1-IN-3 vs. NOT-ALL-EQUAL: Dichotomy of a broken promise*

Lorenzo Ciardo
Department of Computer Science
University of Oxford
Oxford, UK
lorenzo.ciardo@cs.ox.ac.uk

Marcin Kozik
Theoretical Computer Science
Jagiellonian University
Kraków, Poland
marcin.kozik@uj.edu.pl

Andrei Krokhin
Department of Computer Science
Durham University
Durham, UK
andrei.krokhin@durham.ac.uk

Tamio-Vesa Nakajima
Department of Computer Science
University of Oxford
Oxford, UK
tamio-vesa.nakajima@cs.ox.ac.uk

Stanislav Živný
Department of Computer Science
University of Oxford
Oxford, UK
standa.zivny@cs.ox.ac.uk

ABSTRACT

The 1-IN-3 and NOT-ALL-EQUAL satisfiability problems for Boolean CNF formulas are two well-known NP-hard problems. In contrast, the *promise* 1-IN-3 vs. NOT-ALL-EQUAL problem can be solved in polynomial time. In the present work, we investigate this constraint satisfaction problem in a regime where the promise is weakened from either side by a *rainbow-free* structure, and establish a complexity dichotomy for the resulting class of computational problems.

CCS CONCEPTS

• **Theory of computation** → **Design and analysis of algorithms**; **Problems, reductions and completeness**; **Constraint and logic programming**.

KEYWORDS

constraint satisfaction, promise constraint satisfaction, polymorphisms, minions, algebraic approach

ACM Reference Format:
Lorenzo Ciardo, Marcin Kozik, Andrei Krokhin, Tamio-Vesa Nakajima, and Stanislav Živný. 2024. 1-IN-3 vs. NOT-ALL-EQUAL: Dichotomy of a broken promise. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3661814.3662069

*This work was supported by EPSRC Fellowship EP/X033201/1, UKRI EP/X024431/1, and a Clarendon Fund Scholarship. This research was funded in whole or in part by National Science Centre, Poland 2021/03/Y/ST6/00171. For the purpose of Open Access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising. All data is provided in full in the results section of this paper. The full version of the paper is available at arXiv:2210.03343.

LICS '24, July 8–11, 2024, Tallinn, Estonia

ACM ISBN 979-8-4007-0660-8/24/07
https://doi.org/10.1145/3661814.3662069

1 INTRODUCTION

Let φ be a Boolean formula given as a conjunction of clauses, each consisting of three (un-negated) variables. Consider the following question:

Is there a truth assignment such that each clause has exactly one true variable?

This is a well-known NP-hard computational problem, called (monotone) 1-IN-3 SAT [30]. Similarly, the question

Is there a truth assignment such that each clause has at least one true and one false variable?

is the NP-hard problem known in the literature as 3-NOT-ALL-EQUAL (NAE) SAT [30, 47]. In these and other variants, the Boolean satisfiability problem has had a central importance in the development of complexity theory, its investigation dating back at least to [24]. Notice that the first notion of satisfiability is stronger than the second: Any 1-IN-3 assignment is also an NAE assignment. Consider now the *promise* satisfiability problem that asks to distinguish whether a formula φ is satisfiable in the strong sense (a 1-IN-3 assignment exists) or φ is not even satisfiable in the weak sense (an NAE assignment does not exist). This problem — known as "1-IN-3 vs. NAE" [12] — is a relaxation of both problems considered above, in that it admits any answer on those formulas that are satisfiable in the weak but not in the strong sense. Equivalently, one is promised that the input formula is not of that kind. Let us try to solve this problem, following an algorithm from [12]. For any clause in φ involving three variables x, y, z, consider the linear equation

$$x + y + z = 1. \tag{1}$$

This results in a linear system, which may be solved over the integers in polynomial time by using, essentially, Gaussian elimination.[1] If there is no integer solution, we are sure that, in particular, no $\{0, 1\}$ solution exists: φ does not admit a 1-IN-3 assignment. If there is an integer solution, we round it by turning positive values into 1 and non-positive values into 0. Since no three positive (respectively, non-positive) integers can sum up to 1, we are guaranteed that the output of this process is a valid NAE assignment — while it is not

[1] More precisely, by the integral version of Gaussian elimination that corresponds, in matrix terms, to computing the Hermite or Smith normal forms of the matrix of the linear system [40].

necessarily a 1-IN-3 assignment, as is witnessed, for example, by the solution to (1) given by $x = y = 2$, $z = -3$.

In other words, while 1-IN-3 SAT is NP-hard, if we are promised that all of the unsatisfiable formulas we are considering are not even satisfiable in the weaker NAE sense, the problem becomes tractable (solvable in polynomial time). Similarly (and dually), the promise that all NAE-satisfiable formulas are also 1-IN-3-satisfiable turns NAE into a tractable problem. It is then natural to investigate what happens if we modify the promise. Clearly, a stronger promise would lead to an even easier problem and, in particular, to a tractable one. What if we weaken it? How does the promise impact on the complexity behaviour of the problem? Where is the boundary of tractability?

In order to formulate these questions in a formal way, it shall be convenient to use the paradigm of *Constraint Satisfaction Problems* (CSPs), which provides a broader context for capturing Boolean satisfiability problems, as well as other computational problems such as graph and hypergraph colouring. We can phrase a CSP as a homomorphism problem, where the objective is to test for the existence of a homomorphism between an *instance* structure $\mathbf{X}$ and a *template* structure $\mathbf{A}$. In the setting of satisfiability of Boolean formulas, we should think of $\mathbf{X}$ as a proxy for the formula φ, while $\mathbf{A}$ encodes the satisfiability notion we are considering. In this formulation, $\mathbf{X}$ and $\mathbf{A}$ are two similar (finite) *relational structures*, consisting of finite *domains* (X and A, respectively), as well as *relations* ($R^{\mathbf{X}} \subseteq X^r$ and $R^{\mathbf{A}} \subseteq A^r$, respectively) for each *relation symbol* R, where the positive integer r is the *arity* of R. A homomorphism between $\mathbf{X}$ and $\mathbf{A}$ is a map $f : X \to A$ that preserves the relations; i.e., $f(\mathbf{x}) \in R^{\mathbf{A}}$ whenever $\mathbf{x} \in R^{\mathbf{X}}$, where f is applied entrywise. We denote the existence of a homomorphism by $\mathbf{X} \to \mathbf{A}$. The CSP parameterised by $\mathbf{A}$, denoted by $\mathrm{CSP}(\mathbf{A})$, is the computational problem: "Given an instance $\mathbf{X}$, output YES if $\mathbf{X} \to \mathbf{A}$ and NO if $\mathbf{X} \not\to \mathbf{A}$". If we define the Boolean structure **1-in-3** whose unique relation, of arity 3, is the set

$$\{(0,0,1),(0,1,0),(1,0,0)\},$$

then $\mathrm{CSP}(\mathbf{1\text{-}in\text{-}3})$ is precisely the 1-IN-3 SAT problem. Similarly, we can formulate the NAE SAT problem as the CSP parameterised by the Boolean structure **NAE** whose unique relation, of arity 3, is the set

$$\{(0,0,1),(0,1,0),(1,0,0),(0,1,1),(1,0,1),(1,1,0)\}.$$

Other classic examples of CSPs are homomorphisms problems for digraphs (which are relational structures having a single, binary relation) and, more generally, hypergraphs. In particular, the CSP parameterised by the n-clique $\mathbf{K}_n$ is the well-known graph n-colouring problem.

Several decades of research efforts have equipped the framework of CSPs with a rather sharp set of tools — mostly coming from universal algebra — that can be leveraged to explain the computational complexity of satisfiability problems. More precisely, the complexity of $\mathrm{CSP}(\mathbf{A})$ is entirely determined by a certain type of identities holding in the *polymorphism clone* of $\mathbf{A}$, which contains all homomorphisms of the form $\mathbf{A}^k \to \mathbf{A}$ (where $\mathbf{A}^k$ is the k-fold direct power of $\mathbf{A}$) [7, 8, 20, 38, 39, 44].

This categorical phenomenon, known as a *Galois connection*, has been invaluable in the exploration of the complexity landscape of CSPs. Eventually, it has led to the positive resolution of Feder-Vardi's Dichotomy Conjecture (now Theorem) by Zhuk [50] and Bulatov [19], which asserts that a CSP is tractable in polynomial time if it has a polymorphism satisfying an identity of a certain kind, and it is NP-hard otherwise [27].

Promise problems like 1-IN-3 vs. NAE are captured by a paradigm, known as *Promise* CSP*s* (PCSPs for short), that generalises CSPs. Here, the template is a *pair* $(\mathbf{A}, \mathbf{B})$ of (finite) structures, and the computational problem $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is "Given an instance $\mathbf{X}$, output YES if $\mathbf{X} \to \mathbf{A}$ and NO if $\mathbf{X} \not\to \mathbf{B}$". In order for the YES and the NO instances to be disjoint, we require that $\mathbf{A} \to \mathbf{B}$. PCSPs were introduced in [2, 12] to unify the study of approximability of perfectly satisfiable CSPs. The PCSP framework vastly extends CSPs: Firstly, several well-known computational problems can be formulated in the former, but not in the latter. Primary examples include the approximate colouring problems, which we shall discuss in more detail later. Secondly, already the early exploration of PCSPs unveiled a number of new phenomena — absent in the non-promise setting — that quickly called for a more general and conceptually different approach to their study, going beyond the universal-algebraic approach to CSPs. The crux of this need lies in the fact that the Galois connection for PCSPs is far less structured than the one for CSPs: While the complexity of PCSPs is still governed by polymorphisms (which are now homomorphisms $\mathbf{A}^k \to \mathbf{B}$), the algebraic structure that they form does not admit composition in the promise context. A consequence of this fact is that the universal-algebraic tools that allow generating an infinite set of new identities from a single polymorphic identity fails for PCSPs. This, in turn, stimulated the use of different tools to study PCSP polymorphisms, including Boolean function analysis [13], topology [43], matrix and tensor theory [21–23], and Fourier analysis [34].

It is, of course, a very natural question whether the CSP dichotomy extends to PCSPs. Before being able to even conjecture a dichotomy for such a wide class of problems, it would be beneficial to obtain classifications in well-chosen special cases. When the Feder-Vardi conjecture was made, it was supported, in particular, by important special cases that were classified at the time: the Boolean CSP (i.e., the domain of $\mathbf{A}$ is $\{0,1\}$) [47], the afore-mentioned graph colouring problem [41], and the undirected graph homomorphism problem (i.e., $\mathbf{A}$ is an undirected graph) [35]. (Note that the second problem is a special case of the third.) The promise versions of the Boolean CSP, graph colouring, and graph homomorphism problem have been studied and there are some partial complexity classification results about them (see, e.g. [2, 4, 12, 13, 28, 43]), but full classifications even in these cases are believed to be difficult to obtain. In particular, the promise version of graph colouring is the well-known *approximate graph colouring* problem: checking whether a given graph is n-colourable or not even m-colourable, for given $n \leq m$. The complexity of this problem, that corresponds to $\mathrm{PCSP}(\mathbf{K}_n, \mathbf{K}_m)$, is a long-standing open question in computer science [29]. It was only resolved in certain special cases [4, 11, 37, 42, 43] or under strong complexity-theoretic assumptions such as variants of the Unique Games Conjecture [18, 25, 31]. Other particular examples of PCSPs have been studied in [3, 15, 17, 45]. The PCSP templates considered there have either small domains or specific structure.

The 1-IN-3 vs. NAE problem — i.e. PCSP(**1-in-3**, **NAE**) — is in many respects a prototypical example of PCSP, in that it witnesses several of the new behaviours separating the promise and the non-promise worlds. Next, we discuss three of these separating behaviours.

