Solving promise equations over monoids and groups*

Alberto Larrauri
University of Oxford
alberto.larrauri@cs.ox.ac.uk

Stanislav Živný
University of Oxford
standa.zivny@cs.ox.ac.uk

21st February 2025

Abstract

We give a complete complexity classification for the problem of finding a solution to a given system of equations over a fixed finite monoid, given that a solution over a more restricted monoid exists. As a corollary, we obtain a complexity classification for the same problem over groups.

1 Introduction

Constraint satisfaction problems (CSPs) form a large class of fundamental computational problems studied in artificial intelligence, database theory, logic, graph theory, and computational complexity. Since CSPs (with infinite domains) capture, up to polynomial-time Turing reductions, *all* computational problems [11], some restrictions need to be imposed on CSPs in order to have a chance to obtain complexity classifications. One line of work, pioneered in the database theory [36], restricts the interactions of the constraints in the instance [30, 41].

Another line of work, pioneered in [26, 34], restricts the types of relations used in the instance; these CSPs are known as nonuniform CSPs, or as having a fixed template/constraint language. Such CSPs with infinite domains capture graph acyclicity, systems of linear equations over the rationals, and many other problems [10]. Already fixed-template CSPs with finite domains form a large class of fundamental problems, including graph colourings [32], variants of the Boolean satisfiability problem, and, more generally, systems of equations over different types of finite algebraic structures. Even then, the class of finite-domain CSPs avoided a complete complexity classification for two decades despite a sustained effort.

In 2017, Bulatov [20] and, independently, Zhuk [47] classified all finite-domain CSPs as either solvable in polynomial time or NP-hard, thus answering in the affirmative the Feder-Vardi dichotomy conjecture [26]. In the effort to answer the Feder-Vardi conjecture, complexity dichotomies were established for restricted fragments of CSPs, e.g., conservative CSPs [19], and equations over finite algebraic structures such as groups [29] and monoids [35]. In particular, while systems of equations[1] over Abelian groups are solvable in polynomial time, they are NP-hard over non-Abelian groups [29].

*An extended abstract of this work appeared in the Proceedings of the ICALP 2024 [40]. This work was supported by UKRI EP/X024431/1. All data is provided in full in the results section of this paper.

[1]Some papers use the term a *linear* equation.

One of the recent research directions in constraint satisfaction that has attracted a lot of attention is the area of *promise* CSPs (PCSPs) [3,5,13]. The idea is that each constraint has two versions, a strong version and a weak version. Given an instance, one is promised that a solution satisfying all strict constraints exists and the goal is to find a solution satisfying all weak constraints, which may be an easier task. The prototypical example is the approximate graph colouring problem [28]: Given a 3-colourable graph, can one find a 6-colouring? The complexity of this problem is open (but believed to be NP-hard). Despite a flurry of papers on PCSPs, e.g., [1,2,4,6,14,15,17,21,22,24,27,43,44], the PCSP complexity landscape is widely open and unexplored. It is not even clear whether a dichotomy should be expected. Even the case of Boolean PCSPs remain open, the state-of-the-art being a dichotomy for Boolean *symmetric* PCSPs [27]. This should be compared with Boolean (non-promise) CSPs, which were classified by Schaefer in 1978 [45]. Schaefer's tractable cases include the classic and well-known examples of CSPs: equations and graph colouring. Both have been studied on non-Boolean domains and their complexity is well understood. However, the complexity of the promise variant of these fundamental problems is open. The first problem, graph colouring, leads to the already mentioned approximate graph colouring problem, which is a notorious open problem, despite recent progress [5,38]. In this paper, we look at the second problem, and study PCSPs capturing systems of equations.

Contributions The precise statements of all our main results are presented in Section 3.

As our most important contribution, in Section 5 we establish a complexity dichotomy for PCSPs capturing *promise* systems of equations over finite monoids, and over finite groups as a special case. Perhaps unsurprisingly, the tractability boundary is linked to the notion of Abelianness, just like in the non-promise setting, but the result is non-trivial and requires some care. Our main tool for studying the computational complexity of PCSPs is the so-called "algebraic approach", relying crucially on the notion of a *polymorphism*. Polymorphisms can be seen as high-dimensional symmetries of a PCSP template and capture the complexity of the underlying computational problem [5,13]. Polymorphisms of a PCSP template form a *minion* [5]; that is, a family of functions that is closed under permuting arguments, identifying arguments, and adding dummy arguments. As we shall see later, it is useful to study more abstract minions, not only families of functions, cf. [5,16]. Following the approach from [5], hardness of a PCSP is established by showing that the associated polymorphism minion is, in some sense, limited. Conversely, if this minion is rich enough then the PCSP can be shown to be solvable via some efficient algorithm [5,15,16,22].

To prove our main result, we study a class of minions that arise naturally from monoids, which we call monoidal minions. In Section 4 we show a complexity dichotomy for PCSPs whose polymorphism minions are homomorphically equivalent to some monoidal minion. This is our second contribution, which may be of independent interest. In particular, the concept of monoidal minions captures studied minions, cf. Remark 1 in Section 3.

All our tractability results use solvability by the BLP + AIP algorithm [16]. In fact, tractable PCSPs corresponding to promise systems of equations over monoids are finitely tractable in the sense of [1,13]. In the special case of promise systems of equations over groups, the affine integer programming (AIP) algorithm [5,13] suffices, rather than BLP + AIP. However, AIP is provably not enough to solve promise equations over general monoids.

