\mathbf{r} \in R^{\mathbf{A}}} |\gamma(\mathbf{r})| \leq 2\ell + 1$. Now, we define a homomorphism $h_\ell : \mathrm{IP}_\ell(\mathbf{A}) \to \mathbf{B}$ by fixing a $(2\ell + 1)$-ary alternating function $a_{2\ell+1} \in \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ and setting

$$h_\ell(\phi) = a_{2\ell+1}(b_1, \dots, b_{2\ell+1}),$$

where $b_1, \dots, b_{2\ell+1} \in A$ are chosen in such a way that for all $a \in A$ the difference of the number of times a appears among b_i with odd and even indices is exactly $\phi(a)$. This is possible, thanks to $\sum |\phi(a)| \leq 2\ell + 1$. For h to be well-defined, we rely on the fact that $a_{2\ell+1}$ is alternating. To prove that h is a homomorphism, suppose that $(\phi_1, \dots, \phi_k) \in \mathrm{R}^{\mathrm{IP}_\ell(\mathbf{A})}$ and this fact is witnessed by $\gamma : R^{\mathbf{A}} \to \mathbb{Z}$. Again, we pick tuples $\mathbf{r}_1, \dots, \mathbf{r}_{2\ell+1}$ in such a way that for all $\mathbf{r} \in R^{\mathbf{A}}$ the difference of the number of times $\mathbf{r}$ appears among $\mathbf{r}_i$ with odd and even indices is exactly $\phi(\mathbf{r})$. Now,

$$(h_\ell(\phi_1), \dots, h_\ell(\phi_k)) = a_{2\ell+1}(\mathbf{r}_1, \dots, \mathbf{r}_{2\ell+1}) \in R^{\mathbf{B}},$$

where the equality follows from the fact that, for each $i \in [k]$, the difference between the number of times some $a \in A$ appears among $\mathbf{r}_j(i)$ with odd and even j is $\phi_i(a) = \sum_{\mathbf{r} \in R^{\mathbf{A}}, \mathbf{r}(i) = a} \gamma(\mathbf{r})$ (again, we use the fact that $a_{2\ell+1}$ is alternating). We get that each $\mathrm{IP}_\ell(\mathbf{A})$ maps homomorphically to $\mathbf{B}$, and, since every finite substructure of $\mathrm{IP}(\mathbf{A})$ is included in $\mathrm{IP}_\ell(\mathbf{A})$ for some ℓ, we get that it also maps to $\mathbf{B}$. The homomorphism from $\mathrm{IP}(\mathbf{A})$ to $\mathbf{B}$ is then given by a standard compactness argument. □

8 MORE ON TRACTABILITY OF 1-in-3- vs. NAE-SAT

All known tractability results for PCSPs, such as those in the previous section and those in References [29, 30], are obtained by following the same scheme—namely, by showing how a PCSP template is pp-constructed from a tractable CSP template (recall Definition 4.9), possibly with an infinite domain. In this section, we show that using infinite domains in this scheme can be necessary. Namely, we show that this is the case for the 1-in-3- vs. Not-All-Equal-Sat problem (see Example 2.8). This problem is in P [29], since its template $(\mathbf{T}, \mathbf{H}_2)$ can be pp-constructed from $\mathbf{Z}_{\text{aff.}}$ (see Example 7.15 and Theorem 7.19). Specifically, it is easy to check (or see Reference [29]) that this template is a relaxation of the CSP template $(\mathbb{Z}; x + y + z = 1)$, which is a finite reduct of $\mathbf{Z}_{\text{aff.}}$. The template $(\mathbf{T}, \mathbf{H}_2)$ is also a relaxation of other tractable CSP templates with an infinite domain (see References [10, 29, 30]).

Theorem 8.1. *Let $\mathbf{D}$ be a finite relational structure such that $(\mathbf{T}, \mathbf{H}_2)$ is pp-constructible from $\mathbf{D}$. Then CSP($\mathbf{D}$) is NP-complete.*

The rest of this section is devoted to the proof of this theorem.

8.1 Proof Outline

Striving for a contradiction, assume that CSP($\mathbf{D}$) is not NP-complete and $(\mathbf{T}, \mathbf{H}_2)$ is pp-constructible from $\mathbf{D}$. We start by simplifying the latter assumption.

From (6) → (5) in Theorem 4.12, we know that $(\mathbf{T}, \mathbf{H}_2)$ is a homomorphic relaxation of a pp-power of $\mathbf{D}$. Since a pp-power of a finite tractable CSP template is a finite tractable CSP template, we may assume that $(\mathbf{T}, \mathbf{H}_2)$ is a homomorphic relaxation of $\mathbf{D}$. Let $\mathbf{D} = (D; R)$, where $R \subseteq D^3$, and let $f : \mathbf{T} \to \mathbf{D}$ and $g : \mathbf{D} \to \mathbf{H}_2$ be homomorphisms from the definition of homomorphic relaxation, Definition 4.6.

We simplify the situation a bit further. Since gf is a homomorphism, this mapping applied component-wise to the 1-in-3 tuple $(0, 0, 1)$ is a not-all-equal tuple. In particular, $f(0) \neq f(1)$. We rename the elements of D so $\{0, 1\} \subseteq D$ and $f(0) = 0$, $f(1) = 1$. As f and g are homomorphisms, we get

$$\{0, 1\} \subseteq D, \quad \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\} \subseteq R$$

and

$$|\{g(a), g(b), g(c)\}| > 1 \text{ whenever } (a, b, c) \in R.$$

Now, we employ the assumption that CSP($\mathbf{D}$) is not NP-complete. We use a sufficient condition for NP-completeness from Reference [12].

Definition 8.2. An operation $s : D^n \to D$ is called *cyclic* if, for all $(a_1, \ldots, a_n) \in D^n$, we have

$$s(a_1, a_2, \ldots, a_n) = s(a_2, \ldots, a_n, a_1).$$

Theorem 8.3. *Let $\mathbf{D}$ be a finite CSP template. If CSP($\mathbf{D}$) is not NP-complete, then $\mathbf{D}$ has a cyclic polymorphism of arity p for every prime number $p > |D|$.*

Remark 8.4. Cyclic polymorphisms in fact characterise the borderline between NP-complete and tractable CSPs conjectured in Reference [34] and proved in References [36, 92, 93]: CSP($\mathbf{D}$) is tractable if and only if (assuming P ≠ NP) $\mathbf{D}$ has a cyclic polymorphism of arity at least 2 (if and only if $\mathbf{D}$ has a cyclic polymorphism of arity p for every prime number $p > |D|$).

By Theorem 8.3, $\mathbf{D}$ has a cyclic polymorphism of any prime arity $p > |D|$. We fix a cyclic polymorphism s of prime arity $p > 60|D|$.

Recall that the polymorphisms of CSP template can be composed (to produce new polymorphisms). Next, we define an operation t on D of arity p^2 by

$$\begin{aligned} &t(x_{11}, x_{12}, \dots, x_{1p}, x_{21}, x_{22}, \dots x_{2p}, \dots, x_{p1}, \dots, x_{pp}) \\ &\quad = s(s(x_{11}, x_{21}, \dots, x_{p1}), s(x_{12}, x_{22}, \dots, x_{p2}), \dots, s(x_{1p}, x_{2p}, \dots, x_{pp})). \end{aligned}$$

It will be convenient to organise the arguments of t into a $p \times p$ matrix X whose entry in the ith row and jth column is x_{ij}, so the value

$$t\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pp} \end{pmatrix}$$

is obtained by applying s to the column vectors and then s to the resulting row vector.

We introduce several concepts for zero-one matrices, since only 0-1 values for the variables in t will play a role in the proof.

Definition 8.5. Let $X = (x_{ij})$, Y be $p \times p$ zero-one matrices. The *area of* X is the fraction of ones and is denoted

$$\lambda(X) = \left(\sum_{i,j} x_{ij}\right) / p^2.$$

The matrices X, Y are called *g-equivalent*, denoted $X \sim Y$, if $g(t(X)) = g(t(Y))$. The matrix X is called *tame* if

$$\begin{aligned} \text{either} \quad & X \sim 0_{p\times p} \quad \text{and} \quad \lambda(X) < 1/3, \\ \text{or} \quad & X \sim 1_{p\times p} \quad \text{and} \quad \lambda(X) > 1/3, \end{aligned}$$

where $0_{p\times p}$ stands for the zero matrix and $1_{p\times p}$ for the all-ones matrix.

Observe that the equivalence $\sim$ has two blocks, so, e.g., $X \nsim Y \nsim Z$ implies $X \sim Z$. Also recall that $p > 3$ is a prime number, so the area of X is never equal to $1/3$.

The proof now proceeds as follows: We show that certain matrices, called "almost rectangles," are tame. The proof is by induction (although the proof logic, as presented, is a bit different). Section 8.2 provides the base case, and Section 8.3 handles the induction step. In Section 8.4, we construct two tame matrices X_1, X_2 such that $\lambda(X_1) < 1/3$ and $\lambda(X_2) > 1/3$, but $t(X_1) = t(X_2)$ (because the corresponding columns of X_1 and X_2 will be evaluated by s to the same elements). This gives us a contradiction, since $0_{p\times p} \nsim 1_{p\times p}$, as we shall see.

Before launching into the technicalities, we introduce an additional concept and state a consequence of the fact that s is a polymorphism.

Definition 8.6. A triple X, Y, Z of $p \times p$ zero-one matrices is called a *cover* if, for every $1 \leq i, j \leq p$, exactly one of x_{ij}, y_{ij}, z_{ij} is equal to one.

Lemma 8.7. *If X, Y, Z is a cover, then X, Y, Z are not all g-equivalent.*

Proof. By the definition of a cover, the triple (x_{ij}, y_{ij}, z_{ij}) is in $\{(0,0,1), (0,1,0), (1,0,0)\} \subseteq R$ for each i, j. Since t preserves R (because s does), the triple $(t(X), t(Y), t(Z))$ is in R as well. Finally, g is a homomorphism from $\mathbf{D}$ to $\mathbf{H}_2$, therefore $g(t(X)), g(t(Y)), g(t(Z))$ are not all equal. In other words, X, Y, Z are not all g-equivalent, as claimed. □

8.2 Line Segments are Tame

In this subsection it will be more convenient to regard the arguments of t as a tuple $\mathbf{x} = (x_{11}, x_{12}, \ldots)$ of length p^2 rather than a matrix. The concepts of the area, g-equivalence, tameness, and cover are extended to tuples in the obvious way. Since $p > 3$ is a prime number, p^2 is 1 modulo 3. Let q be such that

$$p^2 = 3q + 1.$$

Moreover, let $\langle i \rangle$ denote the following tuple of length p^2:

$$\langle i \rangle = (\underbrace{1, \ldots, 1}_{i \text{ times}}, 0, \ldots 0).$$

We prove in this subsection that all such tuples are tame. We first recall a well-known fact.