(i) If three structures $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are such that $\mathbf{A} \to \mathbf{C} \to \mathbf{B}$, then PCSP($\mathbf{A}, \mathbf{B}$) reduces to CSP($\mathbf{C}$) through the trivial reduction that does not change the instance.[2] Hence, if CSP($\mathbf{C}$) is tractable, the exact same algorithm solving it also solves PCSP($\mathbf{A}, \mathbf{B}$). The tractability of many PCSPs can be certified through this pattern, by exhibiting a suitable structure $\mathbf{C}$. For *finite* $\mathbf{C}$, this is provably not the case for PCSP(**1-in-3**, **NAE**): It was established in [4] that any finite structure $\mathbf{C}$ for which **1-in-3** $\to \mathbf{C} \to$ **NAE** is such that CSP($\mathbf{C}$) is NP-hard.[3] This means that for PCSP(**1-in-3**, **NAE**) the source of its tractability lies outside of the scope of non-promise finite CSPs and comes from *infinite-domain* CSPs. (We note that there is no dichotomy for infinite-domain CSPs [9], although there is a conjecture that the dichotomy for finite-domain CSPs extends to a certain well-behaved class of infinite-domain CSPs, cf. [10].)

(ii) *Local consistency* is an algorithmic technique that consists in relaxing the question "Does a homomorphism $\mathbf{X} \to \mathbf{A}$ exist?" by testing whether all subinstances of $\mathbf{X}$ of bounded size admit a system of compatible homomorphisms to $\mathbf{A}$. This method applies to both CSPs and PCSPs; the templates that can be solved by enforcing local consistency are said to have a *bounded width*. In the realm of CSPs, local consistency has precisely the same power as the linear-programming based Sherali-Adams hierarchy [48]. However, it was recently shown in [1] that PCSP(**1-in-3**, **NAE**) is solved by some fixed round of the Sherali-Adams hierarchy and yet it has unbounded width, thus implying a lack of collapse of the two algorithmic models for PCSPs.

(iii) Another singular behaviour of PCSP(**1-in-3**, **NAE**) emerged in the context of robust algorithms. The class of CSPs that can be solved *robustly* — i.e., through some algorithm that is not highly sensitive to small noise in the instance — coincides with the class of bounded-width problems [5]. Nevertheless, it was established in [14] that the problem PCSP(**1-in-3**, **NAE**) can be solved robustly although it is of unbounded width — thus yielding another discrepancy between the promise and the non-promise settings.

For these reasons, a deeper understanding of *why* problems akin to PCSP(**1-in-3**, **NAE**) are tractable can shed light on the new type of behaviours we expect to find in the complexity landscape of promise problems. To that end, in this paper we investigate the tractability boundary of problems that weaken the promise of PCSP(**1-in-3**, **NAE**) either from the left or from the right — when the weakened version of the promise continues to share key features with **1-in-3** and **NAE** (namely being *symmetric* and *rainbow-free*), we find a dichotomy for the corresponding fragments of PCSPs. We note that [17], motivated by the same goal, considered a certain class of Boolean templates that result from weakening the promise of one of the two structures. Unlike [17], we go beyond Boolean domains.

The relational structures **1-in-3** and **NAE** contain one ternary relation, which is *symmetric*, i.e., it is invariant under permutations of the arguments, and it is *rainbow-free*, i.e., it does not contain any tuple (x, y, z) whose arguments are all distinct (note that this is always the case for Boolean ternary structures). Following [3], we describe this type of relational structures by associating digraphs to them. More precisely, given a digraph $\mathbf{G} = (V, E)$, we let $\widehat{\mathbf{G}}$ be the relational structure defined by

$$\widehat{\mathbf{G}} = (V; \{(x,x,y), (x,y,x), (y,x,x) \mid (x,y) \in E\}).$$

For example, if $\mathbf{G}$ is a single directed edge (from 0 to 1) or a single undirected edge, the corresponding structure $\widehat{\mathbf{G}}$ is **1-in-3** or **NAE**, respectively. It is easy to check that we have $\mathbf{G} \to \mathbf{H}$ if and only if $\widehat{\mathbf{G}} \to \widehat{\mathbf{H}}$. Moreover, PCSP($\mathbf{G}, \mathbf{H}$) always reduces to PCSP($\widehat{\mathbf{G}}, \widehat{\mathbf{H}}$), but the latter problem can be harder — for example, if $\mathbf{G} = \mathbf{H}$ is an undirected edge, PCSP($\mathbf{G}, \mathbf{H}$) = CSP($\mathbf{G}$) is the (tractable) 2-colouring problem, whereas PCSP($\widehat{\mathbf{G}}, \widehat{\mathbf{H}}$) = CSP($\widehat{\mathbf{G}}$) = CSP(**NAE**) is NP-hard.

We are now ready to state our main results, which concern the regime where the promise of the 1-IN-3 vs. NAE problem is "broken" from either side — i.e., one of the two structures in the template (**1-in-3**, **NAE**) is replaced by a different structure.[4] First, fix **1-in-3** and consider any digraph $\mathbf{G}$ such that (**1-in-3**, $\widehat{\mathbf{G}}$) is a valid template — which happens if and only if **1-in-3** $\to \widehat{\mathbf{G}}$ or, equivalently, if and only if $\mathbf{G}$ contains at least one edge.

Theorem 1. PCSP(**1-in-3**, $\widehat{\mathbf{G}}$) *is tractable if* $\mathbf{G}$ *has a directed cycle of length at most 3, and it is* NP*-hard otherwise.*

Second, fix **NAE** and consider any digraph $\mathbf{G}$ such that ($\widehat{\mathbf{G}}$, **NAE**) is a valid template — which happens if and only if $\widehat{\mathbf{G}} \to$ **NAE** or, equivalently, if and only if the graph obtained from $\mathbf{G}$ by forgetting directions is bipartite. Following [36], we say that a digraph is *balanced* if each of its oriented cycles has as many edges in one direction as in the other.

Theorem 2. PCSP($\widehat{\mathbf{G}}$, **NAE**) *is tractable if* $\mathbf{G}$ *is balanced, and it is* NP*-hard otherwise.*

The rest of the paper is dedicated to the proofs of Theorems 1 and 2, which are given in Sections 3 and 4, respectively, after introducing some preliminary notions in Section 2. In Section 5, we shall discuss the implications of the current analysis in the context of the theory of PCSPs, and we shall outline some natural directions for future investigation.

2 PRELIMINARIES

We denote by $[n]$ the set $\{1, 2, \ldots, n\}$. Moreover, we denote by $[n, m)$ the set $\{n, \ldots, m-1\}$. We say that a collection of sets $A_1, \ldots, A_n$ is disjoint if $A_i \cap A_j = \emptyset$ whenever $i \neq j$.

[2] In the PCSP literature, $\mathbf{C}$ is known as a *sandwich*.

[3] We remark that the statement is false if we admit structures having an *infinite* domain. In fact, the algorithm based on Gaussian elimination that was described at the beginning of the Introduction can be reformulated as a reduction of PCSP(**1-in-3**, **NAE**) to CSP($\mathbf{Z}$), where $\mathbf{Z} = (\mathbb{Z}; \{(x, y, z) \in \mathbb{Z}^3 \mid x + y + z = 1\})$ satisfies **1-in-3** $\to \mathbf{Z} \to$ **NAE**.

[4] We remark that PCSPs (and CSPs) can also be formulated in their *search* version, as opposed to the *decision* versions discussed in this Introduction. The two versions are polynomial-time equivalent for CSPs [20]; for PCSPs, decision reduces to search, but it is not known whether an efficient reduction in the other direction always exists. Our results hold for both decision and search versions of PCSPs.

Relational structures and homomorphisms. Except for from digraphs, all relational structures that appear in this paper are pairs $\mathbf{A} = (A; R^{\mathbf{A}})$, where A is the domain and $R^{\mathbf{A}} \subseteq A^3$ is a ternary relation. A relational structure $\mathbf{A}$ is called *symmetric* if the relation $R^{\mathbf{A}}$ is symmetric, i.e., invariant under any permutation of its three arguments.

Given two relational structures $\mathbf{A} = (A; R^{\mathbf{A}})$ and $\mathbf{B} = (B; R^{\mathbf{B}})$, a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a function $h : A \to B$ such that $(h(x), h(y), h(z)) \in R^{\mathbf{B}}$ whenever $(x, y, z) \in R^{\mathbf{A}}$. We denote the existence of a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ by $\mathbf{A} \to \mathbf{B}$. A pair of relational structures $(\mathbf{A}, \mathbf{B})$ with $\mathbf{A} \to \mathbf{B}$ is called a (PCSP) *template*.

Polymorphisms, minions. We say that a function $f : A^n \to B$ has *arity* $\mathrm{ar}(f) = n$. For such a function, we say that a coordinate $i \in [n]$ is *essential* if there exist $a_1, \dots, a_n, a_i' \in A$ such that

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

An n-ary function f has *essential arity at most* k if it has at most k essential coordinates. Let $(\mathbf{A}, \mathbf{B})$ be a template. An n-ary function f is a *polymorphism* of $(\mathbf{A}, \mathbf{B})$ if $(x_i, y_i, z_i) \in R^{\mathbf{A}}$ for every $i \in [n]$ implies

$$(f(x_1, \dots, x_n), f(y_1, \dots, y_n), f(z_1, \dots, z_n)) \in R^{\mathbf{B}}.$$

We denote by $\mathrm{Pol}^{(n)}(\mathbf{A}, \mathbf{B})$ the set of n-ary polymorphisms of $(\mathbf{A}, \mathbf{B})$ and by $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ the set of all polymorphisms of $(\mathbf{A}, \mathbf{B})$. Polymorphisms form a minion, which we define below. Given an n-ary function $f : A^n \to B$ and a map $\pi : [n] \to [m]$, an m-ary function $g : A^m \to B$ is called a *minor* of f given by the map π if

$$g(x_1, \dots, x_m) = f(x_{\pi(1)}, \dots, x_{\pi(n)}).$$

We write $f \xrightarrow{\pi} g$ or $g = f^{\pi}$ if g is the minor of f given by the map π. A *minion* on a pair of sets (A, B) is a non-empty set of functions from A^n to B (for $n \in \mathbb{N}$) that is closed under taking minors. For any template $(\mathbf{A}, \mathbf{B})$, the set of polymorphisms $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is a minion [4]. A map $\psi : \mathscr{M} \to \mathscr{N}$ from a minion $\mathscr{M}$ to a minion $\mathscr{N}$ is a *minion homomorphism* if ψ preserves arities, i.e., $\mathrm{ar}(g) = \mathrm{ar}(\psi(g))$ for any $g \in \mathscr{M}$, and ψ preserves minors, i.e., for each $\pi : [n] \to [m]$ and each n-ary $g \in \mathscr{M}$ we have $\psi(g)(x_{\pi(1)}, \dots, x_{\pi(n)}) = \psi(g(x_{\pi(1)}, \dots, x_{\pi(n)}))$.

NP-*hardness.* Certain properties of the polymorphisms of $(\mathbf{A}, \mathbf{B})$ guarantee the NP-hardness of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ [4, 6, 16]. We use the following result from [16]; see also [6].

Theorem 3 ([16, Corollary 4.2])**.** *Let k, ℓ be integers, and let $(\mathbf{A}, \mathbf{B})$ be a template. Suppose that I is an assignment that takes any polymorphism $f \in \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ to a subset of $[\mathrm{ar}(f)]$ of size at most k. Suppose that for any chain of minors*

$$f_1 \xrightarrow{\pi_{1,2}} f_2 \xrightarrow{\pi_{2,3}} \dots \xrightarrow{\pi_{\ell-1,\ell}} f_\ell$$

where $f_1, \dots, f_\ell \in \mathrm{Pol}(\mathbf{A}, \mathbf{B})$, there exist $1 \leq i < j \leq \ell$ such that $\pi_{i,j}(I(f_i)) \cap I(f_j) \neq \emptyset$, where $\pi_{i,j} = \pi_{j-1,j} \circ \dots \circ \pi_{i,i+1}$. Then $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is NP-hard.

Intuitively, one thinks of the mapping I as nondeterministically "decoding" a polymorphism f to one of its coordinates. We bound the number of elements in $I(f)$ in order to bound the nondeterminism. For the decoding to be good enough to work, it must satisfy the chain condition in the theorem.

Another important hardness result we will use is the following, which is a direct corollary of [4, Corollary 5.3], rephrased in terms of essential arity.

Theorem 4. *Let $(\mathbf{A}, \mathbf{B})$ be a template. If $\mathrm{Pol}^{(3)}(\mathbf{A}, \mathbf{B})$ contains only non-constant functions of essential arity 1, then $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is NP-hard.*

Theorems 3 and 4 hold for both decision and search versions of PCSP.

Digraphs. Unless said otherwise, all digraphs in this paper are finite. A loopless digraph $\mathbf{G}$ is a *tournament* if, for any two distinct vertices x, y, precisely one of the pairs (x, y) and (y, x) is a directed edge of $\mathbf{G}$. A tournament is *transitive* if its edge relation is transitive. Note that a tournament is transitive if and only if it has no directed cycles of length 3. We will use the well-known result that a tournament is acyclic if and only if it is transitive, cf. [33, Corollary 5a, (1–2)]. We let $\mathbf{T}_i$ denote the transitive tournament on i vertices. Following [32], an *oriented path* or *oriented cycle* is a digraph formed by choosing an orientation for each edge of a path or cycle. The *net length* of an oriented path or cycle is the absolute value of the number of forward edges minus the number of backward edges, for an arbitrary direction of the path or cycle. (By taking the absolute value of this difference, the direction we traverse the path or cycle does not matter.) In contrast, a *directed cycle* is a digraph isomorphic to the digraph with edges $1 \to 2 \to \dots \to k \to 1$ for some $k \in \mathbb{N}$. An oriented path is *minimal* if no subpath has a strictly greater net length. For an oriented path and a fixed direction, the *level* of a vertex v, denoted $\mathrm{lvl}(v)$, is the net length of the initial part of the oriented path ending at v. We shall make use of the following result.

Theorem 5 ([32])**.** *Let $\mathbf{P}, \mathbf{P}'$ be minimal oriented paths of the same net length. Then there exists an oriented path $\mathbf{Q}$ that can be homomorphically mapped to $\mathbf{P}$ and $\mathbf{P}'$ with beginnings and ends preserved.*

Note that $\mathbf{Q}$ needs to be a minimal path of the same net length as $\mathbf{P}$ and $\mathbf{P}'$. Hence, Theorem 5 can be extended to a finite number of minimal oriented paths $\mathbf{P}, \mathbf{P}', \mathbf{P}'', \dots$ of equal net length.

For an integer $i \geq 2$, we let $\mathbf{D}_i$ be the directed cycle on i vertices and $\mathbf{L}_i$ be the directed path on i vertices. We also let $\mathbf{D}_1$ be a single vertex with a loop and $\mathbf{L}_1$ be an isolated vertex, while $\mathbf{L}_\omega = (\mathbb{N}; \{(x, x+1) \mid x \in \mathbb{N}\})$ denotes the infinite directed path. A digraph $\mathbf{G}$ is *balanced* if each of its oriented cycles has zero net length. Equivalently, this means that $\mathbf{G} \to \mathbf{L}_\omega$ [36]. We will show (cf. Proposition 8) that this is also equivalent to the condition $\widehat{\mathbf{G}} \to \mathbf{Z}$, where $\mathbf{Z}$ is the relational structure over the domain $\mathbb{Z}$ whose unique, ternary relation is $\{(x, y, z) \in \mathbb{Z}^3 \mid x + y + z = 1\}$, see Footnote 3.

Trees. We will need the following fact about binary trees that can be easily shown by induction. All of our trees will be rooted.

Lemma 6. *A binary tree with more than 2^n leaves must have a path from the root to a leaf containing at least $n + 2$ vertices.*

Primitive positive formulas, definitions. We will define the notions of primitive positive (pp) formulas and definitions for the special case of relational structures with one ternary relation symbol R. A *pp-formula* is an existentially quantified conjunction of (positive) atomic formulas, that are of the form $x = y$ or $R(x, y, z)$ for variables x, y, z. For example,

$$\exists y \exists z\, R(x, y, z) \wedge (x = y)$$

is a pp-formula. For a relational structure $\mathbf{A} = (A; R^{\mathbf{A}})$ and a pp-formula $\varphi(x, y, z)$ in which only x, y, z are free, we define $\varphi^{\mathbf{A}}$ as a ternary relation that contains a tuple (a, b, c) if and only if substituting (a, b, c) for (x, y, z), interpreting existential quantification as being over A, and interpreting R as $R^{\mathbf{A}}$ leads to a true statement. We say that φ interpreted in $\mathbf{A} = (A; R^A)$ pp-defines $\mathbf{B} = (B; R^B)$ if $A = B$ and $R^B = \varphi^{\mathbf{A}}$.

For templates $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}', \mathbf{B}')$, we say that $(\mathbf{A}, \mathbf{B})$ *pp-defines* $(\mathbf{A}', \mathbf{B}')$ whenever there exists a pp-formula $\varphi(x, y, z)$ in which precisely x, y, z are free, for which φ interpeted in $\mathbf{A}$ pp-defines $\mathbf{A}'$, and φ interpreted in $\mathbf{B}$ pp-defines $\mathbf{B}'$.

Theorem 7 ([12]). *Suppose the template $(\mathbf{A}, \mathbf{B})$ pp-defines $(\mathbf{A}', \mathbf{B}')$. Then $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}')$ reduces to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ in logarithmic space.*

3 BREAKING THE PROMISE FROM THE RIGHT

Let $\mathbf{G}$ be a digraph such that $(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ is a valid template (equivalently, the edge set of $\mathbf{G}$ is nonempty). In this section, we will prove the following result.

Theorem 1. $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ *is tractable if $\mathbf{G}$ has a directed cycle of length at most 3, and it is* NP*-hard otherwise.*

The tractability part of Theorem 1 follows from the next, graph-theoretic result.[5]

Proposition 8. *For any digraph $\mathbf{G}$, the following holds*

(i) $\widehat{\mathbf{G}} \to \mathbf{Z}$ if and only if $\mathbf{G} \to \mathbf{L}_\omega$ if and only if $\mathbf{G}$ is balanced, and
(ii) $\mathbf{Z} \to \widehat{\mathbf{G}}$ if and only if $\mathbf{D}_i \to \mathbf{G}$ for some $i \in \{1, 2, 3\}$. Moreover, if a homomorphism from $\mathbf{Z}$ to $\widehat{\mathbf{G}}$ exists, then it can be efficiently computed.

Proof. For a relational structure $\mathbf{A} = (A; R)$ with one ternary symmetric relation R, define $\tilde{\mathbf{A}}$ to be the graph (A, E) where $(x, y) \in E$ if and only if $(x, x, y) \in R$. Note that $\widehat{\mathbf{G}} \to \mathbf{A}$ if and only if $\mathbf{G} \to \tilde{\mathbf{A}}$.[6]

Observe that $\tilde{\mathbf{Z}} = (\mathbb{Z}; \{(a, 1 - 2a) : a \in \mathbb{Z}\})$. This structure is a disjoint union of (countably many) forward-infinite directed paths: First, every vertex has exactly one outgoing and at most one incoming edge (odd a has in-degree 1 while even a has in-degree 0). Second, note that $(0, 1)$ and $(1, -1)$ are edges in $\tilde{\mathbf{Z}}$ and for the remaining edges $(a, 1 - 2a)$ we have $|a| < |1 - 2a|$ which proves that forward paths lead away from 0 and that $\tilde{\mathbf{Z}}$ is acyclic. Thus $\tilde{\mathbf{Z}} \leftrightarrows \mathbf{L}_\omega$. It follows that $\mathbf{G} \to \tilde{\mathbf{Z}}$ if and only if $\mathbf{G} \to \mathbf{L}_\omega$; respectively, $\tilde{\mathbf{Z}} \to \mathbf{G}$ if and only if $\mathbf{L}_\omega \to \mathbf{G}$.

(i) $\widehat{\mathbf{G}} \to \mathbf{Z}$ if and only if $\mathbf{G} \to \tilde{\mathbf{Z}}$, if and only if $\mathbf{G} \to \mathbf{L}_\omega$. For the second equivalence: as $\mathbf{G}$ is finite, $\mathbf{G} \to \mathbf{L}_\omega$ if and only if $\mathbf{G} \to \mathbf{L}_{|G|}$, which is equivalent to $\mathbf{G}$ being balanced by e.g. [36, Proposition 1.13].

(ii) For the "if" direction: a homomorphism $h_i : \mathbf{Z} \to \widehat{\mathbf{D}}_i$ exists for $i \in [3]$ (namely $h_1(x) = 0, h_2(x) = [x > 0], h_3(x) = x \bmod 3$), so $\mathbf{D}_i \to \mathbf{G}$ for $i \in [3]$ implies $\mathbf{Z} \to \widehat{\mathbf{D}}_i \to \widehat{\mathbf{G}}$. Conversely, let $f : \mathbf{Z} \to \widehat{\mathbf{G}}$ be a homomorphism, and assume that f is surjective (we can take the image of the map in place of $\widehat{\mathbf{G}}$) and that $\mathbf{D}_1$ and $\mathbf{D}_2$ do not map to $\mathbf{G}$. We will show that $\mathbf{G}$ is a non-transitive tournament; as such, the graph must contain $\mathbf{D}_3$ and the proof is finished. Take any $v \neq v'$ vertices of $\mathbf{G}$ and find a, a' such that $f(a) = v, f(a') = v'$. The triplet $(a, a', 1 - a - a')$ belongs to the relation of $\mathbf{Z}$. Therefore, either (v, v') or (v', v) is an edge of $\mathbf{G}$ — since $\mathbf{G}$ additionally does not contain directed cycles of length at most 2 by assumption, we have that $\mathbf{G}$ is a tournament. Since $f : \mathbf{Z} \to \widehat{\mathbf{G}}$, we have $\tilde{\mathbf{Z}} \to \mathbf{G}$ and hence $\mathbf{L}_\omega \to \mathbf{G}$, so by the pigeonhole principle $\mathbf{G}$ must contain a directed cycle and, thus, it cannot be transitive.

Finally, if $\mathbf{Z} \to \widehat{\mathbf{G}}$, using the second part of the lemma we deduce that $\mathbf{D}_i \to \mathbf{G}$ (and hence $\widehat{\mathbf{D}}_i \to \widehat{\mathbf{G}}$) for some $i \in \{1, 2, 3\}$. Both homomorphisms $\widehat{\mathbf{D}}_i \to \widehat{\mathbf{G}}$ and $\mathbf{Z} \to \widehat{\mathbf{D}}_i$ are efficiently computable, and their composition yields a concrete homomorphism from $\mathbf{Z}$ to $\widehat{\mathbf{G}}$. □

Proof of tractability in Theorem 1. If $\mathbf{D}_i \to \mathbf{G}$ for some $i \in \{1, 2, 3\}$, it follows from Proposition 8 that $\mathbf{1\text{-}in\text{-}3} \to \mathbf{Z} \to \widehat{\mathbf{G}}$. Therefore, $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ trivially reduces to $\mathrm{CSP}(\mathbf{Z})$, which is tractable as it corresponds to solving linear Diophantine systems [40]. □

In the rest of this section, we prove the hardness part of Theorem 1. We note that this proof can be seen as a generalisation of the proof of hardness of $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{T}}_3)$ from [3], where $\widehat{\mathbf{T}}_k$ is a directed tournament on k vertices.[7] To that end, fix $\mathbf{G}$ and suppose it contains no directed cycles of length at most 3. If we consider the edge relation of $\mathbf{G}$, we see that it is irreflexive (since $\mathbf{G}$ has no loops) and antisymmetric (since $\mathbf{G}$ has no directed cycles of length 2). Unfortunately it is not transitive — nonetheless we will see the edge relation as a kind of weak order relation, and we will prove that it is "transitive enough" for our purposes. Thus, write $<$ for the edge relation of $\mathbf{G}$ (keeping in mind that $<$ is not in general transitive), and define $\leq, >, \geq$ in the obvious way. Write $x \simeq y$ if $x = y$ or $x < y$ or $x > y$.