Lemma 8.8. *The operation t is cyclic.*

Proof. By cyclically shifting the arguments, we get the same result:

$$t(x_{12}, \ldots, x_{pp}, x_{11}) = t\begin{pmatrix} x_{12} & x_{13} & \cdots & x_{1p} & x_{21} \\ x_{22} & x_{23} & \cdots & x_{2p} & x_{31} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{p2} & x_{p3} & \cdots & x_{pp} & x_{11} \end{pmatrix} = t\begin{pmatrix} x_{21} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{31} & x_{22} & x_{23} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{11} & x_{p2} & x_{p3} & \cdots & x_{pp} \end{pmatrix}$$
$$= t\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pp} \end{pmatrix} = t(x_{11}, x_{12}, \ldots, x_{pp}),$$

where the second equality uses the cyclicity of the outer "s" in the definition of t, while the third one the cyclicity of the first inner "s." □

Lemma 8.9. $\langle 0 \rangle \sim \langle 1 \rangle \sim \cdots \sim \langle q \rangle \nsim \langle q+1 \rangle \sim \cdots \sim \langle 2q \rangle \sim \langle 2q+1 \rangle$.

Proof. By induction on $i = 0, 1, \ldots, q$, we prove

$$\langle q-i \rangle \sim \langle q-i+1 \rangle \sim \cdots \sim \langle q \rangle \nsim \langle q+1 \rangle \sim \cdots \sim \langle q+i \rangle \sim \langle q+i+1 \rangle.$$

For the induction base, $i = 0$, let $\mathbf{x} = \langle q \rangle$, let $\mathbf{y}$ be $\langle q \rangle$ (cyclically) shifted q times to the right (so the first 1 is at the $(q+1)$-st position), and let $\mathbf{z}$ be $\langle q+1 \rangle$ shifted $2q$ times to the right. The tuples $\mathbf{x}, \mathbf{y}, \mathbf{z}$ form a cover, therefore they are not all g-equivalent by Lemma 8.7. But t is cyclic, thus $t(\mathbf{x}) = t(\mathbf{y}) = t(\langle q \rangle)$ and $t(\mathbf{z}) = t(\langle q+1 \rangle)$. It follows that $\langle q \rangle, \langle q \rangle, \langle q+1 \rangle$ are not all g-equivalent and we get $\langle q \rangle \nsim \langle q+1 \rangle$.

Now, we prove the claim for $i > 0$ assuming it holds for $i-1$. To verify $\langle q-i \rangle \sim \langle q-i+1 \rangle$ consider $\langle q-i \rangle, \langle q+1 \rangle, \langle q+i \rangle$. Since $(q-i)+(q+1)+(q+i) = 3q+1 = p^2$, these tuples can be cyclically shifted to form a cover and then the same argument as above gives us that $\langle q-i \rangle$, $\langle q+1 \rangle$, $\langle q+i \rangle$ are not all g-equivalent. But $\langle q+1 \rangle \sim \langle q+i \rangle$ by the induction hypothesis, therefore $\langle q-i \rangle \nsim \langle q+1 \rangle$. Since $\langle q+1 \rangle \nsim \langle q-i+1 \rangle$ (again, by the induction hypothesis), we get $\langle q-i \rangle \sim \langle q-i+1 \rangle$, as required. It remains to check $\langle q+i \rangle \sim \langle q+i+1 \rangle$. This is done in a similar way, using the tuples $\langle q-i \rangle$, $\langle q \rangle$, $\langle q+i+1 \rangle$. □

We have proved that $\langle 0 \rangle \sim \cdots \sim \langle q \rangle \nsim \langle q+1 \rangle \sim \cdots \sim \langle 2q+1 \rangle$. Using the same argument as in the previous lemma once more for $\langle 0 \rangle, \langle p^2 - i \rangle, \langle i \rangle$ with $p^2 \geq i > 2q+1$, we get $\langle i \rangle \nsim \langle 0 \rangle$. In summary, $\langle i \rangle \sim \langle 0 \rangle$ whenever $i \leq q$ and $\langle i \rangle \sim \langle p^2 \rangle \nsim \langle 0 \rangle$ when $i \geq q+1$. Observing that $\lambda(\langle i \rangle) < 1/3$ if and only if $i \leq q$, we obtain the following lemma:

Lemma 8.10. *Each $\langle i \rangle$, $i \in \{0, 1, \ldots, p^2\}$, is tame and $\langle 0 \rangle \nsim \langle p^2 \rangle$.*

8.3 Almost Rectangles Are Tame

We start by introducing a special type of zero-one matrices.

Definition 8.11. Let $1 \leq k_1, \dots, k_p \leq p$. By $[k_1, k_2, \dots, k_p]$, we denote the matrix whose ith column begins with k_i ones followed by $(p - k_i)$ zeros, for each $i \in \{1, \dots, p\}$.

An *almost rectangle* is a matrix of the form $[k, k, \dots, k, l, l, \dots, l]$ (the number of k's can be arbitrary, including 0 or p) where $0 \leq k - l \leq 5|D|$. The quantity $k - l$ is referred to as the *size of the step.*

In the remainder of this subsection, we prove the following proposition:

Proposition 8.12. *Each almost rectangle is tame.*

Let

$$X = [\underbrace{k, \dots, k}_{m \text{ times}}, l, \dots, l]$$

be a minimal counterexample in the following sense:

- X has the minimum size of the step and,
- among such counterexamples, $|\lambda(X) - 1/3|$ is maximal.

Lemma 8.13. *The size of the step of X is at least 2.*

Proof. This lemma is just a different formulation of Lemma 8.10, since an almost rectangle with step of size 0 or 1 represents the same choice of arguments as $\langle i \rangle$ for some i. □

We handle two cases $\lambda(X) \geq 5/12$ and $\lambda(X) \leq 5/12$ separately, but the basic idea for both of them is the same as in the proof of Lemma 8.10. To avoid puzzling the reader, let us remark that any number strictly between 1/3 and 1/2 (instead of 5/12) would work with a sufficiently large p.

The following lemma deals with the first case:

Lemma 8.14. *The area of X is less than* 5/12.

Proof. Assume that $\lambda(X) \geq 5/12$. Let k_1, k_2, l_1, and l_2 be the non-negative integers such that

$$l_1 + l_2 + k = p = k_1 + k_2 + l, \tag{8.1}$$

$$1 \geq k_1 - k_2 \geq 0, \text{ and } 1 \geq l_1 - l_2 \geq 0. \tag{8.2}$$

We have $k_1 \geq l_1$ and $k_2 \geq l_2$. Moreover, since $k - l \geq 2$ by the previous lemma, it follows that both $k_1 - l_1$ and $k_2 - l_2$ are strictly smaller than $k - l$.

Consider the matrices

$$Y_i = [\underbrace{l_i, \dots, l_i}_{m \text{ times}}, k_i, \dots, k_i], \quad i = 1, 2.$$

By shifting all the rows of Y_i, $i \in \{1, 2\}$, m times to the left, we obtain an almost rectangle with a smaller step size than X, which is thus tame by the minimality assumption on X. Since such a shift changes neither the value of t (as the outer "s" in the definition of t is cyclic) nor the area, both Y_1 and Y_2 are tame matrices.

Let Y_1' (Y_2', respectively) be the matrices obtained from Y_1 (Y_2, respectively) by shifting the first m columns k times ($k + l_1$ times, respectively) down and the remaining columns l times ($l + k_1$ times, respectively) down. Since X, Y_1', Y_2' is a cover (by Equation (8.1)) and cyclically shifting columns does not change the value of t (as the inner occurrences of "s" in the definition of t are cyclic), Lemma 8.7 implies that X, Y_1, Y_2 are not all g-equivalent.

From X, Y_1', Y_2' being a cover, it also follows that

$$\lambda(X) + \lambda(Y_1') + \lambda(Y_2') = \lambda(X) + \lambda(Y_1) + \lambda(Y_2) = 1.$$

Moreover, by Equation (8.2), we have $\lambda(Y_2) \leq \lambda(Y_1)$ and these areas differ by at most $p/p^2 = 1/p$. Therefore

$$\lambda(Y_1) = 1 - \lambda(X) - \lambda(Y_2) \leq 1 - 5/12 - \lambda(Y_1) + 1/p$$

and, since $p > 12$ by the choice of p, we obtain

$$\lambda(Y_2) \leq \lambda(Y_1) < 1/3.$$

The tameness of Y_i now gives us $Y_1 \sim Y_2 \sim 0_{p\times p}$ and then, since Y_1, Y_2, X are not all g-equivalent and $0_{p\times p} \nsim 1_{p\times p}$ (by the second part of Lemma 8.10), we get $X \sim 1_{p\times p}$. But $\lambda(X) \geq 5/12 > 1/3$, hence X is tame, a contradiction with the choice of X. □

It remains to handle the case $\lambda(X) < 5/12$.

We first claim that $2k$ (and thus $k + l$ and $2l$) is less than p. Indeed, since the step size of X is at most $5|D|$ (by the definition of an almost rectangle) and $p > 60|D|$, we get

$$5/12 > \lambda(X) \geq p(k - 5|D|)/p^2, \text{ and hence}$$
$$k \leq 5p/12 + 5|D| < 5p/12 + p/12 = p/2.$$

We now again need to distinguish two cases. Assume first that $m < p/2$.

Let

$$Y = [\underbrace{l, \ldots, l}_{m \text{ times}}, \underbrace{k, \ldots, k}_{m \text{ times}}, l, \ldots, l],$$
$$Z = [\underbrace{p - k - l, \ldots, p - k - l}_{2m \text{ times}}, p - 2l, \ldots, p - 2l].$$

The definition of Z makes sense, since $p - k - l, p - 2l \geq 0$ by the inequality $2k < p$ derived above.

The triple X, Y, Z (similarly to X, Y_1, Y_2 in the proof of Lemma 8.14) is such that we can obtain a cover by shifting the columns down. Therefore, X, Y, Z are not all g-equivalent and $\lambda(X) + \lambda(Y) + \lambda(Z) = 1$.

However, by shifting all the rows of Y m times to the left, we obtain X. We get $\lambda(X) = \lambda(Y)$ and $t(X) = t(Y)$, therefore $Z \nsim X$ by the previous paragraph.

Moreover, by shifting all the rows of Z $2m$ times to the left, we obtain an almost rectangle Z' with $t(Z) = t(Z')$ and $\lambda(Z) = \lambda(Z')$. The step size of Z' is $(p - 2l) - (p - k - l) = k - l$, which is the same as the step size of X. However, the distance of its area from $1/3$ is strictly greater as shown by the following calculation:

$$\frac{|\lambda(Z) - 1/3|}{|\lambda(X) - 1/3|} = \frac{|(1 - 2\lambda(X)) - 1/3|}{|\lambda(X) - 1/3|} = \frac{|2(1/3 - \lambda(X))|}{|\lambda(X) - 1/3|} = 2 > 1.$$

By the minimality of X, the almost rectangle Z' is tame and so is Z. It is also apparent from the calculation that the signs of $\lambda(X) - 1/3$ and $\lambda(Z) - 1/3$ are opposite. Combining these two facts with $Z \nsim X$ derived above, we obtain that X is tame, a contradiction.

In the other case, when $m > p/2$, the proof is similar using the tuples

$$Y = (l, \ldots, l, \underbrace{k, \ldots, k}_{m \text{ times}}),$$
$$Z = (\underbrace{p - k - l, \ldots, p - k - l}_{(p-m) \text{ times}}, p - 2k, \ldots, p - 2k, \underbrace{p - k - l, \ldots, p - k - l}_{(p-m) \text{ times}}).$$

The proof of Proposition 8.12 is concluded.