In this section, all polymorphisms will be from the polymorphism minion $\mathrm{Pol}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$. Note that such polymorphisms are simply functions from $\{0, 1\}^n$ to V, the vertex set of $\mathbf{G}$. We can see such a function as a function from $2^{[n]}$ to V, i.e. from subsets of $[n]$ to V. The fact that $f \in \mathrm{Pol}^{(n)}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ is a polymorphism then implies that, for any partition A, B, C of $[n]$, two of $f(A), f(B), f(C)$ are equal, and the last is strictly greater (i.e., there is an edge in $\mathbf{G}$ from the two equal elements to the third one). In this interpretation, for $\pi : [n] \to [m]$, observe that $f^\pi = f \circ \pi^{-1}$, where π^{-1} is the preimage of π.

Definition 9. For a polymorphism $f \in \mathrm{Pol}^{(n)}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$, call a set $X \subseteq [n]$ a *hitting set* (for f) if it has a subset $Y \subseteq X$ such that,

[5] The second part of the proposition that follows may be derived from a result in [15]. We include a shorter, self-contained proof for completeness.
[6] In other words, $\widehat{\cdot}$ and $\tilde{\cdot}$ form a *Galois connection* under the preordering given by $\to$.
[7] $\widehat{\mathbf{T}}_3$ is denoted by $\mathbf{T}_1$ in [3].

for all $Z \subseteq [n]$ with $f(Z) = f(Y)$, we have $X \cap Z \neq \emptyset$. In this case, say that $f(Y)$ is a *hitting value* for X.

Note that the property of being a hitting set is upwards closed; i.e., if X is hitting and $W \supseteq X$, then W is hitting as well.

Theorem 10. *Let* $\mathbf{G}$ *be a digraph without directed cycles of length at most 3 and let* N *be the number of vertices of* $\mathbf{G}$*. Then every polymorphism* $f \in \mathrm{Pol}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ *has a nonempty hitting set of size at most* $(N+1)(1+N+N^2 2^N)$.

As we shall see next, the result above is sufficient for establishing the NP-hardness part of Theorem 1.

Proof of Theorem 1. For any $f \in \mathrm{Pol}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$, define $I(f)$ to be a hitting set for f of minimum size — it exists and its size is bounded by some function of N by Theorem 10. Observe that, as $\mathbf{G}$ is fixed, N is a constant, so the size of $I(f)$ is bounded by a constant. Moreover, it is nonempty as $\emptyset$ is never a hitting set. Consider any $N+1$ polymorphisms of $\mathrm{Pol}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ connected by a chain of minors. By the pigeonhole principle, polymorphisms $f \xrightarrow{\pi} g$ must exist within this chain such that $I(f)$ and $I(g)$ have a hitting value in common. Suppose this value is c; thus $I(f), I(g)$ intersect all sets X for which $f(X) = c$, and both contain such a set. Suppose that $X \subseteq I(g)$ has $f(X) = c$; thus $f(\pi^{-1}(X)) = c$ and $I(f) \cap \pi^{-1}(X) \neq \emptyset$. Thus $\pi(I(f)) \cap I(g) \neq \emptyset$. By Theorem 3 (with $\ell = N+1$ and $k = (N+1)(1+N+N^2 2^N)$), it follows that $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ is NP-hard. □

In the remainder of this section, we shall prove Theorem 10. Henceforth, we fix a directed graph $\mathbf{G}$ with N vertices and no directed cycles of length at most 3, and a polymorphism $f \in \mathrm{Pol}^{(n)}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$. We first show that, in this case, $[n]$ is a hitting set for f.

Lemma 11. $[n]$ *is a hitting set for* f.

Proof. Since f is a polymorphism, the tuple

$$(f(\emptyset), f(\emptyset), f([n]))$$

must belong to the relation of $\widehat{\mathbf{G}}$, which means that $f(\emptyset) < f([n])$. Since $\mathbf{G}$ has no loops, $f(\emptyset) \neq f([n])$. We then have from Definition 9 that $[n]$ is a hitting set, as witnessed by the subset $[n] \subseteq [n]$. □

Next, we give two laws governing f.

Lemma 12 (Disjointness law)**.** *For disjoint* $A, B \subseteq [n]$*, we have* $f(A) \simeq f(B)$.

Proof. Consider the partition $A, B, [n] \setminus (A \cup B)$. Since f is a polymorphism, the tuple

$$(f(A), f(B), f([n] \setminus (A \cup B)))$$

belongs to the relation of $\widehat{\mathbf{G}}$. Hence, either $f(A) = f(B) < f([n] \setminus (A \cup B))$, or $f(A) = f([n] \setminus (A \cup B)) < f(B)$, or $f(B) = f([n] \setminus (A \cup B)) < f(A)$. □

Lemma 13 (Union law)**.** *Consider disjoint* $A, B, C \subseteq [n]$*. Let* M *be the multiset* $[f(A), f(B), f(A \cup C), f(B \cup C)]$*. There exists an element* m *of* M *with multiplicity 2 or 4, which we call the* pseudo-minimum *of* M*. If* m *has multiplicity 2 and* $\hat{m}, \tilde{m}$ *are the remaining (possibly equal) elements of* M*, then at least one of the following occurs:*

- $m < \hat{m}$ *and* $m < \tilde{m}$*, or*
- $m < \hat{m} < \tilde{m}$*, or*
- $m < \tilde{m} < \hat{m}$.

Proof. Let $D = [n] \setminus (A \cup B \cup C)$. By disjointness, $f(A \cup C) \simeq f(B)$ and $f(A) \simeq f(B \cup C)$. If neither of these pairs have equal values, then one element in each pair is equal to $f(D)$ (due to the partitions $A \cup C, B, D$ and $A, B \cup C, D$), and is less than the other value in the pair. This corresponds to the case where $m < \hat{m}$ and $m < \tilde{m}$. Now assume that the elements in at least one of the pairs coincide — say $f(A) = f(B \cup C)$. If the elements of the other pair are not equal, then one of the two values is equal to $f(D)$, and the other is greater; since $f(A) < f(D)$ (due to partition $A, B \cup C, D$), this corresponds to the case where $m < \tilde{m} < \hat{m}$ or the case where $m < \hat{m} < \tilde{m}$. If also the other pair has equal values, then by disjointness $f(A) \simeq f(B)$. If $f(A) \neq f(B)$ we get the case where $m < \hat{m}$ and $m < \tilde{m}$; and if $f(A) = f(B)$ then all four elements of M coincide, so we are in the case where m has multiplicity 4. □

Remark 14. The pseudo-minimum is not necessarily a true minimum, since in the case where $m < \hat{m} < \tilde{m}$ (or symmetrically $m < \tilde{m} < \hat{m}$) it is possible that $m \not\leq \tilde{m}$ (respectively $m \not\leq \hat{m}$). Nonetheless, since $\mathbf{G}$ has no directed cycles of length 3, we have that $m \not> \tilde{m}$ (respectively, $m \not> \hat{m}$) even in this case. Indeed, in all cases, for any $m' \in M$ we have that $m \not> m'$, since otherwise a directed cycle of length at most 3 would appear.

Our final goal is to establish that f has a hitting set of bounded size. The following lemma shows that a long sequence of sets whose images under f are strictly increasing yields a hitting set.

Lemma 15. *Suppose there exist sets* $X_1, \ldots, X_k \subseteq [n]$ *with* $k > N$*, where* $f(X_1) < \ldots < f(X_k)$*. Then* $\bigcup_\ell X_\ell$ *is a hitting set.*

Proof. Suppose for contradiction that for all $i, j \in [k]$ we have $f(X_i) \simeq f(X_j)$. Thus, $f(X_1), \ldots, f(X_k)$ induce a tournament in $\mathbf{G}$. This tournament is not acyclic, since $f(X_1) < \ldots < f(X_k)$ must contain a directed cycle by the pigeonhole principle. Therefore, $\mathbf{G}$ is not transitive, which means that it must contain a directed cycle of length at most 3, a contradiction.

Thus, there exist $i, j \in [k]$ such that $f(X_i) \not\simeq f(X_j)$. By (the contrapositive of) the disjointness law, any set $Y \subseteq [n]$ such that $f(Y) = f(X_i) \not\simeq f(X_j)$ is not disjoint from X_j. Hence, $X_i \cup X_j$ is a hitting set. Since hitting sets are upwards closed and $\bigcup_\ell X_\ell \supseteq X_i \cup X_j$, we find that $\bigcup_\ell X_\ell$ is a hitting set, too. □

The next three lemmata will partially determine $f(X \cup S \cup T)$ in terms of $f(X \cup S)$ and $f(X \cup T)$, under the assumption that $f(X) = f(Y)$ and X, Y, S, T are disjoint. These proofs are based on repeated applications of the union law, and involve a case analysis.

Lemma 16. *For disjoint* X, Y, S*, if* $f(X \cup S) \not> f(X) = f(Y)$ *then* $f(X \cup S) = f(Y \cup S)$.

Proof. By the disjointness law, $f(X \cup S) \simeq f(Y) = f(X)$. Since $f(X \cup S) \not> f(X)$, we have $f(X \cup S) \leq f(X)$. Apply the union law to X, Y, and S.

Suppose first that the pseudo-minimum of the multiset

$$[f(X), f(Y), f(X \cup S), f(Y \cup S)]$$

is $f(X) = f(Y)$. Note that $f(X \cup S) \leq f(X)$, and since $f(X)$ is the pseudo-minimum, $f(X \cup S) \not< f(X)$. So $f(X \cup S) = f(X)$. Since the pseudo-minimum must appear 2 or 4 times, it follows that $f(Y \cup S) = f(X) = f(Y) = f(X \cup S)$, as required.

Otherwise, if the pseudo-minimum is $f(X \cup S)$ (or symmetrically $f(Y \cup S)$), since the minimum must appear 2 or 4 times, and $f(X) = f(Y)$ is not the pseudo-minimum, we have that $f(X \cup S) = f(Y \cup S)$. □

Lemma 17. *If X, Y, S, T are disjoint, $f(X \cup S) = f(X \cup T)$, and $f(X) = f(Y)$, then either $f(X) \leq f(X \cup S \cup T)$ or $f(X \cup S) < f(X \cup S \cup T)$.*

Proof. Apply the union law to X, Y, and S. Since $f(X) = f(Y)$, there are two possibilities for the pseudo-minimum of

$$[f(X), f(Y), f(X \cup S), f(Y \cup S)].$$

Case 1. Assume that the pseudo-minimum is $f(X \cup S) = f(Y \cup S)$. Since $f(X) = f(Y)$, we deduce that $f(X \cup S) = f(Y \cup S) \leq f(X) = f(Y)$. Apply the union law to Y, $X \cup T$, and S. Since $f(X \cup T) = f(X \cup S) = f(Y \cup S) \leq f(Y)$, the pseudo-minimum of

$$[f(Y), f(X \cup T), f(Y \cup S), f(X \cup T \cup S)]$$

is $f(X \cup T) = f(Y \cup S)$. We thus have four cases:
- $f(X \cup T) = f(Y \cup S) = f(Y) = f(X \cup T \cup S)$,
- $f(Y) > f(X \cup T) = f(Y \cup S) < f(X \cup T \cup S)$,
- $f(X \cup T) = f(Y \cup S) < f(Y) < f(X \cup T \cup S)$,
- $f(X \cup T) = f(Y \cup S) < f(X \cup T \cup S) < f(Y)$.

Keeping in mind that $f(X) = f(Y)$ and $f(X \cup S) = f(Y \cup S)$, the conclusion follows in all 4 cases.

Case 2. Assume that the pseudo-minimum is equal to neither $f(X \cup S)$ nor $f(Y \cup S)$. Thus the pseudo-minimum is $f(X) = f(Y)$. In this case, $f(Y) \not> f(X \cup S) = f(X \cup T)$ and $f(Y) \not> f(Y \cup S)$. By assumption, $f(Y) \neq f(X \cup T)$ and $f(Y) \neq f(Y \cup S)$. By disjointness, $f(Y) \simeq f(X \cup T)$ and $f(Y) = f(X) \simeq f(Y \cup S)$, so it follows that $f(Y) < f(X \cup T)$ and $f(Y) < f(Y \cup S)$. Now, apply the union law to Y, $X \cup T$, and S. Since $f(Y) < f(X \cup T)$ and $f(Y) < f(Y \cup S)$, the pseudo-minimum of

$$[f(Y), f(X \cup T), f(Y \cup S), f(X \cup T \cup S)]$$

must be $f(Y) = f(X \cup T \cup S)$, and the conclusion follows. □

Lemma 18. *If X, Y, S, T are disjoint and $f(X) = f(Y) = f(X \cup S) = f(X \cup T)$, then $f(X) = f(X \cup S \cup T)$.*

Proof. Since $\mathbf{G}$ has no loops, $f(X \cup S) \not> f(X)$ and $f(X \cup T) \not> f(X)$. Therefore, by Lemma 16, $f(Y \cup S) = f(X \cup S) = f(X)$ and $f(Y \cup T) = f(X \cup T) = f(X)$. Let us now apply the union law to Y, $X \cup S$, and T. Since three of the elements of $[f(Y), f(X \cup S), f(Y \cup T), f(X \cup S \cup T)]$ are equal, the fourth is also. □

The next lemma is the crucial one: Given some nonempty set X, it either creates a hitting set of bounded size immediately, or finds some set X' of bounded size such that $f(X) < f(X')$. The conclusion will follow by repeatedly applying this lemma.

Lemma 19. *Consider a nonempty set $X \subseteq [n]$. Then either a hitting set of size at most $1 + |X| + 2^N$ exists, or we can find a nonempty set $X' \subseteq [n]$ of size at most $1 + |X| + N2^N$ such that $f(X) < f(X')$.*

Proof. We can assume that $|X| \leq n - 2$ as, otherwise, it would follow from Lemma 11 that $[n]$ is the required hitting set. Thus, let $a \neq b \in [n] \setminus X$, and suppose that $X \cup \{a, b\}$ is not hitting (if it were, it would be the required hitting set). Thus, some set $Y \subseteq [n] \setminus (X \cup \{a, b\})$ exists such that $f(X) = f(Y)$. Let $S = [n] \setminus (X \cup Y \cup \{a\})$; since $b \in S$, $S \neq \emptyset$.

Create a partition $P_1, \ldots, P_k$ of S, that initially consists of one singleton for each element of S. While

(i) all the parts P such that $f(X \cup P) < f(X)$ have size $|P| \leq 2^{N-1}$, and
(ii) there exist distinct parts P_i, P_j such that $f(X \cup P_i) = f(X \cup P_j) < f(X)$,

merge any two such parts P_i and P_j. Observe that at all times all parts must have size at most 2^N, since they are either singletons or the union of two parts with size at most 2^{N-1}. It follows that this procedure must eventually terminate; we now consider what happens when it does, depending on the reason for termination.

First, suppose that the procedure terminates because (i) ceases to hold; i.e., we arrive at a part P with size greater than 2^{N-1} with $f(X \cup P) < f(X)$. P was created by repeated merges of parts in the partition; thus, consider a binary tree rooted at P where each vertex is labelled by a subset of P that was at some point a part in the partition, and the children of a vertex are the parts that were merged to form that part. Consider any non-root vertex in the tree, and suppose it is labelled by part P'. Since P' was merged with some other part, it must be the case that $f(X \cup P') < f(X)$. Since this is true by assumption for P as well, it is true for all the parts that appear in the tree. Now, consider any non-leaf vertex, labelled by part $Q \cup R$, where its children are labelled by Q and R. Since Q and R are merged, $f(X \cup Q) = f(X \cup R)$ and $Q \cap R = \emptyset$. Thus, apply Lemma 17 to X, Y, Q, R and note that $f(X) \not\leq f(X \cup Q \cup R)$, since $f(X \cup Q \cup R) < f(X)$; it follows that $f(X \cup Q) < f(X \cup Q \cup R)$ and $f(X \cup R) < f(X \cup Q \cup R)$. In other words, for any two parts A and B where the vertex corresponding to A is a child of the vertex corresponding to B, we have $f(X \cup A) < f(X \cup B)$. Since the tree has more than 2^{N-1} leaves, by Lemma 6 we can find a path in the tree starting at the root with at least $N + 1$ vertices. Call the labels of the vertices of such path $P'_k = P, P'_{k-1}, \ldots, P'_1$, starting from the root and going to the leaves. Since $f(X \cup P'_1) < f(X \cup P'_2) < \cdots < f(X \cup P'_k)$, we obtain from Lemma 15 that the set $X \cup \bigcup_{\ell \in [k]} P'_\ell$ is a hitting set. Using that P is a superset of all sets appearing in the path and that hitting sets are upwards closed, we conclude that $X \cup P$ is a hitting set. Recall that $|P| \leq 2^N$. So, the required hitting set is $X \cup P$.

Second, suppose that the procedure terminates because the condition (ii) ceases to hold (regardless of whether (i) holds or not). By the pigeonhole principle there exist at most N parts P for which $f(X \cup P) < f(X)$. Let Z be the union of these parts and the set $\{a\}$ (if there are not any such parts then $Z = \{a\}$), and let $Q_1, \ldots, Q_\ell$ be the remaining parts; i.e., $f(X \cup Q_i) \not< f(X)$. Observe that $|Z| \leq 1 + N2^N$. There are now two cases. First, suppose that for some $i \in [\ell]$ we have $f(X) < f(X \cup Q_i)$. In this case, since $|X \cup Q_i| \leq |X| + 2^N$, the set $X' = X \cup Q_i$ witnesses that the statement of the lemma

holds. Otherwise, for every $i \in [\ell]$, since $f(X) \not\leq f(X \cup Q_i)$ and $f(X \cup Q_i) \not\leq f(X)$, yet $f(X) = f(Y) \simeq f(X \cup Q_i)$ by disjointness, we find that $f(X) = f(X \cup Q_i)$. By Lemma 18 applied $\ell - 1$ times, $f(X \cup \bigcup_i Q_i) = f(X) = f(Y)$. (This also holds when there are no sets Q_i, since $X \cup \bigcup_i Q_i = X$ in this case.) Now, the partition $X \cup \bigcup_i Q_i, Y, Z$ implies that $f(X \cup \bigcup_i Q_i) = f(Y) < f(Z)$ (as f is a polymorphism), and thus $f(X) = f(Y) < f(Z)$. Thus, as $|Z| \leq 1 + N2^N$, the set $X' = Z$ witnesses that the statement of the lemma holds. □

Proof of Theorem 10. Create a sequence of nonempty sets

$$X_1, \dots, X_{N+1} \subseteq [n]$$

in the following way. Let $X_1 = \{1\}$. For $i \in [N]$, apply Lemma 19 to X_i. If it yields a nonempty hitting set of size at most $1 + |X_i| + 2^N$, then let X_{i+1} be this hitting set. Otherwise, if it yields a nonempty set X' of size at most $1 + |X_i| + N2^N$ for which $f(X_i) < f(X')$, set $X_{i+1} = X'$. Note that $|X_{i+1}| \leq 1 + |X_i| + N2^N$ and $|X_1| = 1$, so, for all $i \in [N]$, $|X_{i+1}| \leq 1 + i(1 + N2^N) \leq 1 + N(1 + N2^N) = 1 + N + N^2 2^N$. If X_ℓ is a hitting set for some $\ell \in [N+1]$, the conclusion follows. Otherwise, we must have $f(X_1) < f(X_2) < \dots < f(X_{N+1})$, in which case, by Lemma 15, $\bigcup_i X_i$ is a hitting set. Since $|\bigcup_i X_i| \leq (N+1)(1 + N + N^2 2^N)$, the conclusion follows in this case, too. □

4 BREAKING THE PROMISE FROM THE LEFT

Let $\mathbf{G}$ be a digraph such that $(\widehat{\mathbf{G}}, \mathbf{NAE})$ is a valid template (equivalently, the graph obtained from $\mathbf{G}$ by forgetting the directions is bipartite; in this case, we say that $\mathbf{G}$ is bipartite). In this section, we will prove the following result.

Theorem 2. $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$ *is tractable if* $\mathbf{G}$ *is balanced, and it is* NP*-hard otherwise.*

The tractability part is a direct application of Proposition 8.

Proof of tractability in Theorem 2. If $\mathbf{G}$ is balanced, applying both parts of Proposition 8, we find

$$\widehat{\mathbf{G}} \to \mathbf{Z} \to \widehat{\mathbf{D}}_2 = \mathbf{NAE}.$$

Therefore, $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$ reduces to $\mathrm{CSP}(\mathbf{Z})$ and is thus tractable. □

We now turn to prove the hardness part of Theorem 2. Suppose that $\mathbf{G}$ is unbalanced — i.e., $\mathbf{G} \not\to \mathbf{L}_\omega$. Note that, since $\mathbf{G}$ is bipartite, the net length of any oriented cycle is even. The proof of hardness in Theorem 2 shall follow from the combination of the next two facts.

Proposition 20. *For any positive integer* k, $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ *is* NP*-hard.*

Proposition 21. *For any bipartite digraph* $\mathbf{G}$ *that contains an oriented cycle with net length* $2k \neq 0$, $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ *reduces in polynomial time to* $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$.

These two results will be proved in Section 4.1 and Section 4.2, respectively.

Proof of hardness in Theorem 2. Suppose that $\mathbf{G}$ contains an oriented cycle of nonzero net length. Since $\mathbf{G}$ is bipartite, this net length must be even, say $2k$. By Proposition 21, $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ reduces to $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$. NP-hardness of $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$ then follows by Proposition 20. □

4.1 Hardness of cycle vs. NAE

In this section, we prove the following result.

Proposition 20. *For any positive integer* k, $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ *is* NP*-hard.*

In the following proof, we make use of the fact that, for $f \in \mathrm{Pol}^{(3)}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$, if

$$f(x, y, z) = f(x', y', z')$$

$$(x, x', x''), (y, y', y''), (z, z', z'') \in R^{\widehat{\mathbf{D}}_{2k}},$$

then $f(x, y, z) \neq f(x'', y'', z'')$. This fact follows directly from the definitions of polymorphisms and of $\mathbf{NAE}$. Furthermore, clearly, if we have $f(x, y, z) \neq f(x', y', z') \neq f(x'', y'', z'')$, then $f(x, y, z) = f(x'', y'', z'')$, since $\mathbf{NAE}$ is Boolean. Finally, since $\widehat{\mathbf{D}}_{2k}$ can be described as the template whose domain is $[0, 2k)$ and whose relation contains all permutations of tuples of the form $(x, x, x + 1 \bmod 2k)$, we will consider addition over $[0, 2k)$ to be done modulo $2k$. In particular, these facts imply that $f(x, y, z) \neq f(x + 1, y + 1, z + 1)$ for $x, y, z \in [0, 2k)$.