8.4 Contradiction

Let $m = (p-1)/2$ and choose natural numbers l_1 and l_2 so

$$p/3 - 2|D| < l_1 < l_2 < p/3$$

and

$$s(\underbrace{1,\dots,1}_{l_1 \text{ times}},0,\dots,0) = s(\underbrace{1,\dots,1}_{l_2 \text{ times}},0,\dots,0).$$

This is possible by the pigeonhole principle, since there are $2|D| > |D|$ integers in the interval and $p/3 - 2|D| > 0$ by the choice of p.

The required contradiction will be obtained by considering the two matrices

$$X_i = [\underbrace{k,\dots,k}_{m \text{ times}}, l_i,\dots,l_i], \quad i \in \{1,2\},$$

where k will be specified soon.

Before choosing k, we observe that $t(X_1) = t(X_2)$. Indeed, the first m columns of these matrices are the same (and thus so are their images under s) and the remaining columns have the same image under s by the choice of l_1 and l_2. The claim thus follows from the definition of t.

Next, note that for $k \leq p/3$ the area of both matrices is less than $1/3$, since $l_i < p/3$. However, for $k \geq p/3 + 3|D|$ the area is greater:

$$\lambda(X_i) = \frac{mk + (p-m)l_i}{p^2} \geq \frac{\frac{p-1}{2}(p/3 + 3|D|) + \frac{p+1}{2}(p/3 - 2|D|)}{p^2} = \frac{p^2/3 + |D|(p-5)/2}{p^2} > 1/3.$$

Choose the maximum k so $\lambda(X_1) < 1/3$. The derived inequalities and the choice of l_i implies

$$l_1 < l_2 \leq k < p/3 + 3|D| \leq l_1 + 5|D| < l_2 + 5|D|,$$

therefore both X_1 and X_2 are almost rectangles. By Proposition 8.12, X_1 and X_2 are tame.

Since the area of X_1 is less than $1/3$, we get $X_1 \sim 0_{p\times p}$. We chose k so increasing k by 1 makes the area of X_1 greater than $1/3$. From $m < p/2$ it follows that increasing l_1 by 1 makes the area even greater, hence $\lambda(X_2) > 1/3$ (recall that $l_2 > l_1$) and we obtain $X_2 \sim 1_{p\times p}$.

Recall that $0_{p\times p} \nsim 1_{p\times p}$ by the second part of Lemma 8.10. Therefore $X_1 \nsim X_2$, contradicting $t(X_1) = t(X_2)$ and concluding the proof of Theorem 8.1.

Remark 8.15. The proof of Theorem 8.1 could be simplified if we had stronger or more suitable polymorphisms than cyclic operations. Alternative versions of Theorem 8.3 could also help in simplifying the proof of the CSP dichotomy conjecture. In particular, the following question seems open: Does every finite $\mathbf{D}$ with a cyclic polymorphism of arity at least 2 necessarily have a polymorphism s of arity $n > 1$ such that, for any $a, b \in D$ and $(x_1,\dots,x_n) \in \{a,b\}^n$, the value $s(x_1,\dots,x_n)$ depends only on the number of occurrences of a in $(x_1,\dots,x_n)$? Note that a more optimistic version involving evaluations with $|\{x_1,\dots,x_n\}| = 3$ is false; a counterexample is the disjoint union of a directed 2-cycle and a directed 3-cycle.

9 ALGEBRAIC CONSTRUCTIONS

The main goal of this section is to provide a more detailed picture of the algebraic theory of minions and to align the algebraic approach to PCSP more closely with the standard theory as presented in Reference [15]. The theory that we present is a natural generalisation of Reference [16] from clones to minions. We note that, as in Reference [16], many of these results can be generalised to minions over infinite sets. Nevertheless, we keep our focus on the case of finite domains.

Let us start with describing algebraic counterparts of pp-constructions.

Definition 9.1. Let $\mathscr{M}_1$ and $\mathscr{M}_2$ be two minions on (A_1, B_1) and (A_2, B_2), respectively. We use the following terminology:

- $\mathscr{M}_2$ is a *reflection* of $\mathscr{M}_1$ if there are maps $h_A : A_2 \to A_1$ and $h_B : B_1 \to B_2$ such that
$$\mathscr{M}_2 = \{(x_1, \dots, x_{arf}) \mapsto h_B(f(h_A(x_1), \dots, h_A(x_{arf}))) \mid f \in \mathscr{M}_1\},$$
- $\mathscr{M}_2$ is an *expansion* of $\mathscr{M}_1$ if $A_1 = A_2$, $B_1 = B_2$, and $\mathscr{M}_1 \subseteq \mathscr{M}_2$, and
- Let N be a set, we say that $\mathscr{M}_2$ is the N*th power* of $\mathscr{M}_1$, if $A_2 = A_1^N$, $B_2 = B_1^N$, and $\mathscr{M}_2$ is the set of coordinate-wise actions of all functions in $\mathscr{M}_1$. The $[n]$th power (usually referred to as n*th power* and denoted $\mathscr{M}_2 = \mathscr{M}_1^n$) is described as the set of all functions of the form
$$((x_{11}, \dots, x_{1n}), \dots, (x_{t1}, \dots, x_{tn})) \mapsto (f(x_{11}, \dots, x_{t1}), \dots, f(x_{1n}, \dots, x_{tn})),$$
where $t \geq 1$ and f is a t-ary function from $\mathscr{M}_1$. In this case, we say that $\mathscr{M}_2$ is a *finite power of* $\mathscr{M}_1$.

Following the standard notation, we denote the class of all reflections of $\mathscr{M}$ by $\mathsf{R}\mathscr{M}$, the class of all expansions by $\mathsf{E}\mathscr{M}$, the class of all powers by $\mathsf{P}\mathscr{M}$, and the class of all finite powers by $\mathsf{P}_{\text{fin}}\mathscr{M}$.

We remark that a reflection of a minion is always a minion. This is in contrast with a reflection of a clone (as defined in Reference [16, Definition 4.3]), which is not necessarily a clone (see Reference [16, p. 379]).

The following lemma relates the above to the relational constructions from Sections 2.3 and 4.2:

Lemma 9.2. *Let* $(\mathbf{A}, \mathbf{B})$ *and* $(\mathbf{A}', \mathbf{B}')$ *be two templates. Denote their polymorphism minions by* $\mathscr{M}$ *and* $\mathscr{M}'$*, respectively.*

(1) $(\mathbf{A}', \mathbf{B}')$ *is ppp-definable from* $(\mathbf{A}, \mathbf{B})$ *if and only if* $\mathscr{M}' \in \mathsf{E}\mathscr{M}$*.*
(2) $(\mathbf{A}', \mathbf{B}')$ *is a simple relaxation of a pp-power of* $(\mathbf{A}, \mathbf{B})$ *if and only if* $\mathscr{M}' \in \mathsf{E}\mathsf{P}_{\text{fin}}\mathscr{M}$*.*
(3) $(\mathbf{A}', \mathbf{B}')$ *is a relaxation of a structure that is pp-definable from* $(\mathbf{A}, \mathbf{B})$ *if and only if* $\mathscr{M}' \in \mathsf{E}\mathsf{R}\mathscr{M}$*.*

Proof. Item (1) is proved in References [27, 83] (see also Theorem 2.26).

The proof of item (2) is similar to Reference [9, Proposition 3.1]. Let us first define templates $(\mathbf{A}_n, \mathbf{B}_n)$ that will have the property that all nth pp-powers of $(\mathbf{A}, \mathbf{B})$ are pp-definable in $(\mathbf{A}_n, \mathbf{B}_n)$. We set $A_n = A^n$, $B_n = B^n$, and for each relation R of $(\mathbf{A}, \mathbf{B})$ of arity k and each tuple $(i_1, \dots, i_k) \in [n]^k$, we define a new k-ary relation $R_{i_1 \dots i_k}$ of $(\mathbf{A}_n, \mathbf{B}_n)$ by
$$((a_{11}, \dots, a_{1n}), \dots, (a_{k1}, \dots, a_{kn})) \in R_{i_1 \dots i_k}^{\mathbf{A}_n} \text{ if } (a_{1i_1}, \dots, a_{ki_k}) \in R^{\mathbf{A}},$$
and similarly for $R_{i_1 \dots i_k}^{\mathbf{B}_n}$. Further, for all $i, j \in [n]$, we add a new relation symbol E_{ij} interpreted as $((a_1, \dots, a_n), (b_1, \dots, b_n)) \in E_{ij}^{\mathbf{A}_n}$ if $a_i = b_j$, and similarly for $E_{ij}^{\mathbf{B}_n}$. It follows directly from definitions that any structure that is pp-definable in $(\mathbf{A}_n, \mathbf{B}_n)$ is an nth pp-power of $(\mathbf{A}, \mathbf{B})$ and vice versa. Further, we claim that $\mathscr{M}^n = \operatorname{Pol}(\mathbf{A}_n, \mathbf{B}_n)$. It is straightforward to verify that $\mathscr{M}^n \subseteq \operatorname{Pol}(\mathbf{A}_n, \mathbf{B}_n)$. For the reverse inclusion, first observe that every polymorphism of $(\mathbf{A}_n, \mathbf{B}_n)$ acts independently on each of the coordinates, as follows from the compatibility with the relations E_{ii}, and the actions on different coordinates are identical, as follows from the compatibility with the relations E_{ij} for $i \neq j$. From the compatibility with the relations $R_{i_1 \dots i_k}$, we obtain that the action on each coordinate is a polymorphism of $(\mathbf{A}, \mathbf{B})$. Now, assume that $(\mathbf{A}', \mathbf{B}')$ is a simple relaxation of the nth pp-power of $(\mathbf{A}, \mathbf{B})$, in particular it is ppp-definable in $(\mathbf{A}_n, \mathbf{B}_n)$. Thus, $\mathscr{M}' \in \mathsf{E}\mathscr{M}^n \subseteq \mathsf{E}\mathsf{P}_{\text{fin}}\mathscr{M}$. The converse is also proved using the template $(\mathbf{A}_n, \mathbf{B}_n)$. Given that $\mathscr{M}' \in \mathsf{E}\mathsf{P}_{\text{fin}}\mathscr{M}$, there is n such that $\mathscr{M}' \in \mathsf{E}\mathscr{M}^n$. The rest again follows from the first item.

To prove item (3), assume that $(\mathbf{A}', \mathbf{B}')$ is a relaxation of a template $(\mathbf{A}_0, \mathbf{B}_0)$ that is pp-definable from $(\mathbf{A}, \mathbf{B})$, and let $h_A : \mathbf{A}' \to \mathbf{A}_0$ and $h_B : \mathbf{B}_0 \to \mathbf{B}'$ be the relaxing homomorphisms. We claim that all mappings of the form $(x_1, \dots, x_n) \to h_B(f(h_A(x_1), \dots, h_A(x_n)))$, where $f \in \mathscr{M}$, are polymorphisms from $\mathbf{A}'$ to $\mathbf{B}'$. Since $(\mathbf{A}_0, \mathbf{B}_0)$ is pp-definable in $(\mathbf{A}, \mathbf{B})$, each $f \in \mathscr{M}$ is also a polymorphism from $\mathbf{A}_0$ to $\mathbf{B}_0$. Consequently, we can prove that the above composition is a polymorphism from $\mathbf{A}'$ to $\mathbf{B}'$ in the same way as proving that a composition of homomorphisms of relational structures is a homomorphism. This concludes that $\mathscr{M}' \in \mathsf{ER}\mathscr{M}$.