Proof. The result clearly holds for $k = 1$, since in this case $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE}) = \mathrm{CSP}(\mathbf{NAE})$, which is NP-hard. Thus, assume $k \geq 2$. We show that every ternary polymorphism of $(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ is non-constant and has essential arity 1. This is sufficient for hardness by Theorem 4. Since $f(x, y, z) \neq f(x+1, y+1, z+1)$ for any $x, y, z \in [0, 2k)$, we deduce that $f(x, y, z) = z + f(x - z, y - z, 0) \bmod 2$ for $x, y, z \in [0, 2k)$. It thus follows that it is sufficient to describe the matrix $M_{xy} = f(x, y, 0)$ in order to characterise f entirely. We now observe the following.

(i) If M contains two adjacent equal elements, then the row or column on which they are found is constant. To see why, suppose without loss of generality that $f(x, y, 0) = M_{xy} = M_{x,y+1} = f(x, y + 1, 0)$ for some $x, y \in [0, 2k)$. Thus $f(x + 1, y, 1) \neq f(x, y, 0)$. Furthermore, $f(x, y-1, 0) \neq f(x+1, y, 1)$. Thus, $M_{x,y-1} = f(x, y - 1, 0) = f(x, y, 0) = M_{xy}$. Repeating this observation $2k - 2$ times yields the result.

(ii) If a row or column of M is non-constant, then it alternates between 0 and 1. This is just the contrapositive of (i).

(iii) The following configurations cannot appear in M:

$$\begin{bmatrix} 0 & \star & \star \\ \star & \star & 1 \\ \star & 1 & \star \end{bmatrix}, \begin{bmatrix} 1 & \star & \star \\ \star & \star & 0 \\ \star & 0 & \star \end{bmatrix}.$$

Equivalently, if $M_{x,y+1} = M_{x+1,y}$ then $M_{x-1,y-1} = M_{x,y+1}$. To see why this is the case, note that $f(x, y+1, 0) = M_{x,y+1} = M_{x+1,y} = f(x+1, y, 0)$, so $M_{x+1,y} = f(x+1, y, 0) \neq f(x, y, 1)$. Furthermore, $M_{x-1,y-1} = f(x - 1, y - 1, 0) \neq f(x, y, 1)$.

These facts together show that M is completely determined by its 2×2 submatrix M' located in the upper-left corner of M: These four entries dictate the rows and columns they are on by (i) and (ii),

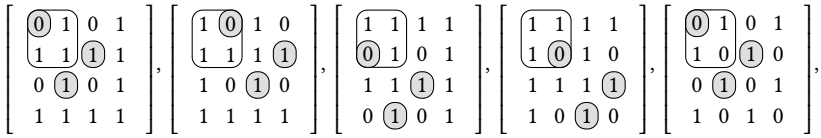

Figure 1: Patterns from the proof of Proposition 20.

and these rows and columns determine all of M in the same way. By propagating out in this way, one easily shows that, if

$$M' \in \left\{ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \right\},$$

then f is non-constant and has essential arity 1.[8] For the remaining ten configurations, by propagating out we see that the 4×4 submatrix located in the upper-left corner of M must follow one of the five patterns in Figure 1 or one of the patterns obtained from those in Figure 1 by swapping 0s and 1s. These patterns do not respect rule (iii) due to the shaded elements. □

4.2 Reduction

In this section, we prove the following result.

Proposition 21. *For any bipartite digraph* $\mathbf{G}$ *that contains an oriented cycle with net length* $2k \neq 0$, $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ *reduces in polynomial time to* $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$.

In order to design the reduction, we will need a key insight, which we prove now. The reasoning builds up on an idea in [49]. In the following, for two oriented paths $\mathbf{P} = p_1 \dots p_n$ and $\mathbf{Q} = q_1 \dots q_m$, we let $\mathbf{P} + \mathbf{Q}$ denote the path formed by identifying p_n with q_1, and we let $\mathbf{P} - \mathbf{Q}$ denote the path formed by identifying p_n with q_m. Furthermore, recall that $\mathbf{L}_2$ denotes the oriented path $p_1 p_2$ with one directed edge (p_1, p_2).

Proposition 22. *Let* $\mathbf{C}$ *be an oriented cycle of net length* $n \geq 1$. *There exists an oriented path* $\mathbf{P}$ *and a set of vertices* $a_0, \dots, a_{n-1}$ *of* $\mathbf{C}$ *such that*

(i) for each i, $\mathbf{P} - \mathbf{P}$ *homomorphically maps into* $\mathbf{C}$ *in a way that its endpoints map to* a_i, *and*
(ii) for each i, $\mathbf{L}_2 + \mathbf{P} - \mathbf{P}$ *homomorphically maps into* $\mathbf{C}$ *in a way that its first vertex maps to* a_i *and its last vertex maps to* $a_{i+1 \bmod n}$.

Proof. We first choose a positive orientation of $\mathbf{C}$, i.e., an orientation such that the difference between the numbers of edges directed forward and backward is positive (and thus equal to the net length).

We now proceed to choose a "starting point". Label the vertices of $\mathbf{C}$ as $c_0, c_1, c_2, \dots, c_p = c_0$, where c_0 is an arbitrary vertex and the other vertices are picked according to the positive orientation, in the natural way. Let $\tilde{\mathbf{C}}$ be the oriented path obtained from $\mathbf{C}$ by deleting the edge connecting c_0 and c_{p-1}, let μ be the minimum level of the vertices of $\tilde{\mathbf{C}}$, and let v be the last vertex of $\tilde{\mathbf{C}}$ of level μ.

Let now $\mathbf{C}^*$ be an oriented path starting at v and wounding around $\mathbf{C}$ a certain amount of times (the number is inessential, as long as it is sufficiently large) in the positive orientation. Consider the levels of vertices in the oriented path $\mathbf{C}^*$. Clearly $\mathrm{lvl}(v) = 0$; the choice of v guarantees that the levels of all other vertices are strictly positive. (Indeed, the first time we wind around $\mathbf{C}$ this is the case, and every successive time we wind around $\mathbf{C}$ the levels are increased by the net length of $\mathbf{C}$, which is positive.) Further, for $i = 0, \dots, n$, we let a_i be the last vertex in $\mathbf{C}^*$ of level i. Note that $a_0, \dots, a_{n-1}$ appear in the first copy of $\mathbf{C}$. Moreover, $a_0 = v$, while a_n coincides with the duplicate of v in the second copy of $\mathbf{C}$. It remains to find the path $\mathbf{P}$.

First, for every $i = 0, \dots, n-1$, we will find a minimal path $\mathbf{P}_i$ of net length p so that $\mathbf{P}_i - \mathbf{P}_i$ connects a_i to itself. This is very easy: We start with a_i and go along $\mathbf{C}^*$ until we hit the first element of level $i + p$. This is our $\mathbf{P}_i$; the path is minimal since no element after a_i has level i, and since we chose the first element of level $i + p$ to be the end. Moreover, a_i is clearly connected to itself via $\mathbf{P}_i - \mathbf{P}_i$ in $\mathbf{C}$.

Next, for each $i = 0, \dots, n-1$, we let a'_i be the element on $\mathbf{C}^*$ coming after a_i. Clearly, by the definition of a_i, $\mathrm{lvl}(a'_i) = i + 1$, and all the vertices of $\mathbf{C}^*$ after a'_i have level $\geq i + 1$. As before, we go along $\mathbf{C}^*$ until we hit the first vertex of level $i + 1 + p$, say b_i. Let $\mathbf{Q}_i$ be the part of $\mathbf{C}^*$ connecting a'_i to b_i; it has net length p and it is minimal. By the choice of p, along the way, we hit the element a_{i+1}. Let $\mathbf{Q}'_i$ be the part of $\mathbf{Q}_i$ connecting a_{i+1} to b_i; it is clearly minimal by the choice of a_{i+1} and b_i, and it has net length p. Additionally, $\mathbf{L}_2 + \mathbf{Q}_i - \mathbf{Q}'_i$ connects a_i to a_{i+1} by construction.

Now let $\mathbf{P}$ be the oriented path homomorphically mapping to every $\mathbf{P}_i$, $\mathbf{Q}_i$, and $\mathbf{Q}'_i$ (it exists by Theorem 5). Using that a_0 and a_n are the same vertex in $\mathbf{C}$, it follows that $\mathbf{P}$ satisfies the conclusion of the lemma. □

Remark 23. Proposition 22 also holds if one replaces $\mathbf{C}$ with any directed graph $\mathbf{G}$ that contains an oriented cycle of net length $n \geq 1$. Indeed, one only needs to apply the result to any oriented cycle contained in $\mathbf{G}$.

Reduction. We reduce $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ to $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$. For any oriented path $\mathbf{P}$ with vertex set $\{v_1, \dots, v_n\}$, whose first vertex is v_1, whose last vertex is v_n, and whose edge set is $E(\mathbf{P})$, define the pp-formula $x \xrightarrow{\mathbf{P}} y$ by

$$\exists u_1, \dots, u_n \; (x = u_1) \wedge (y = u_n) \wedge \bigwedge_{(v_i, v_j) \in E(\mathbf{P})} R(u_i, u_i, u_j).$$

This pp-formula is such that, in the structure $\widehat{\mathbf{G}}$, $x \xrightarrow{\mathbf{P}} y$ is true if and only if $\mathbf{P}$ homomorphically maps into $\mathbf{G}$ in a way that v_1 is mapped to x and v_n is mapped to y.

Lemma 24. *There is a pp-formula* φ *such that*

(i) φ *interpreted in* $\widehat{\mathbf{D}}_i$ *pp-defines* $\widehat{\mathbf{D}}_i$ *for any* $i \in \mathbb{N}$, *and*

[8] In particular, f is one of the 6 functions $(x_1, x_2, x_3) \mapsto x_i + c \bmod 2$, for $i \in [3], c \in [2]$.

(ii) if φ interpreted in $\widehat{\mathbf{G}}$ pp-defines $\mathbf{T}$, then $\widehat{\mathbf{D}}_{2k}$ homomorphically maps into $\mathbf{T}$.

Proof. Consider the path $\mathbf{P}$ for $\mathbf{G}$ provided by Proposition 22 (see Remark 23), and define the formula $\varphi(x, y, z)$ by

$$\exists x', y', z'\ R(x', y', z') \wedge x \xrightarrow{\mathbf{P}-\mathbf{P}} x' \wedge y \xrightarrow{\mathbf{P}-\mathbf{P}} y' \wedge z \xrightarrow{\mathbf{P}-\mathbf{P}} z'.$$

This is not a pp-formula, but it is equivalent to one, by "pulling out" the existential quantifiers hidden in $\star \xrightarrow{\mathbf{P}-\mathbf{P}} \star$.

To see why the first item follows, note that $\mathbf{P} - \mathbf{P}$ has net length 0 and thus, in $\widehat{\mathbf{D}}_i$, we have $x' = x$, $y' = y$, and $z' = z$.