For the other implication, suppose that $\mathscr{M}' \in \mathsf{ER}\mathscr{M}$, and let the reflection be defined by mappings h_A and h_B. We define an intermediate template $(\mathbf{A}_0, \mathbf{B}_0)$ by putting $A_0 = A$, $B_0 = B$, $R^{\mathbf{A}_0} = h_A(R^{\mathbf{A}'})$, and $R^{\mathbf{B}_0} = h_B^{-1}(R^{\mathbf{B}'})$ for all R. This definition ensures that $(\mathbf{A}', \mathbf{B}')$ is a relaxation of $(\mathbf{A}_0, \mathbf{B}_0)$. To prove that $(\mathbf{A}_0, \mathbf{B}_0)$ is ppp-definable in $(\mathbf{A}, \mathbf{B})$, it is enough to show that $\mathscr{M} \subseteq \operatorname{Pol}(\mathbf{A}_0, \mathbf{B}_0)$; the rest follows from item (1).

Let $f \in \mathscr{M}$ be of arity n and $\mathbf{c}_1, \dots, \mathbf{c}_n \in R^{\mathbf{A}_0}$. From the definition of $R^{\mathbf{A}_0}$, we get $\mathbf{b}_1, \dots, \mathbf{b}_n \in R^{\mathbf{A}'}$ such that $\mathbf{c}_i = h_A(\mathbf{b}_i)$ for each $i = 1, \dots, n$. Now,

$$h_B(f(h_A(\mathbf{b}_1), \dots, h_A(\mathbf{b}_n))) \in R^{\mathbf{B}'},$$

since the composition $h_B(f(h_A, \dots, h_A)) \in \mathscr{M}'$, and consequently

$$f(\mathbf{c}_1, \dots, \mathbf{c}_n) = f(h_A(\mathbf{b}_1), \dots, h_A(\mathbf{b}_n)) \in h_B^{-1}(R^{\mathbf{B}'}) = R^{\mathbf{B}_0}.$$

This shows that $(\mathbf{A}', \mathbf{B}')$ is a relaxation of a structure that is ppp-definable from $(\mathbf{A}, \mathbf{B})$, therefore it is pp-constructible from $(\mathbf{A}, \mathbf{B})$, which is the same as being a relaxation of a structure that is pp-definable from $(\mathbf{A}, \mathbf{B})$ by Theorem 4.12. □

The following theorem is an algebraic version of Theorem 4.12. Its proof is therefore very similar, and a connection to constructions from Section 4 should be immediate.

Theorem 9.3. *Let $\mathscr{M}$ and $\mathscr{M}'$ be two minions, respectively. Then $\mathscr{M}' \in \mathsf{ERP}_{\text{fin}}\mathscr{M}$ if and only if there exists a minion homomorphism from $\mathscr{M}$ to $\mathscr{M}'$.*

Proof. Assume that $\mathscr{M} \subseteq \mathscr{O}(A, B)$ and $\mathscr{M}' \subseteq \mathscr{O}(A', B')$. If $\mathscr{M}' \in \mathsf{ERP}_{\text{fin}}\mathscr{M}$, then there is a natural mapping $\xi : \mathscr{M} \to \mathscr{M}'$ that is clearly a minion homomorphism.

For the converse, suppose that $\xi : \mathscr{M} \to \mathscr{M}'$ is a minion homomorphism. We will use this mapping to find a reflection of a suitable power of $\mathscr{M}$ that gives a subset of $\mathscr{M}'$. A suitable exponent of the power is $N = A^{A'}$. For the reflection, we need to define two mappings, $h_A : A' \to A^N$ and $h_B : B^N \to B'$. We will choose them in such a way that the following holds:

$$h_B\big(f^{\mathscr{M}^N}(h_A(x_1), \dots, h_A(x_n))\big) = \xi(f)(x_1, \dots, x_n) \tag{$\diamond$}$$

for all $f \in \mathscr{M}$ and $x_1, \dots, x_n \in A'$. First, we define $h_A : A' \to A^N$ as the long code encoding of an element of A' over the alphabet A. More formally, we put $h_A(a') = p_{a'}$ where $p_{a'} : A^{A'} \to A$ is the a'th projection (i.e., $p_{a'}(f) = f(a')$).

To define the mapping $h_B : B^N \to B'$, we view an element of B^N of the form $f^{\mathscr{M}^N}(h_A(a'_1), \dots, h_A(a'_n))$ as a function from $N = A^{A'}$ to B that is obtained as the composition of the projection functions $p_{a'_i} : A^{A'} \to A$, for $i \in [n]$, with $f : A^n \to B$. Note that this function has at most n essential variables, because every $p_{a'_i}$ depends only on the coordinate a'_i. If $b \in B^N$ can be expressed as $b = f(p_{a'_1}, \dots, p_{a'_n})$, then we define $h_B(b) = \xi(f)(a'_1, \dots, a'_n)$, and for b for which such f and $a'_1, \dots, a'_n$ do not exist, we define h_B arbitrarily.

We need to show that h_B is well-defined, i.e., if $f(a'_1, \dots, a'_n) = g(b'_1, \dots, b'_m)$, then $\xi(f)(a'_1, \dots, a'_n) = \xi(g)(b'_1, \dots, b'_m)$. But this is true, since the premise gives a minor identity between f and g and ξ preserves such identities. It is also easy to see that h_A and h_B satisfy ($\diamond$). Thus, we obtain that $\mathscr{M}' \in \mathsf{ER}\mathscr{M}^N$, which gives $\mathscr{M}' \in \mathsf{ERP}_{\text{fin}}\mathscr{M}$. □

Remark 9.4. The finiteness of sets B and B' is not used in the proof above, and therefore the theorem is true as stated as long as both A and A' are finite. Dropping even that assumption, following the same proof, one can show that the statement is true for minions over infinite sets if P_{fin} is replaced by P.

Finally, returning to polymorphism minions of PCSP templates, we can relate the above to Theorem 4.12.

Corollary 9.5. *Let $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}', \mathbf{B}')$ be two templates and let $\mathscr{M}$ and $\mathscr{M}'$, respectively, be their polymorphism minions. The following are equivalent:*

(1) $(\mathbf{A}', \mathbf{B}')$ is pp-constructible from $(\mathbf{A}, \mathbf{B})$ (consequently, PCSP$(\mathbf{A}', \mathbf{B}')$ reduces in log-space to PCSP$(\mathbf{A}, \mathbf{B})$).
(2) $\mathscr{M}' \in \mathsf{ERP}_{\text{fin}}\mathscr{M}$.
(3) There exists a minion homomorphism $\xi : \mathscr{M} \to \mathscr{M}'$.

Proof. The equivalence of (2) and (3) is given by the previous theorem. The equivalence of (1) and (3) follows from Theorem 4.12. Alternatively, we can argue that (1) is equivalent with the rest of the items in the following way: (1) → (3) is the easier direction, it follows easily from Lemma 4.8 (this mirrors the proof of Theorem 4.12). We prove the other implication by proving that (2) implies (1): Since $\mathscr{M}' \in \mathsf{ERP}_{\text{fin}}\mathscr{M}$ there is a minion $\mathscr{M}'' \in \mathsf{P}_{\text{fin}}\mathscr{M}$, say that $\mathscr{M}'' = \mathscr{M}^n$, such that $\mathscr{M}' \in \mathsf{ER}\mathscr{M}''$. Furthermore, this minion is the polymorphism minion of the template $(\mathbf{A}_n, \mathbf{B}_n)$ introduced in the proof of Lemma 9.2(2). Finally, Lemma 9.2(3) concludes that $(\mathbf{A}', \mathbf{B}')$ is a relaxation of a structure pp-definable from $(\mathbf{A}_n, \mathbf{B}_n)$, and therefore it is pp-constructible from $(\mathbf{A}, \mathbf{B})$. (In fact, this directly shows that $(\mathbf{A}', \mathbf{B}')$ is a relaxation of a pp-power of $(\mathbf{A}, \mathbf{B})$.) □

10 BIPARTITE MINOR CONDITIONS SATISFIED IN Pol($\mathbf{K}_k, \mathbf{K}_c$)

The key in our approach to the complexity of PCSPs is the analysis of bipartite minor conditions satisfied in their polymorphism minions. While we cannot yet fully resolve the complexity of approximate graph colouring, in this section, we provide some analysis and comparison of bipartite minor conditions satisfied in Pol$(\mathbf{K}_k, \mathbf{K}_c)$ for various k and c. We remark that all hard CSPs satisfy the same (i.e., only trivial) bipartite minor conditions, but this is not the case for PCSPs, as seen from our results in this section.

The hardness of PCSP$(\mathbf{K}_k, \mathbf{K}_c)$ with $c \leq 2k - 2$ was proved in Reference [26], essentially by showing that, for such c, Pol$(\mathbf{K}_k, \mathbf{K}_c)$ has a minion homomorphism to the minion $\mathscr{P}_2$ of projections (though they did not use this terminology). In other words, all bipartite minor conditions satisfied in Pol$(\mathbf{K}_k, \mathbf{K}_c)$ for such k and c are trivial. It was noted in Reference [26] that their approach cannot be extended to $c \geq 2k - 1$, because then the polymorphisms satisfy some non-trivial minor conditions (we gave a specific example above; see Example 3.4). We settled the case $c = 2k - 1$ by proving that the bipartite minor conditions satisfied in Pol$(\mathbf{K}_k, \mathbf{K}_c)$, though possibly non-trivial, are limited. Specifically, all such conditions are also satisfied in Pol$(\mathbf{H}_2, \mathbf{H}_K)$ for some K, because Pol$(\mathbf{K}_k, \mathbf{K}_{2k-1})$ does not contain an Olšák function. We now show that this does not hold for $c \geq 2k$.

Proposition 10.1. Pol$(\mathbf{K}_k, \mathbf{K}_{2k})$ *contains an Olšák function.*

Proof. Define a 6-ary operation o from K_k to K_{2k} in the following way:

$$o(x_1, \dots, x_6) = \begin{cases} a & \text{if at least two of } x_1, x_2, x_3 \text{ are equal to } a, \\ x_1 + k & \text{otherwise.} \end{cases}$$

Note that o depends essentially only on its first three variables. It is straightforward to check that o is an Olšák function. To see that it is a polymorphism suppose that $x_1 \neq y_1, \dots, x_6 \neq y_6$ while

$o(x_1, \dots, x_6) = o(y_1, \dots, y_6) = a$. If a is obtained using the last case of the definition of o, then $a = x_1 + k = y_1 + k$, and consequently $x_1 = y_1$, a contradiction. If $a \in K_k$, then only the first row of the definition had to be used to compute the value for both $o(x_1, \dots, x_6)$ and $o(y_1, \dots, y_6)$ but, since any two of two-element subsets of $\{1, 2, 3\}$ intersect, we get that $a = x_i = y_i$ for any i in the intersection, a contradiction again. □

We now show that Theorem 5.9 cannot be directly applied to to prove hardness of PCSP$(\mathbf{K}_k, \mathbf{K}_{2k})$.