To see why the second item follows, consider the vertices

$$a_0, \dots, a_{2k-1}$$

given by Proposition 22. Suppose we label the vertices of $\widehat{\mathbf{D}}_{2k}$ by $0, 1, \dots, 2k-1$. Then the desired homomorphism is the one sending i to a_i. Letting $\oplus$ be addition modulo $2k$, that this is a homomorphism is equivalent to saying that

$$\varphi(a_i, a_i, a_{i\oplus 1}) \wedge \varphi(a_i, a_{i\oplus 1}, a_i) \wedge \varphi(a_{i\oplus 1}, a_i, a_i)$$

is true in $\widehat{\mathbf{G}}$ for all $i = 0, \dots, 2k-1$. Since φ is clearly symmetric, it is only necessary to prove that $\varphi(a_i, a_i, a_{i\oplus 1})$ holds. Recall that $\mathbf{L}_2 + \mathbf{P} - \mathbf{P}$ homomorphically maps into $\mathbf{G}$ such that its beginning is mapped to a_i and its end is mapped to $a_{i\oplus 1}$. In other words, there exists some vertex b such that an edge from a_i to b exists in $\mathbf{G}$, and $\mathbf{P} - \mathbf{P}$ maps into $\mathbf{G}$ such that its beginning is mapped to b and its end is mapped to $a_{i\oplus 1}$. As a consequence, $b \xrightarrow{\mathbf{P}-\mathbf{P}} a_{i\oplus 1}$ is true when interpreted in $\widehat{\mathbf{G}}$; by the symmetry of $P - P$, $a_{i\oplus 1} \xrightarrow{\mathbf{P}-\mathbf{P}} b$ is true as well. Thus, the witnesses to the truth of $\varphi(a_i, a_i, a_{i\oplus 1})$ are $x' = a_i, y' = a_i$, and $z' = b$. Indeed, $R(a_i, a_i, b)$ is true since there is an edge from a_i to b; $a_i \xrightarrow{\mathbf{P}-\mathbf{P}} a_i$ is true by Proposition 22; and $a_{i\oplus 1} \xrightarrow{\mathbf{P}-\mathbf{P}} b$ is true as shown above. □

Proof of Proposition 21. Applying Lemma 24, we find that there exists a pp-formula φ such that, if φ interpreted in $\widehat{\mathbf{G}}$ pp-defines $\mathbf{T}$, then $\widehat{\mathbf{D}}_{2k} \to \mathbf{T}$, and φ interpreted in $\mathbf{NAE} = \widehat{\mathbf{D}}_2$ pp-defines $\mathbf{NAE}$. Hence, $(\widehat{\mathbf{G}}, \mathbf{NAE})$ pp-defines $(\mathbf{T}, \mathbf{NAE})$. By Theorem 7, $\mathrm{PCSP}(\mathbf{T}, \mathbf{NAE})$ reduces to $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$. On the other hand, we have that $\mathrm{PCSP}(\widehat{\mathbf{D}}_{2k}, \mathbf{NAE})$ reduces to $\mathrm{PCSP}(\mathbf{T}, \mathbf{NAE})$, since $\widehat{\mathbf{D}}_{2k} \to \mathbf{T}$. Combining the two reductions, we obtain the required result. □

5 RELATED WORK AND FUTURE DIRECTIONS

The results obtained in the current work identified two features of the satisfiability problem 1-in-3 vs. NAE that can be regarded as the reason for its tractability: The problem is solvable in polynomial time because 1-in-3 corresponds to a balanced digraph, or because NAE corresponds to a small cycle. We completely classified the complexity of the extensions of the 1-in-3 vs. NAE problem obtained by breaking the promise either from the left or from the right, in the symmetric rainbow-free regime. It is noteworthy that the classifications for the two cases are in terms of structural properties of the underlying graphs having two different natures. Indeed, the key property guaranteeing tractability of templates of the form $(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{G}})$ is *local*, in that it corresponds to the presence of small directed cycles in $\mathbf{G}$. In contrast, the balancedness property, regulating the complexity of templates of the form $(\widehat{\mathbf{G}}, \mathbf{NAE})$, cannot be detected by looking at small subgraphs of $\widehat{\mathbf{G}}$, and is thus a *global* property. As mentioned in the Introduction, our results fit within the broader picture of the complexity investigation of PCSPs. In this sense, we see two natural directions for future analysis:

(i) breaking the promise from *both* sides simultaneously;
(ii) relaxing the symmetricity and rainbow-freeness assumptions.

The direction (i) corresponds to studying the complexity of problems $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ for arbitrary pairs of digraphs $\mathbf{G}$ and $\mathbf{H}$ such that $\mathbf{G} \to \mathbf{H}$. Our results imply a classification of problems of this sort in the bipartite regime.

Corollary 25. *Let $\mathbf{G} \to \mathbf{H}$ be two bipartite digraphs with nonempty edge sets. Then $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is tractable if $\mathbf{G}$ has no oriented cycles with nonzero net length and $\mathbf{H}$ has a directed cycle of length at most 3; otherwise, $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is* NP*-hard.*

Proof. The tractability result follows since when $\mathbf{G}$ has no oriented cycles with nonzero net length and $\mathbf{H}$ has a directed cycle of length at most 3, we have that $\widehat{\mathbf{G}} \to \mathbf{Z} \to \widehat{\mathbf{H}}$. For hardness, note that $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is at least as hard as $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{H}})$ and $\mathrm{PCSP}(\widehat{\mathbf{G}}, \mathbf{NAE})$. Hence, the hardness follows from Theorems 1 and 2. □

Observe that the tractability boundary in this case is a conjunction of a condition on $\mathbf{G}$ and a condition on $\mathbf{H}$, with these conditions being *independent*. Is this a coincidence, or does the independence of the properties drawing the tractability boundary for PCSPs hold in a more general regime?

All tractable cases of problems $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ that we are aware of are those that can sandwich $\mathbf{Z}$, $\widehat{\mathbf{D}}_1$, or $\widehat{\mathbf{D}}_3$ (excluding trivial cases when $\mathbf{G}$ has no edges).[9] Are all remaining problems in this class NP-hard?[10] We proved this to be true when $\mathbf{G}$ is balanced (and contains at least one edge), or when $\mathbf{G}$ is an arbitrary unbalanced bipartite digraph and $\mathbf{H} = \mathbf{D}_2$. It is known (see e.g. [32]) and easy to show that $\mathbf{G} \to \mathbf{D}_3$ if and only if all oriented cycles in G have net length divisible by 3. Thus, to answer the above question in the positive, it would be sufficient to prove NP-hardness for the following two cases: (a) when $\mathbf{G}$ is an oriented cycle of net length not divisible by 3 and $\mathbf{H}$ is a complete graph with at least three vertices, and (b) when $\mathbf{G}$ is an oriented cycle of nonzero net length divisible by 3 and $\mathbf{H}$ is $\mathbf{D}_3$-free.[11]

[9] Sandwiching $\widehat{\mathbf{D}}_1$ is the same as $\widehat{\mathbf{H}}$ having a loop, hence all of these examples are trivial. $\mathrm{CSP}(\widehat{\mathbf{D}}_3)$ is tractable since it is equivalent to solving mod-3 linear equations. For an example of a PCSP that sandwiches $\widehat{\mathbf{D}}_3$ (but not $\mathbf{Z}$) without having any structure in the template be homomorphically equivalent to $\widehat{\mathbf{D}}_3$, let $\mathbf{X}$ be the unique tournament on 4 vertices that has a cycle of length 4, and consider $\mathrm{PCSP}(\widehat{\mathbf{D}}_9, \mathbf{X})$.

[10] Note that the statement is true if $\mathbf{G}$ contains an undirected edge $\mathbf{D}_2$. Indeed, in this case, letting h be the size of $\mathbf{H}$, and $\mathbf{NAE}_h = ([h]; [h]^3 \setminus \{(1,1,1), \dots, (h,h,h)\})$, as $\mathbf{D}_1 \not\to \mathbf{H}$, we have the sandwich $\mathbf{NAE} = \widehat{\mathbf{D}}_2 \to \widehat{\mathbf{G}} \to \widehat{\mathbf{H}} \to \mathbf{NAE}_h$, and $\mathrm{PCSP}(\mathbf{NAE}, \mathbf{NAE}_h)$ is NP-hard by [26].

[11] To see why, note that when $\mathbf{G}$ has no edges $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is tractable; when $\mathbf{G}$ is balanced and has an edge then $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is tractable if and only if $\mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{H}})$ is; when all cycles of $\mathbf{G}$ have net length divisible by 3 and $\mathbf{D}_3 \to \widehat{\mathbf{H}}$ then $\mathrm{PCSP}(\widehat{\mathbf{G}}, \widehat{\mathbf{H}})$ is tractable by reduction to $\mathrm{CSP}(\widehat{\mathbf{D}}_3)$. The remaining cases are (a) when $\mathbf{G}$ is unbalanced yet all cycles have net length divisible by 3, and $\mathbf{H}$ is $\mathbf{D}_3$-free, which is hard by reducing

We also note that all our tractable cases can be solved by the Affine Integer Programming (AIP) relaxation of [12]. This provides more evidence for the conjecture, first formally stated in the work [46], namely that PCSP(**1-in-3**, **A**) is tractable if and only if it is solved by AIP.

We now discuss direction (ii). It was observed in [3, 17] that assuming the structure **A** in a problem PCSP(**1-in-3**, **A**) to be symmetric does not yield a loss of generality, as, if **A** is non-symmetric, replacing it with the maximal symmetric substructure does not alter the complexity of the problem. In [3], essentially the following question was posed: Are templates of the form (**1-in-3**, **A**) tractable precisely when **A** contains one of the three structures $\widehat{\mathbf{D}}_1, \widehat{\mathbf{D}}_2, \widehat{\mathbf{D}}_3$ as a substructure? When **A** is rainbow-free, the current work shows that this is indeed the case.

Given a digraph **G**, let $\widehat{\mathbf{G}}^+$ denote the ternary structure obtained from $\widehat{\mathbf{G}}$ by adding all rainbow tuples (i.e., tuples (x, y, z) with x, y, z all distinct) to the relation. Recall that $\mathbf{T}_i$ denotes the transitive tournament with i vertices. [3] classified the complexity of all problems PCSP(**1-in-3**, **A**) where **A** has a 3-element domain, with the exception of PCSP(**1-in-3**, $\widehat{\mathbf{T}}_3^+$).[12] Note that **1-in-3** is precisely $\widehat{\mathbf{T}}_2^+$. Within [3] it is conjectured that PCSP($\widehat{\mathbf{T}}_k^+, \widehat{\mathbf{T}}_\ell^+$) is NP-hard for $k \leq \ell$.[13] Versions of this problem for higher arities were shown to be hard in [45], and the techniques of [45, Theorem 26] show that to prove this conjecture it is necessary and sufficient to show that

$$\mathrm{PCSP}(\widehat{\mathbf{T}}_2^+, \widehat{\mathbf{T}}_k^+) = \mathrm{PCSP}(\mathbf{1\text{-}in\text{-}3}, \widehat{\mathbf{T}}_k^+)$$

is NP-hard for all $k \geq 2$. We view our hardness results as a partial step towards the resolution of this conjecture, in the rainbow-free regime: It follows from Theorem 1 that PCSP(**1-in-3**, $\widehat{\mathbf{T}}_k$) is NP-hard for $k \geq 2$; what the conjecture then requires is for this hardness to hold even if rainbow tuples are allowed.

Nakajima and Živný [46] classified all PCSPs which have the form PCSP(**1-in-3**, **A**) where **A** is *functional*, that is, the relation of **A** does not contain tuples $(x, y, z), (x, y, z')$ with $z \neq z'$. Thus, these results fail to classify PCSP(**1-in-3**, $\widehat{\mathbf{G}}$) whenever **G** has a vertex with out-degree at least 2. On the other hand, unlike this paper, some of the structures **A** for which [46] gave a classification do have rainbow tuples.

Comparing with the literature on PCSP(**1-in-3**, **A**), we find it interesting that the dual extension PCSP(**A**, **NAE**) has been significantly less investigated. We are aware of only two works in this line of work. Firstly, [17] studied PCSP(**A**, **NAE**) for Boolean (possibly non-symmetric) **A** obtained from **1-in-3** by adding extra tuples. Secondly, [28] established a classification of Boolean symmetric PCSPs — however, up to isomorphism, the only Boolean symmetric relational structures with one ternary relation that map to **NAE** are $(\{0, 1\}; \emptyset)$, **1-in-3**, and **NAE**. Thus, for the case PCSP($\widehat{\mathbf{G}}$, **NAE**) with $\widehat{\mathbf{G}}$ Boolean, the results of [28] only yield trivial results: The problem is tractable when **G** is empty or has a loop, or when $\widehat{\mathbf{G}}$ = **1-in-3**, and is otherwise NP-complete. Our results thus additionally cover all other digraphs **G** — i.e., the non-Boolean setting.

Finally, we observe that extending our results in both directions (i) and (ii) *simultaneously* would amount to classifying the complexity of all problems PCSP(**A**, **B**) with **A** and **B** having a single, ternary relation. Through a similar argument as in [12] (see also [27]), this is easily seen to be equivalent to a classification for *all* PCSPs. Such a wide classification appears to be out of reach for the current techniques.

ACKNOWLEDGMENT

We thank all anonymous reviewers for their comments and suggestions for changes.

REFERENCES

[1] Albert Atserias and Víctor Dalmau. 2022. Promise Constraint Satisfaction and Width. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. 1129–1153. https://doi.org/10.1137/1.9781611977073.48 arXiv:2107.05886

[2] Per Austrin, Venkatesan Guruswami, and Johan Håstad. 2017. (2+ϵ)-Sat Is NP-hard. *SIAM J. Comput.* 46, 5 (2017), 1554–1573. https://doi.org/10.1137/15M1006507 eccc:2013/159

[3] Libor Barto, Diego Battistelli, and Kevin M. Berg. 2021. Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case. In *Proc. 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21) (LIPIcs, Vol. 187)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 10:1–10:16. https://doi.org/10.4230/LIPIcs.STACS.2021.10 arXiv:2010.04623

[4] Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. 2021. Algebraic approach to promise constraint satisfaction. *J. ACM* 68, 4 (2021), 28:1–28:66. https://doi.org/10.1145/3457606 arXiv:1811.00970

[5] Libor Barto and Marcin Kozik. 2016. Robustly Solvable Constraint Satisfaction Problems. *SIAM J. Comput.* 45, 4 (2016), 1646–1669. https://doi.org/10.1137/130915479 arXiv:1512.01157

[6] Libor Barto and Marcin Kozik. 2022. Combinatorial Gap Theorem and Reductions between Promise CSPs. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. 1204–1220. https://doi.org/10.1137/1.9781611977073.50 arXiv:2107.09423

[7] Libor Barto, Andrei Krokhin, and Ross Willard. 2017. *Polymorphisms, and How to Use Them*. Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 1–44. https://doi.org/10.4230/DFU.Vol7.15301.1

[8] Libor Barto, Jakub Opršal, and Michael Pinsker. 2018. The wonderland of reflections. *Isr. J. Math* 223, 1 (Feb 2018), 363–398. https://doi.org/10.1007/s11856-017-1621-9 arXiv:1510.04521

[9] Manuel Bodirsky and Martin Grohe. 2008. Non-dichotomies in Constraint Satisfaction Complexity. In *Proc.35th International Colloquium on Automata, Languages and Programming (ICALP'08) (Lecture Notes in Computer Science, Vol. 5126)*. Springer, 184–196. https://doi.org/10.1007/978-3-540-70583-3_16

[10] Manuel Bodirsky, Michael Pinsker, and András Pongrácz. 2021. Projective clone homomorphisms. *The Journal of Symbolic Logic* 86, 1 (2021), 148–161. https://doi.org/10.1017/jsl.2019.23.

[11] Joshua Brakensiek and Venkatesan Guruswami. 2016. New Hardness Results for Graph and Hypergraph Colorings. In *Proc. 31st Conference on Computational Complexity (CCC'16) (LIPIcs, Vol. 50)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 14:1–14:27. https://doi.org/10.4230/LIPIcs.CCC.2016.14

[12] Joshua Brakensiek and Venkatesan Guruswami. 2021. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.* 50, 6 (2021), 1663–1700. https://doi.org/10.1137/19M128212X arXiv:1704.01937

[13] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2023. Conditional Dichotomy of Boolean Ordered Promise CSPs. *TheoretiCS* 2 (2023). arXiv:2102.11854

[14] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2023. SDPs and Robust Satisfiability of Promise CSP. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 609–622. https://doi.org/10.1145/3564246.3585180 arXiv:2211.08373

[15] Alex Brandts. 2022. *Promise Constraint Satisfaction Problems*. Ph. D. Dissertation. University of Oxford. https://ora.ouls.ox.ac.uk/objects/uuid:5efeff56-d5c8-42e8-9bd2-95243a1367ad

[16] Alex Brandts, Marcin Wrochna, and Stanislav Živný. 2021. The Complexity of Promise SAT on Non-Boolean Domains. *ACM Trans. Comput. Theory* 13, 4 (2021), 26:1–26:20. https://doi.org/10.1145/3470867 arXiv:1911.09065

[17] Alex Brandts and Stanislav Živný. 2022. Beyond PCSP(1-in-3,NAE). *Information and Computation* 289, Part A (2022), 104954. https://doi.org/10.1016/j.ic.2022.104954 arXiv:2104.12800

[18] Mark Braverman, Subhash Khot, Noam Lifshitz, and Dor Minzer. 2021. An Invariance Principle for the Multi-slice, with Applications. In *Proc. 62nd IEEE*

from case (a) above, and (b) when **G** has an oriented cycle of net length indivisible by 3, which is hard by a reduction from case (b) above.

[12] In [3], $\widehat{\mathbf{T}}_3$ and $\widehat{\mathbf{T}}_3^+$ are denoted by $\mathbf{T}_1$ and $\mathbf{T}_1^+$, respectively.

[13] In [3], $\widehat{\mathbf{T}}_k^+$ is denoted by $\mathbf{LO}_k$.

Annual Symposium on Foundations of Computer Science (FOCS'21). IEEE, 228–236. https://doi.org/10.1109/FOCS52979.2021.00030 arXiv:2110.10725

[19] Andrei Bulatov. 2017. A Dichotomy Theorem for Nonuniform CSPs. In *Proc. 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS'17)*. 319–330. https://doi.org/10.1109/FOCS.2017.37 arXiv:1703.03021

[20] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM J. Comput.* 34, 3 (2005), 720–742. https://doi.org/10.1137/S0097539700376676

[21] Lorenzo Ciardo and Stanislav Živný. 2023. Approximate Graph Colouring and Crystals. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. 2256–2267. https://doi.org/10.1137/1.9781611977554.ch86 arXiv:2210.08293

[22] Lorenzo Ciardo and Stanislav Živný. 2023. Approximate Graph Colouring and the Hollow Shadow. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 623–631. https://doi.org/10.1145/3564246.3585112 arXiv:2211.03168

[23] Lorenzo Ciardo and Stanislav Živný. 2023. CLAP: A New Algorithm for Promise CSPs. *SIAM J. Comput.* 52, 1 (2023), 1–37. https://doi.org/10.1137/22M1476435 arXiv:2107.05018

[24] Stephen A. Cook. 1971. The Complexity of Theorem-Proving Procedures. In *Proc. 3rd Annual ACM Symposium on Theory of Computing (STOC'71)*. ACM, 151–158. https://doi.org/10.1145/800157.805047

[25] Irit Dinur, Elchanan Mossel, and Oded Regev. 2009. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.* 39, 3 (2009), 843–873. https://doi.org/10.1137/07068062X

[26] Irit Dinur, Oded Regev, and Clifford Smyth. 2005. The Hardness of 3-Uniform Hypergraph Coloring. *Combinatorica* 25, 5 (Sept. 2005), 519–535. https://doi.org/10.1007/s00493-005-0032-4

[27] 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. https://doi.org/10.1137/S0097539794266766

[28] Miron Ficak, Marcin Kozik, Miroslav Olšák, and Szymon Stankiewicz. 2019. Dichotomy for Symmetric Boolean PCSPs. In *Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP'19) (LIPIcs, Vol. 132)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 57:1–57:12. https://doi.org/10.4230/LIPIcs.ICALP.2019.57 arXiv:1904.12424

[29] M. R. Garey and David S. Johnson. 1976. The Complexity of Near-Optimal Graph Coloring. *J. ACM* 23, 1 (1976), 43–49. https://doi.org/10.1145/321921.321926

[30] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman.

[31] Venkatesan Guruswami and Sai Sandeep. 2020. d-To-1 Hardness of Coloring 3-Colorable Graphs with $O(1)$ Colors. In *Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP'21) (LIPIcs, Vol. 198)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 62:1–62:12. https://doi.org/10.4230/LIPIcs.ICALP.2020.62

[32] Roland Häggkvist, Pavol Hell, Donald J. Miller, and Victor Neumann Lara. 1988. On multiplicative graphs and the product conjecture. *Combinatorica* 8, 1 (1988), 63–74. https://doi.org/10.1007/BF02122553

[33] Frank Harary and Leo Moser. 1966. The Theory of Round Robin Tournaments. *The American Mathematical Monthly* 73, 3 (1966), 231–246. https://doi.org/10.2307/2315334

[34] Yahli Hecht, Dor Minzer, and Muli Safra. 2023. NP-Hardness of Almost Coloring Almost 3-Colorable Graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA (LIPIcs, Vol. 275)*, Nicole Megow and Adam D. Smith (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 51:1–51:12. https://doi.org/10.4230/LIPICS.APPROX/RANDOM.2023.51

[35] Pavol Hell and Jaroslav Nešetřil. 1990. On the complexity of H-coloring. *J. Combin. Theory Ser. B* 48, 1 (1990), 92–110. https://doi.org/10.1016/0095-8956(90)90132-J

[36] Pavol Hell and Jaroslav Nešetřil. 2004. *Graphs and Homomorphisms*. Oxford University Press.

[37] Sangxia Huang. 2013. Improved Hardness of Approximating Chromatic Number. In *Proc. 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques and the 17th International Workshop on Randomization and Computation (APPROX-RANDOM'13)*. Springer, 233–243. https://doi.org/10.1007/978-3-642-40328-6_17 arXiv:1301.5216

[38] Peter G. Jeavons. 1998. On the Algebraic Structure of Combinatorial Problems. *Theor. Comput. Sci.* 200, 1-2 (1998), 185–204. https://doi.org/10.1016/S0304-3975(97)00230-2

[39] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. 1997. Closure Properties of Constraints. *J. ACM* 44, 4 (1997), 527–548. https://doi.org/10.1145/263867.263489

[40] Ravindran Kannan and Achim Bachem. 1979. Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix. *SIAM J. Comput.* 8, 4 (1979), 499–507. https://doi.org/10.1137/0208040

[41] Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proc. Complexity of Computer Computations*. 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9

[42] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. 2000. On the Hardness of Approximating the Chromatic Number. *Comb.* 20, 3 (2000), 393–415. https://doi.org/10.1007/s004930070013

[43] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. 2023. Topology and adjunction in promise constraint satisfaction. *SIAM J. Comput.* (2023). arXiv:2003.11351

[44] Benoît Larose and Pascal Tesson. 2009. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.* 410, 18 (2009), 1629–1647. https://doi.org/10.1016/J.TCS.2008.12.048

[45] Tamio-Vesa Nakajima and Stanislav Živný. 2022. Linearly Ordered Colourings of Hypergraphs. *ACM Trans. Comput. Theory* 13, 3–4 (2022). Issue Article no. 12. https://doi.org/10.1145/3570909 arXiv:2204.05628

[46] Tamio-Vesa Nakajima and Stanislav Živný. 2023. Boolean symmetric vs. functional PCSP dichotomy. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 1–12. https://doi.org/10.1109/LICS56636.2023.10175746

[47] Thomas Schaefer. 1978. The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on the Theory of Computing (STOC'78)*. 216–226. https://doi.org/10.1145/800133.804350

[48] Johan Thapper and Stanislav Živný. 2017. The power of Sherali–Adams relaxations for general-valued CSPs. *SIAM J. Comput.* 46, 4 (2017), 1241–1279. https://doi.org/10.1137/16M1079245 arXiv:1606.02577

[49] Xuding Zhu. 1992. A simple proof of the multiplicativity of directed cycles of prime power length. *Discrete applied mathematics* 36, 3 (1992), 313–316. https://doi.org/10.1016/0166-218X(92)90262-9

[50] Dmitriy Zhuk. 2020. A Proof of the CSP Dichotomy Conjecture. *J. ACM* 67, 5 (Aug. 2020), 30:1–30:78. https://doi.org/10.1145/3402029 arXiv:1704.01914