Proposition 10.2. *For any $\varepsilon > 0$, $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k})$ satisfies some ε-robust bipartite minor condition.*

Proof. Let m be the ternary Boolean majority function, i.e., $m(x, y, z)$ outputs the repeated value among x, y, z. Further, let $m_1 = m$ and, for $n \geq 1$, define

$$m_n(x_1, \dots, x_{3^n}) = m_{n-1}(m(x_1, x_2, x_3), \dots, m(x_{3^n-2}, x_{3^n-1}, x_{3^n})).$$

For example, $m_2(x_1, \dots, x_9) = m(m(x_1, x_2, x_3), m(x_4, x_5, x_6), m(x_7, x_8, x_9))$.

For a subset $I \subseteq \{1, \dots, 3^n\}$, let $\mathbf{x}_I$ be the tuple in $\{0, 1\}^{3n}$ such that $x_i = 1$ if and only if $i \in I$. Let $S_n = \{I \subseteq \{1, \dots, 3^n\} \mid m_n(\mathbf{x}_I) = 1\}$, and let MS_n be the set of all minimal (under inclusion) sets in S_n. Let Σ_n be the bipartite minor condition that contains, for each $I \in MS_n$, the identity

$$f_n(x_0, y_1, \dots, y_\ell) \approx g_n(x_{\pi_I(1)}, \dots, x_{\pi_I(3^n)})$$

where

- $y_1, \dots, y_\ell$ is the list of all variables of the form $x_{(J,i)}$ with $J \in MS_n$ and $1 \leq i \leq 3^n$ such that $i \notin J$, and
- $\pi_I(i) = 0$ if $i \in I$ and $\pi_I(i) = (I, i)$ otherwise.

Note that the function symbols f_n and g_n are the same in all identities in Σ_n.

Fix any $\varepsilon > 0$ and choose n so $\varepsilon > (2/3)^n$. We will show that Σ_n is ε-robust. Note that each variable of the form $x_{(I,i)}$ appears on the right-hand side of only one identity in Σ_n (namely, the one corresponding to I). Assume first that f_n is assigned a projection not on the first variable (i.e., not on x_0). Then this projection corresponds to some variable of the form $x_{(I,i)}$ on the left-hand side, and any assignment of a projection to g_n would satisfy at most one identity in Σ_n (again, the one corresponding to I), which is certainly less than an ε-fraction of all identities in Σ_n. So assume that f_n is assigned a projection on the first variable x_0. It is not hard to check that, for a fixed $1 \leq i \leq 3^n$, the probability that $\pi_I(i) = 0$ (i.e., that $i \in I$) for a randomly chosen set $I \in MS_n$ is $(2/3)^n$. Thus, any assignment of a projection to g_n satisfies less than an ε-fraction of identities in Σ_n and we conclude that Σ_n is ε-robust.

It remains to show that Σ_n is satisfied in $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k})$. Define the following functions on $\mathbf{K}_k$:

$$g_n(x_1, \dots, x_{3^n}) = \begin{cases} a & \text{if } \{i \mid x_i = a\} \in S_n \text{ for some } a, \\ x_1 + k & \text{otherwise.} \end{cases}$$

Note that, by the choice of S_n, the condition in the first case of the definition of g_n cannot hold for more than one value a, so this function is well-defined. Let $f_n(x_0, y_1, \dots, y_\ell) = x_0$. It is straightforward to check that $f_n, g_n \in \mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_{2k})$ and that these functions satisfy all identities in Σ_n. □

We now show that, for any $k \geq 3$, there is a bipartite minor condition that is satisfied in $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$ for some c, but not in $\mathrm{Pol}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ for any $k < k' \leq c'$. This means that even if one proves NP-hardness of PCSP$(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ for some *fixed* $k' > 3$ and *all* $c' \geq k'$, Theorem 3.1 would not immediately imply the same for any fixed k with $3 \leq k < k'$.

Proposition 10.3. *For each $k \geq 3$, there exists $c \geq k$ such that* $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$ *has no minion homomorphism to* $\mathrm{Pol}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ *for any $k < k' \leq c'$.*

Proof. A simple bipartite minor condition that is not satisfied in $\mathrm{Pol}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ is the $\mathbf{K}_{k+1}$-loop condition—this follows from the fact that $\mathbf{K}_{k'}$ contains $\mathbf{K}_{k+1}$, but $\mathbf{K}_{c'}$ has no loop. The $\mathbf{K}_{k+1}$-loop condition is described by the following two identities:

$$t(x_1, \dots, x_{k+1}) \approx s(x_1, x_2, \dots, x_k, x_{k+1}),$$
$$t(x_1, \dots, x_{k+1}) \approx s(x_2, x_1, \dots, x_{k+1}, x_k).$$

The variables on the right-hand side correspond to the edges of $\mathbf{K}_{k+1}$ (viewed as a digraph); we will index them accordingly: The first one is $x_{1,2}$, the second $x_{2,1}$, and so on, so the arity of s is $k(k+1)$. We claim that this condition is satisfied in $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$ for some $c \geq k$. To show that, we define functions t and s without regard to the size of their range by picking a new colour whenever we are not forced by any of the identities to do otherwise. More precisely, we set

$$t(a_1, \dots, a_{k+1}) = (a_1, \dots, a_{k+1}).$$

In other words, if we view t as a colouring of $\mathbf{K}_k^{k+1}$, then each vertex gets its own colour. Further, let

$$s(a_{1,2}, \dots, a_{k+1,k}) = \begin{cases} t(a_1, \dots, a_{k+1}) & \text{if } a_{i,j} = a_i \text{ for all } i \neq j, \\ t(a_1, \dots, a_{k+1}) & \text{if } a_{i,j} = a_j \text{ for all } i \neq j, \text{ and} \\ (a_{1,2}, \dots, a_{k+1,k}) & \text{otherwise.} \end{cases}$$

If we view s as a colouring of $\mathbf{K}_k^{k(k+1)}$, then the last case in the definition of s assigns each vertex (satisfying the condition of the case) its own colour, and that colour is not used in t. Note that s is well-defined, because if both $a_{i,j} = a_i$ and $a_{i,j} = a'_j$ for all $i \neq j$, then $a_{i,j} = a_{i',j'}$ for all $i \neq j$ and $i' \neq j'$, and in that case both rows give the same value. It is straightforward to check that these functions satisfy the $\mathbf{K}_{k+1}$-loop condition, since the tuples that require checking are exactly those in the first two rows of the definition of s. We also claim that both t and s are proper colourings. This is obvious for t. For s, we have to show that if $s(a_{1,2}, \dots, a_{k+1,k}) = s(b_{1,2}, \dots, b_{k+1,k})$, then $a_{i,j} = b_{i,j}$ for some $i \neq j$. Clearly, if the two values of s agree, then the resulting colours must have been obtained using one or both of the first two rows of definition of s. If the same row was used for both, then there is nothing to prove, as the resulting colour is uniquely determined by the tuples $(a_1, \dots, a_{k+1})$ and $(b_1, \dots, b_{k+1})$, respectively. The remaining case is when both rows were used. Since the situation is symmetric, we may assume that $a_{i,j} = a_i$ and $b_{i,j} = b_j$ for all $i \neq j$. Further, since the result is $t(a_1, \dots, a_{k+1}) = t(b_1, \dots, b_{k+1})$, we have that $a_i = b_i$ for all i. Finally, since there are only k possible values for a_i, we have that $a_i = a_j$ for some $i \neq j$, which implies that $a_{i,j} = a_i = a_j = b_j = b_{i,j}$. The total number c of colours used in the range of t and s is bounded by $k^{k+1} + k^{k(k+1)}$. □

The above proposition does not give the best possible value of c. In fact, it was shown by Olšák [79] that for $k = 3$, a possible value is $c = 6$. We include a proof for completeness.

Proposition 10.4. $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_6)$ *does not map by a minion homomorphism to* $\mathrm{Pol}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ *for any $3 < k' \leq c'$.*

Proof. As in the case of the previous proposition, the distinguishing bipartite minor condition is the $\mathbf{K}_4$-loop condition. Again, it is clear to see that it is not satisfied in $\mathrm{Pol}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$, so it is enough to define two polymorphisms from $\mathbf{K}_3$ to $\mathbf{K}_6$ that satisfy this condition. We consider:

$$t(x_1, x_2, x_3, x_4) = \begin{cases} x_1 & \text{if } x_1 = x_2 = x_3, \text{ and} \\ x_4 + 3 & \text{otherwise;} \end{cases}$$

$$s(x_{1,2}, x_{2,1}, x_{1,3}, \dots, x_{4,3}) = \begin{cases} t(x'_1, \dots, x'_4) & \text{if } x_{i,j} = x'_i \text{ for all } i \neq j, \\ t(x'_1, \dots, x'_4) & \text{if } x_{i,j} = x'_j \text{ for all } i \neq j, \text{ and} \\ x_{1,2} & \text{otherwise.} \end{cases}$$

Note that s is well-defined, because if both $x_{i,j} = x_i$ and $x_{i,j} = a'_j$ for all $i \neq j$, then $x_{i,j} = x_{i',j'}$ for all $i \neq j$ and $i' \neq j'$, and in that case both rows give the same value. Clearly, s and t satisfy the required identities. It is straightforward to check that t is a polymorphism; let us then prove that s is. As before, it is enough to show that if

$$s(x_{1,2}, \dots, x_{3,4}) = s(y_{1,2}, \dots, y_{3,4})$$

for some $x_{i,j}$'s and $y_{i,j}$'s, then there are $i \neq j$ such that $x_{i,j} = y_{i,j}$. Let us assume that the above common value of s is a and consider two cases: First, $a \in K_3$. Such value can be obtained for x's if $x_{1,2} = a$, or $x_{i,j} = a$ for all $i \neq j$ and $i \in \{1, 2, 3\}$, or $x_{i,j} = a$ for all $i \neq j$ and $j \in \{1, 2, 3\}$. Either way, $x_{1,2} = a$. By the same argument, we get $y_{1,2} = x_{1,2} = a$. Second, $a \in K_6 \setminus K_3$. This implies that one of the two rows of the definition of s have been used for both values. If the first rows have been used for both, we have $x_{4,i} = y_{4,i} = a - 3$ for all $i \neq 4$; similarly, if the second row has been used for both, then $x_{i,4} = y_{i,4} = a - 3$ for all $i \neq 4$. Hence, the first row was used for one value and the second for the other. Using the symmetry, we may assume that the first row was used for the x's and the second row for the y's, that is

$$a = x_4 + 3 = t(x_1, x_2, x_3, x_4) = s(x_1, x_2, x_1, x_3, x_1, x_4, x_2, x_3, x_2, x_4, x_3, x_4),$$
$$a = y_4 + 3 = t(y_1, y_2, y_3, y_4) = s(y_2, y_1, y_3, y_1, y_4, y_1, y_3, y_2, y_4, y_2, y_4, y_3).$$

Since the second row of the definition of t was used, we know that y_1, y_2, y_3 are not all equal, so assume by symmetry that $y_2 \neq y_3$. Note that we now need to show that $x_i = y_j$ for some $i \neq j$. If it is not the case, then $y_4 = x_4 \neq y_2$ and also $y_4 = x_4 \neq y_3$, which means that y_2, y_3, and y_4 take three different values from K_3, and consequently x_1 must be equal to one of them, a contradiction. □

Note that, for any fixed $k \geq 3$, the case $c = 2k$ is the smallest one for which NP-hardness of $\mathrm{PCSP}(\mathbf{K}_k, \mathbf{K}_c)$ is still open, unless k is large enough for Huang's result [59] to apply. The smallest (in terms of k, c) open case is $\mathrm{PCSP}(\mathbf{K}_3, \mathbf{K}_6)$, and the above result shows that even if one proves NP-hardness of $\mathrm{PCSP}(\mathbf{K}_{k'}, \mathbf{K}_{c'})$ for all $4 \leq k' \leq c'$, Theorem 3.1 would not immediately imply NP-hardness of $\mathrm{PCSP}(\mathbf{K}_3, \mathbf{K}_6)$.

Finally, we prove that if we fix k and increase c, then the family of bipartite minor conditions satisfied in $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$ also grows.

Proposition 10.5. *For any $c \geq k > 2$, there exists $C > c$ such that $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_C)$ has no minion homomorphism to $\mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_c)$.*

Proof. To simplify the notation, let $\mathscr{K}_n = \mathrm{Pol}(\mathbf{K}_k, \mathbf{K}_n)$. We need to find a bipartite minor condition that is not satisfied in $\mathscr{K}_c$, but is satisfied in $\mathscr{K}_C$ for some $C > c$. A simple condition that is not satisfied in $\mathscr{K}_c$ is the $\mathbf{K}_k$-loop condition. However, this condition is also not satisfied in $\mathscr{K}_C$ for any C either. Nevertheless, Theorem 4.12(3)↔(1) implies that there is a condition of the form $\Sigma(\mathbf{K}_k, \mathbf{F})$ where $\mathbf{F}$ is some graph that witnesses that there is no minion homomorphism. We consider the condition $\Sigma(\mathbf{K}_k, \mathbf{K}_{c+1})$. Recall that $\Sigma(\mathbf{K}_k, \mathbf{K}_{c+1})$ is constructed as follows: Introduce a k-ary symbol f_u for each $u \in [c+1]$, and for each $\{u, v\} \subseteq [c+1]$, $u < v$ a $k(k-1)$-ary symbol $g_{\{u,v\}}$ and all identities of the following form:

$$f_u(x_1, \dots, x_k) \approx g_{\{u,v\}}(x_1, x_2, \dots, x_{k-1}, x_k),$$
$$f_v(x_1, \dots, x_k) \approx g_{\{u,v\}}(x_2, x_1, \dots, x_k, x_{k-1}),$$

where the variables on the right-hand side are distributed in such a way that x_i and x_j appear together in some column for all $i \neq j$. First, we claim that this condition is not satisfied in $\mathscr{K}_c$. This follows directly from Lemma 3.14(2): If it was, then there would be a homomorphism from $\mathbf{K}_{c+1}$ to $\mathbf{K}_c$.

We now show that this bipartite minor condition is satisfied in $\mathscr{K}_C$ for some C. As in the proof of Proposition 10.3, we define functions first and determine C later. Define functions f_u and $g_{\{u,v\}}$ without regard to the size of the range by setting $f_u(a, \dots, a) = a$, $g_{\{u,v\}}(a, \dots, a) = a$, and then picking a fresh colour whenever we are not forced by any of the identities to do otherwise. More precisely, we set

$$f_u(a_1, \dots, a_k) = \begin{cases} a & \text{if } a_1 = \dots = a_k = a \text{, and} \\ (u; a_1, \dots, a_k) & \text{otherwise,} \end{cases}$$

and similarly,

$$g_{u,v}(a_{1,2}, \dots, a_{k,k-1}) = \begin{cases} a & \text{if } a_{1,2} = \dots = a_{k,k-1} = a \text{, else} \\ (u; a_1, \dots, a_k) & \text{if } a_{i,j} = a_i \text{ for all } i \neq j, \\ (v; a_1, \dots, a_k) & \text{if } a_{i,j} = a_j \text{ for all } i \neq j \text{, and} \\ (u, v; a_{1,2}, \dots, a_{k,k-1}) & \text{otherwise.} \end{cases}$$

Note that $g_{u,v}$ is well defined, because if both $a_{i,j} = a_i$ and $a_{i,j} = a'_j$ for all $i \neq j$, then $a_{i,j} = a_{i',j'}$ for all $i \neq j$ and $i' \neq j'$, and in that case the first row is used. It is straightforward to check that these functions satisfy the minor condition. We also claim that both f_u and $g_{u,v}$ are proper colourings, but that is easy to see, since for a fixed u ($\{u, v\}$, respectively) no colour is used twice for a value of f_u ($g_{\{u,v\}}$, respectively). The total number C of colours used is bounded by $k + (c+1)k^k + (c+1)^2 k^{k(k-1)}$. □

We remark that $\Sigma(\mathbf{K}_3, \mathbf{K}_4)$ is not satisfied in $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_4)$, since it is a non-trivial condition (follows from Lemma 3.14(2)) and $\mathrm{Pol}(\mathbf{K}_3, \mathbf{K}_4)$ maps to $\mathscr{P}_2$ by a minion homomorphism (see Example 2.22). So in the previous proof, the obvious lower bound on C cannot be always met.

11 CONCLUSION

This article deals with the Promise CSP framework, which is a significant generalisation of the (finite-domain) CSP. The PCSP provides a nice interplay between the study of approximability and universal-algebraic methods in computational complexity. We presented a general abstract algebraic theory that captures the complexity of PCSPs with a fixed template $(\mathbf{A}, \mathbf{B})$. The key element in our approach is the bipartite minor conditions satisfied in the polymorphism minion $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ of a template. We have shown that such conditions determine the complexity $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. We gave some applications of our general theory, in particular, in approximate graph colouring.

The complexity landscape of PCSP (beyond CSP) is largely unknown, even in the Boolean case (despite some progress in References [30, 49]) and includes many specific problems of interest. We hope that our theory will provide the basis for a fruitful research programme of charting this landscape. Below, we discuss some of the possible directions within this programme.

Let us first discuss how the complexity classification quest for PCSPs compares with that for CSPs. As we said above, the gist of the algebraic approach is that lack or presence of (high-dimensional) symmetries determines the complexity. For CSPs, there is a sharp algebraic dichotomy: having only trivial symmetries (i.e., satisfying only those systems of minor identities that are satisfied in polymorphisms of every CSP) leads to NP-hardness, while any non-trivial symmetry implies rather strong symmetry and thus leads to tractability. Moreover, the algorithms for tractable cases are (rather involved) combinations of only two basic algorithms—one is based on local propagation [13] and the other can be seen as a very general form of Gaussian elimination [60]. It is already clear that the situation is more complicated for PCSPs: There are hard PCSPs with non-trivial (but limited in some sense) symmetries, and tractable cases are more varied [7, 29, 30, 47]. This calls for more advanced methods, and we hope that our article will provide the basis for such methods. There is an obvious question whether PCSPs exhibit a dichotomy as CSPs do, but there is not enough evidence yet to conjecture an answer. More specifically, it is not clear whether there is any PCSP whose polymorphisms are not limited enough (in terms of satisfying systems of minor identities) to give NP-hardness, but also not strong enough to ensure tractability. Classifications for special cases such as Boolean PCSPs and graph homomorphisms would help to obtain more intuition about the general complexity landscape of PCSPs, but these special cases are currently open.

The sources of hardness in PCSP appear to be much more varied than in CSP (that has a unique such source), and much remains to be understood there. What limitations on the bipartite minor conditions satisfied in polymorphisms lead to NP-hardness? We gave some general and some specific results in this direction, but it is clear that our general theory needs to be further developed. Currently, variants of the Gap Label Cover provide the source of hardness, but it is possible that new versions of GLC may need to be used. It is not clear in advance what these versions would be—their form would be dictated by the analysis of polymorphisms and minor conditions. However, it would be interesting to eventually go even further and avoid dependency on deep approximation results such as the PCP theorem and parallel repetition, which are more appropriate for quantitative approximation concerning the number of (un)satisfied constraints. Instead, one would aim to provide a self-contained theory that includes a new type of reduction between PCSPs so one could bypass the PCP Theorem and parallel repetition altogether. In particular, can one come up with purely algebraic reductions (e.g., by extending pp-constructions) that create and amplify the algebraic gap in problems $\mathrm{PMC}_{\mathscr{M}}(N)$, which is what the PCP theorem and parallel repetition do for the quantitative gap in Gap Label Cover?

The analysis of polymorphisms of approximate graph colouring problems (and their relatives) may provide further intuition as to what limitations on minor conditions can be used for hardness proofs. To give another specific problem, it may be interesting to understand for what structures $\mathbf{B}$ the problem $\mathrm{PCSP}(\mathbf{T}, \mathbf{B})$ is hard. Here $\mathbf{T}$ is the Boolean "1-in-3" structure from Example 2.8 and $\mathbf{B}$ is a (not necessarily Boolean) structure with a single ternary relation. We know that the problem is NP-hard for $\mathbf{B} = \mathbf{T}$ and tractable for $\mathbf{B} = \mathbf{H}_2$.

What algorithmic techniques are needed to solve tractable PCSPs? One general approach to proving tractability of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is presented in Reference [30]. The main idea is to find a structure $\mathbf{D}$ such that $\mathbf{A} \to \mathbf{D} \to \mathbf{B}$ and $\mathrm{CSP}(\mathbf{D})$ is tractable. Such structures are called "homomorphic sandwiches" in Reference [30]. It is clear that an algorithm for $\mathrm{CSP}(\mathbf{D})$ solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. In general, $\mathbf{D}$ may have infinite domain (but instances of $\mathrm{CSP}(\mathbf{D})$ are still finite). Indeed, for $\mathrm{PCSP}(\mathbf{T}, \mathbf{H}_2)$ from Example 2.8 no such finite $\mathbf{D}$ exists (Theorem 8.1), but there are several infinite ones [10, 29, 30], such as $\mathbf{D} = (\mathbb{Z}; x + y + z = 1)$. It would be interesting to find out what other tractable PCSPs (e.g., those considered in References [29, 49]) require infinite-domain CSPs. Another natural question is "how infinite" such a template $\mathbf{D}$ needs to be—while the class of all infinite-domain CSPs essentially covers all decision problems [21], some parts of the theory can be generalised from finite structures to the rich class of so called ω-categorical structures [20, 82]. Can a structure $\mathbf{D}$ such that $\mathbf{A} \to \mathbf{D} \to \mathbf{B}$ be ω-categorical, say, for $(\mathbf{A}, \mathbf{B}) = (\mathbf{T}, \mathbf{H}_2)$? It would be also very interesting to develop a general theory of how such a structure $\mathbf{D}$ can be constructed from bipartite minor conditions satisfied in Pol($\mathbf{A}, \mathbf{B}$) and what properties of such conditions guarantee tractability of CSP($\mathbf{D}$). Overall, it is another interesting feature of the (finite-domain) PCSP framework that it seems to require research into infinite-domain CSP.

The polynomial-time algorithms of Bulatov and Zhuk [36, 92, 93] that solve all tractable finite-domain CSPs are quite involved and both use deep structural analysis of finite algebras. Can algorithms for tractable PCSPs that involve infinite-domain CSPs (e.g., such as those in References [30, 31]) lead to a new simpler algorithm that solves all tractable CSPs?

Every tractable finite-domain CSP can be solved [36, 92, 93] by a (rather involved) combination of local consistency checking [13] and the "few subpowers" algorithm that handles compact representation of constraints [60]. The study of local consistency checking algorithms generally played a very important role in the algebraic theory of CSP. Apart from being one of the two main algorithmic approaches, it connected, via homomorphism dualities [37], the algebraic theory of CSP with the combinatorial [58] and logical [66] approaches. As discussed in Section 7, it makes sense to use local consistency checking for PCSP($\mathbf{A}, \mathbf{B}$). Although it is obviously an interesting problem to characterise PCSPs solvable by local consistency, it is not clear yet whether this method still plays a key role for PCSP or must be superseded by more powerful algorithms in this context. The study of duality for CSP is intimately connected with investigating CSPs inside P, e.g., those in NL, L, or AC^0—it is natural to extend this line of research to PCSP. There are many other (both open and resolved) questions about local consistency for CSP that can be investigated for PCSP.

Unlike with the local consistency method, it is not even clear how to transfer the "few subpowers" algorithm to the realm of PCSP. It may be that a different algorithmic approach is required that will handle all that "few subpowers" could and that can be used for PCSP.

There are many results in CSP exactly characterising the power of a given algorithm. We generalised some of such results to PCSP in Section 7, but this line of research will need to continue, especially as new algorithmic techniques are being developed for PCSP.

Finally, the PCSP framework itself can be extended by incorporating approximation based on counting (un)satisfied constraints—see Reference [8] for an example—this would give another range of interesting open questions, but now we will not discuss this direction further.

ACKNOWLEDGMENTS

We would like to thank Mirek Olšák, Alex Kazda, Josh Brakensiek, Venkat Guruswami, Manuel Bodirsky, Erkko Lehtonen, and Marcello Mamino for valuable discussions. We would also like to thank Standa Živný, Marcin Wrochna, and the referees of both the conference and journal versions for useful comments on the manuscript.

REFERENCES

[1] Erhard Aichinger and Peter Mayr. 2016. Finitely generated equational classes. *J. Pure Appl. Algeb.* 220, 8 (2016), 2816–2827. DOI: https://doi.org/10.1016/j.jpaa.2016.01.001.

[2] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. 1997. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.* 54, 2 (Apr. 1997), 317–331. DOI: https://doi.org/10.1006/jcss.1997.1472.

[3] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. 1998. Proof verification and the hardness of approximation problems. *J. ACM* 45, 3 (1998), 501–555.

[4] Sanjeev Arora and Shmuel Safra. 1998. Probabilistic checking of proofs: A new characterization of NP. *J. ACM* 45, 1 (Jan. 1998), 70–122. DOI: https://doi.org/10.1145/273865.273901.

[5] Per Austrin, Amey Bhangale, and Aditya Potukuchi. 2019. Simplified inapproximability of hypergraph coloring via t-agreeing families. (Apr. 2019). arXiv:1904.01163.

[6] Per Austrin, Amey Bhangale, and Aditya Potukuchi. 2020. Improved inapproximability of rainbow coloring. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*. 1479–1495. DOI: https://doi.org/10.1137/1.9781611975994.90.

[7] Per Austrin, Venkatesan Guruswami, and Johan Håstad. 2017. $(2 + \varepsilon)$-sat is NP-hard. *SIAM J. Comput.* 46, 5 (2017), 1554–1573. DOI: https://doi.org/10.1137/15M1006507.

[8] Per Austrin and Johan Håstad. 2013. On the usefulness of predicates. *ACM Trans. Comput. Theor.* 5, 1 (2013), 1–24.

[9] Libor Barto. 2013. Finitely related algebras in congruence distributive varieties have near unanimity terms. *Canad. J. Math.* 65, 1 (2013), 3–21.

[10] Libor Barto. 2019. Promises make finite (constraint satisfaction) problems infinitary. In *Proceedings of the 34th ACM/IEEE Symposium on Logic in Computer Science (LICS'19)*. IEEE, 1–8.

[11] Libor Barto, Michael Kompatscher, Miroslav Olšák, Trung Van Pham, and Michael Pinsker. 2017. The equivalence of two dichotomy conjectures for infinite domain constraint satisfaction problems. In *Proceedings of the 32nd ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. 1–12. DOI: https://doi.org/10.1109/LICS.2017.8005128.

[12] Libor Barto and Marcin Kozik. 2012. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Log. Meth. Comput. Sci.* 8, 1:07 (2012), 1–26. DOI: https://doi.org/10.2168/LMCS-8(1:7)2012.

[13] Libor Barto and Marcin Kozik. 2014. Constraint satisfaction problems solvable by local consistency methods. *J. ACM* 61, 1 (Jan. 2014), 3:1–3:19. DOI: https://doi.org/10.1145/2556646.

[14] Libor Barto and Marcin Kozik. 2016. Robustly solvable constraint satisfaction problems. *SIAM J. Comput.* 45, 4 (2016), 1646–1669. DOI: https://doi.org/10.1137/130915479.

[15] Libor Barto, Andrei Krokhin, and Ross Willard. 2017. Polymorphisms, and how to use them. In *The Constraint Satisfaction Problem: Complexity and Approximability*, Andrei Krokhin and Stanislav Živný (Eds.). Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 1–44. DOI: https://doi.org/10.4230/DFU.Vol7.15301.1.

[16] Libor Barto, Jakub Opršal, and Michael Pinsker. 2018. The wonderland of reflections. *Israel J. Math.* 223, 1 (Feb. 2018), 363–398. DOI: https://doi.org/10.1007/s11856-017-1621-9.

[17] Libor Barto and Michael Pinsker. 2020. Topology is irrelevant (in a dichotomy conjecture for infinite domain constraint satisfaction problems). *SIAM J. Comput.* 49, 2 (2020), 365–393. DOI: https://doi.org/10.1137/18M1216213.

[18] Mihir Bellare, Oded Goldreich, and Madhu Sudan. 1998. Free bits, PCPs, and nonapproximability—Towards tight results. *SIAM J. Comput.* 27, 3 (1998), 804–915. DOI: https://doi.org/10.1137/S0097539796302531.

[19] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. 1988. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th ACM Symposium on Theory of Computing (STOC'88)*. ACM, New York, NY, 113–131. DOI: https://doi.org/10.1145/62212.62223.

[20] Manuel Bodirsky. 2008. Constraint satisfaction problems with infinite templates. In *Complexity of Constraints* (2009-05-05) (Lecture Notes in Computer Science), Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer (Eds.), Vol. 5250. Springer, 196–228. DOI: https://doi.org/10.1007/978-3-540-92800-3_8.

[21] Manuel Bodirsky and Martin Grohe. 2008. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP'08)*. 184–196.

[22] Manuel Bodirsky and Marcello Mamino. 2017. Constraint satisfaction problems over numeric domains. In *The Constraint Satisfaction Problem: Complexity and Approximability*, Andrei Krokhin and Stanislav Živný (Eds.). Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 79–111. DOI: https://doi.org/10.4230/DFU.Vol7.15301.79.

[23] Manuel Bodirsky, Antoine Mottet, Miroslav Olšák, Jakub Opršal, Michael Pinsker, and Ross Willard. 2019. Topology is relevant (in a dichotomy conjecture for infinite-domain constraint satisfaction problems). In *Proceedings of the 34th ACM/IEEE Symposium on Logic in Computer Science (LICS'19)*. IEEE, 1–12.

[24] V. G. Bodnarchuk, L. A. Kaluzhnin, Viktor N. Kotov, and Boris A. Romov. 1969. Galois theory for Post algebras. I. *Cybernetics* 5, 3 (1969), 243–252.

[25] V. G. Bodnarchuk, L. A. Kaluzhnin, Viktor N. Kotov, and Boris A. Romov. 1969. Galois theory for Post algebras. II. *Cybernetics* 5, 5 (1969), 531–539.

[26] Joshua Brakensiek and Venkatesan Guruswami. 2016. New hardness results for graph and hypergraph colorings. In *Proceedings of the 31st Conference on Computational Complexity (CCC'16) (Leibniz International Proceedings in Informatics (LIPIcs))*, Ran Raz (Ed.), Vol. 50. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 14:1–14:27. DOI: https://doi.org/10.4230/LIPIcs.CCC.2016.14.

[27] Joshua Brakensiek and Venkatesan Guruswami. 2016. Promise constraint satisfaction: Algebraic structure and a symmetric Boolean dichotomy. *ECCC* Report No. 183 (2016). Retrieved from https://eccc.weizmann.ac.il/report/2016/183/.

[28] Joshua Brakensiek and Venkatesan Guruswami. 2017. The quest for strong inapproximability results with perfect completeness. In *Proceedings of the Approximation, Randomization, and Combinatorial Optimization Conference: Algorithms and Techniques (APPROX/RANDOM'17)*. 4:1–4:20. DOI: https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.4.

[29] Joshua Brakensiek and Venkatesan Guruswami. 2018. Promise constraint satisfaction: Structure theory and a symmetric Boolean dichotomy. In *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms (SODA'18)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1782–1801. DOI: https://doi.org/10.1137/1.9781611975031.117.

[30] Joshua Brakensiek and Venkatesan Guruswami. 2019. An algorithmic blend of LPs and ring equations for promise CSPs. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*. 436–455. DOI: https://doi.org/10.1137/1.9781611975482.28.

[31] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. 2020. The power of the combined basic LP and affine relaxation for promise CSPs. *SIAM J. Comput.* 49, 6 (2020), 1232–1248. DOI: https://doi.org/10.1137/20M1312745 arXiv:1907.04383v3.

[32] Jonah Brown-Cohen and Prasad Raghavendra. 2015. Combinatorial Optimization Algorithms via Polymorphisms. (Jan. 2015). arXiv:1501.01598.

[33] Andrei Bulatov and Peter Jeavons. 2001. *Algebraic Structures in Combinatorial Problems.* Technical Report. Technische Universität Dresden.

[34] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.* 34, 3 (Mar. 2005), 720–742. DOI: https://doi.org/10.1137/S0097539700376676.

[35] Andrei A. Bulatov. 2013. The complexity of the counting constraint satisfaction problem. *J. ACM* 60, 5 (Oct. 2013).

[36] Andrei A. Bulatov. 2017. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the IEEE 58th Symposium on Foundations of Computer Science (FOCS'17)*. 319–330. DOI: https://doi.org/10.1109/FOCS.2017.37.

[37] Andrei A. Bulatov, Andrei Krokhin, and Benoit Larose. 2008. Dualities for constraint satisfaction problems. In *Complexity of Constraints: An Overview of Current Research Themes.* Springer Berlin, 93–124. DOI: https://doi.org/10.1007/978-3-540-92800-3_5.

[38] Jakub Bulín, Andrei Krokhin, and Jakub Opršal. 2019. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st ACM SIGACT Symposium on the Theory of Computing (STOC'19).* ACM, New York, NY, 602–613. DOI: https://doi.org/10.1145/3313276.3316300.

[39] Hubie Chen and Benoit Larose. 2017. Asking the metaquestions in constraint tractability. *ACM Trans. Comput. Theor.* 9, 3 (Oct. 2017), 11:1–11:27. DOI: https://doi.org/10.1145/3134757.

[40] Hubie Chen, Matthew Valeriote, and Yuichi Yoshida. 2016. Testing assignments to constraint satisfaction problems. In *Proceedings of the IEEE 57th Symposium on Foundations of Computer Science (FOCS'16)*. 525–534. DOI: https://doi.org/10.1109/FOCS.2016.63.

[41] Víctor Dalmau, Marcin Kozik, Andrei A. Krokhin, Konstantin Makarychev, Yury Makarychev, and Jakub Opršal. 2017. Robust algorithms with polynomial loss for near-unanimity CSPs. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*. 340–357. DOI: https://doi.org/10.1137/1.9781611974782.22.

[42] Victor Dalmau, Andrei Krokhin, and Rajsekar Manokaran. 2018. Towards a characterization of constant-factor approximable finite-valued CSPs. *J. Comput. System Sci.* 97 (2018), 14–27.

[43] Víctor Dalmau and Justin Pearson. 1999. Closure functions and width 1 problems. In *Principles and Practice of Constraint Programming – CP'99*, Joxan Jaffar (Ed.). Springer Berlin, 159–173.

[44] Irit Dinur. 2007. The PCP theorem by gap amplification. *J. ACM* 54, 3 (June 2007). DOI: https://doi.org/10.1145/1236457.1236459.

[45] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. 2005. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Comput.* 34, 5 (May 2005), 1129–1146. DOI: https://doi.org/10.1137/S0097539704443057.

[46] Irit Dinur, Elchanan Mossel, and Oded Regev. 2009. Conditional hardness for approximate coloring. *SIAM J. Comput.* 39, 3 (2009), 843–873. DOI: https://doi.org/10.1137/07068062X.

[47] Irit Dinur, Oded Regev, and Clifford Smyth. 2005. The hardness of 3-uniform hypergraph coloring. *Combinatorica* 25, 5 (Sept. 2005), 519–535. DOI: https://doi.org/10.1007/s00493-005-0032-4.

[48] Tomás Feder and Moshe Y. Vardi. 1998. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.* 28, 1 (Feb. 1998), 57–104. DOI: https://doi.org/10.1137/S0097539794266766.

[49] Miron Ficak, Marcin Kozik, Miroslav Olšák, and Szymon Stankiewicz. 2019. Dichotomy for symmetric Boolean PCSPs. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP'19) (Leibniz International Proceedings in Informatics (LIPIcs))*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.), Vol. 132. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 57:1–57:12. DOI: https://doi.org/10.4230/LIPIcs.ICALP.2019.57 arXiv:1904.12424.

[50] M. R. Garey and David S. Johnson. 1976. The complexity of near-optimal graph coloring. *J. ACM* 23, 1 (1976), 43–49. DOI: https://doi.org/10.1145/321921.321926.

[51] David Geiger. 1968. Closed systems of functions and predicates. *Pacific J. Math.* 27 (1968), 95–100.

[52] Venkatesan Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. 2017. Superpolylogarithmic hypergraph coloring hardness via low-degree long codes. *SIAM J. Comput.* 46, 1 (2017), 132–159. DOI: https://doi.org/10.1137/140995520.

[53] Venkatesan Guruswami and Sanjeev Khanna. 2004. On the hardness of 4-coloring a 3-colorable graph. *SIAM J. Disc. Math.* 18, 1 (2004), 30–40. DOI: https://doi.org/10.1137/S0895480100376794.

[54] Venkatesan Guruswami and Euiwoong Lee. 2017. Strong inapproximability results on balanced rainbow-colorable hypergraphs. *Combinatorica* (14 Dec. 2017). DOI: https://doi.org/10.1007/s00493-016-3383-0.

[55] Venkatesan Guruswami and Sai Sandeep. 2020. d-to-1 hardness of coloring 3-colorable graphs with $O(1)$ colors. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP'20) (Leibniz International Proceedings in Informatics (LIPIcs))*, Artur Czumaj, Anuj Dawar, and Emanuela Merelli (Eds.), Vol. 168. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 62:1–62:12. DOI: https://doi.org/10.4230/LIPIcs.ICALP.2020.62.

[56] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (July 2001), 798–859. DOI: https://doi.org/10.1145/502090.502098.

[57] Pavol Hell and Jaroslav Nešetřil. 1990. On the complexity of H-coloring. *J. Combin. Theor. Ser. B* 48, 1 (1990), 92–110. DOI: 10.1016/0095-8956(90)90132-J

[58] Pavol Hell and Jaroslav Nešetřil. 2004. *Graphs and Homomorphisms*. Oxford University Press.

[59] Sangxia Huang. 2013. Improved hardness of approximating chromatic number. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim (Eds.). Springer, Berlin, 233–243. DOI: https://doi.org/10.1007/978-3-642-40328-_17.

[60] Paweł M. Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2010. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.* 39, 7 (2010), 3023–3037. DOI: https://doi.org/10.1137/090775646.

[61] Peter Jeavons. 1998. On the algebraic structure of combinatorial problems. *Theor. Comput. Sci.* 200, 1–2 (1998), 185–204.

[62] Peter Jeavons, David Cohen, and Marc Gyssens. 1997. Closure properties of constraints. *J. ACM* 44, 4 (July 1997), 527–548. DOI: https://doi.org/10.1145/263867.263489.

[63] Ken-ichi Kawarabayashi and Mikkel Thorup. 2017. Coloring 3-colorable graphs with less than $n^{1/5}$ colors. *J. ACM* 64, 1 (2017), 4:1–4:23. DOI: https://doi.org/10.1145/3001582.

[64] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. 2000. On the hardness of approximating the chromatic number. *Combinatorica* 20, 3 (01 Mar. 2000), 393–415. DOI: https://doi.org/10.1007/s004930070013.

[65] Subhash Khot. 2001. Improved inapproximability results for MaxClique, chromatic number and approximate graph coloring. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 600–609.

[66] Phokion G. Kolaitis and Moshe Y. Vardi. 2008. A logical approach to constraint satisfaction. In *Complexity of Constraints: An Overview of Current Research Themes*, Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer (Eds.). LNCS, Vol. 5250. Springer-Verlag, 125–155.

[67] Vladimir Kolmogorov, Andrei Krokhin, and Michal Rolínek. 2017. The complexity of general-valued CSPs. *SIAM J. Comput.* 46, 3 (2017), 1087–1110.

[68] Andrei Krokhin and Jakub Opršal. 2019. The complexity of 3-colouring H-colourable graphs. In *Proceedings of the IEEE 60th Symposium on Foundations of Computer Science (FOCS'19)*. 1227–1239. DOI: https://doi.org/10.1109/FOCS.2019.00076 arxiv:1904.03214.

[69] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. 2020. Topology and adjunction in promise constraint satisfaction. (Mar. 2020). arXiv:2003.11351.

[70] Andrei Krokhin and Stanislav Živný (Eds.). 2017. *The Constraint Satisfaction Problem: Complexity and Approximability*. Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[71] Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. 2012. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS'12)*. ACM, New York, NY, 484–495. DOI: https://doi.org/10.1145/2090236.2090274.

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

[73] Benoit Larose. 2017. Algebra and the complexity of digraph CSPs: A survey. In *The Constraint Satisfaction Problem: Complexity and Approximability*, Andrei Krokhin and Stanislav Živný (Eds.). Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 267–285. DOI: https://doi.org/10.4230/DFU.Vol7.15301.267.

[74] Benoit Larose and László Zádori. 2007. Bounded width problems and algebras. *Algeb. Univers.* 56, 3 (01 June 2007), 439–466. DOI: https://doi.org/10.1007/s00012-007-2012-6.

[75] Lászlo Lovász. 1978. Kneser's conjecture, chromatic number, and homotopy. *J. Combin. Theor. Ser. A* 25, 3 (1978), 319–324. DOI: https://doi.org/10.1016/0097-3165(78)90022-5.

[76] Colin McDiarmid. 1993. A random recolouring method for graphs and hypergrams. *Combinat. Proba. Comput.* 2 (1993), 363–365. DOI: https://doi.org/10.1017/S0963548300000730.

[77] Walter D. Neumann. 1974. On Malcev conditions. *J. Austral. Math. Soc.* 17, 3 (5 1974), 376–384. DOI: https://doi.org/10.1017/S1446788700017122.

[78] Miroslav Olšák. 2017. The weakest nontrivial idempotent equations. *Bull. London Math. Soc.* 49, 6 (2017), 1028–1047. DOI: https://doi.org/10.1112/blms.12097.

[79] Miroslav Olšák. 2018. *Personal communication.*

[80] Miroslav Olšák. 2019. Loop conditions. *Algeb. Univers.* 81, 1 (Nov. 2019), 2:1–2:11. DOI: https://doi.org/10.1007/s00012-019-0631-3 arXiv:1701.00260.

[81] Jakub Opršal. 2017. Taylor's modularity conjecture and related problems for idempotent varieties. *Order* (Nov. 2017). DOI: https://doi.org/10.1007/s11083-017-9441-4.

[82] Michael Pinsker. 2015. Algebraic and model theoretic methods in constraint satisfaction. (2015). arXiv:1507.00931.

[83] Nicholas Pippenger. 2002. Galois theory for minors of finite functions. *Disc. Math.* 254, 1 (2002), 405–419. DOI: https://doi.org/10.1016/S0012-365X(01)00297-7.

[84] Ran Raz. 1998. A parallel repetition theorem. *SIAM J. Comput.* 27, 3 (1998), 763–803. DOI: https://doi.org/10.1137/S0097539795280895.

[85] Omer Reingold. 2008. Undirected connectivity in log-space. *J. ACM* 55, 4 (Sept. 2008), 17:1–17:24. DOI: https://doi.org/10.1145/1391289.1391291.

[86] Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC'78).* ACM, New York, NY, 216–226. DOI: https://doi.org/10.1145/800133.804350.

[87] Alexander Schrijver. 1978. Vertex-critical subgraphs of Kneser graphs. *Nieuw Arch. Wisk. (3)* 26, 3 (1978), 454–461.

[88] Mark H. Siggers. 2010. A strong Mal'cev condition for locally finite varieties omitting the unary type. *Algeb. Univers.* 64, 1 (Oct. 2010), 15–20. DOI: https://doi.org/10.1007/s00012-010-0082-3.

[89] Walter Taylor. 1973. Characterizing Mal'cev conditions. *Algeb. Univers.* 3 (1973), 351–397.

[90] Johan Thapper and Stanislav Živný. 2016. The complexity of finite-valued CSPs. *J. ACM* 63, 4 (2016), 37:1–37:33. DOI: https://doi.org/10.1145/2974019.

[91] Marcin Wrochna and Stanislav Živný. 2020. Improved hardness for H-colourings of G-colourable graphs. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA'20).* 1426–1435. DOI: https://doi.org/10.1137/1.9781611975994.86 arXiv:1907.00872.

[92] Dmitriy Zhuk. 2017. A proof of CSP dichotomy conjecture. In *Proceedings of the IEEE 58th Symposium on Foundations of Computer Science (FOCS'17).* 331–342. DOI: https://doi.org/10.1109/FOCS.2017.38.

[93] Dmitriy Zhuk. 2020. A proof of the CSP dichotomy conjecture. *J. ACM* 67, 5 (Aug. 2020), 30:1–30:78. DOI: https://doi.org/10.1145/3402029.

Received July 2019; revised November 2020; accepted March 2021