As our final contribution, in Section 6 we show that our dichotomy for systems of equations over monoids cannot be easily extended to semigroups, as this would imply a dichotomy for all PCSPs. We do so by showing that every PCSP is polynomial-time equivalent to a PCSP capturing systems of equations over semigroups, a phenomenon observed for CSPs in [35].

Related work PCSPs are a qualitative approximation of CSPs; the goal is still to satisfy all constraints, but in a weaker form. A recent related line of work includes the series [7–9]. A traditional approach to approximation is quantitative: maximising the number of satisfied constraints. Regarding approximation of equations, Håstad showed that, for any Abelian group G and any $\varepsilon > 0$, it is NP-hard to find a solution satisfying $1/|G| + \varepsilon$ constraints [31] even if $1 - \varepsilon$ constraints can be satisfied. Hence, the random assignment, which satisfies $1/|G|$ constraints, is optimal! Håstad's result has been extended to non-Abelian groups in [7, 25]. Systems of equations have been studied, e.g., over semigroups in [46], over monoids and semigroups in [35], and over arbitrary finite algebras in [12, 37, 39, 42].

The work of Nakajima and Živný on symmetric functional PCSPs [44] is somewhat related to PCSPs and equations but is incomparable to the work in the present article. Since we do not need it, we do not use the language of category theory but we remark that minions and other concepts can be presented in a category-theoretical way, cf. [24, 44].

2 Preliminaries

We denote by $[k]$ the set $\{1, 2, \ldots, k\}$. We write id_X for the identity map on a set X. We use the lowercase boldface font for tuples; e.g., we write $\boldsymbol{b}$ for a tuple $(b_1, \ldots, b_n)$. We say that a function f *extends* another function g if $\mathrm{dom}(g) \subseteq \mathrm{dom}(f)$, and $f|_{\mathrm{dom}(g)} = g$.

Algebraic structures A *semigroup* S is a set equipped with an associative binary operation, for which we use multiplicative notation. Two elements $a, b \in S$ commute if $ab = ba$. An element a is *idempotent* if $aa = a$. An *Abelian* semigroup is a semigroup in which every two elements commute. A *semigroup homomorphism* from a semigroup S_1 to a semigroup S_2 is a map $\varphi : S_1 \to S_2$ satisfying $\varphi(s \cdot_{S_1} t) = \varphi(s) \cdot_{S_2} \varphi(t)$.[2] Given two elements $s, t \in S$ we write $s \sqsubseteq t$ if $s = t$ or there is an element $r \in S$ satisfying $tr = s$. Note that $\sqsubseteq$ constitutes a preorder over any semigroup, i.e., $\sqsubseteq$ is reflexive and transitive. We define the equivalence relation $\sim$ by $s \sim t$ whenever $s \sqsubseteq t$ and $t \sqsubseteq s$.

A *monoid* is a semigroup containing an identity element for its binary operation, denoted by e. A *monoid homomorphism* from a monoid M_1 to a monoid M_2 is a map $\varphi : M_1 \to M_2$ satisfying $\varphi(x \cdot_{M_1} y) = \varphi(x) \cdot_{M_2} \varphi(y)$ and $\varphi(e_{M_1}) = e_{M_2}$. We say that φ is *Abelian* if its image $\mathrm{Im}(\varphi)$ is an Abelian monoid.

A *group* is a monoid in which each element has an inverse. A *group homomorphism* from a group G_1 to a group G_2 is a map $\varphi : G_1 \to G_2$ satisfying $\varphi(x \cdot_{G_1} y) = \varphi(x) \cdot_{G_2} \varphi(y)$ (which implies that also the inverses and the identity element are preserved).

Given a semigroup S, a subset $G \subseteq S$ is called a *subgroup* if G equipped with S's binary operation is a group, meaning that there is a distinguished element $e_G \in G$ satisfying that (1) $e_G \cdot_M g = g \cdot_M e_G = g$ for each $g \in G$, and (2) for each element $g \in G$ there exists $h \in G$ satisfying $g \cdot_M h = h \cdot_M g = e_G$. We say that S is a *union of subgroups* if every element $s \in S$ belongs to a subgroup of S.

[2] I.e., the multiplication on the LHS is in S_1, whereas the multiplication on the RHS is in S_2.

We call an element s of a semigroup S *regular*[3] if $s^2t = s$ for some t in S.[4] Intuitively, t acts as some type of inverse of s. It is known that s belongs to a subgroup of S if and only if s is regular [33, Theorem 2.2.5]. We will make use of the following equivalent characterisations of regularity.

Lemma 1. *Let S be a finite semigroup and $s \in S$. Then the following are equivalent:*

(1) s is regular,

(2) $s^k = s$ for some $k > 1$,

(3) s belongs to a subgroup of S,

(4) $s \sqsubseteq s^2$.

Proof. **(3) $\implies$ (2):** For any finite group G there exists number $k > 1$ such that $g^k = g$ for all $g \in G$.

(2) $\implies$ (3): Consider the set $G = \{s^\ell \mid 1 \leqslant \ell < k\}$. We claim that G is a group whose identity is s^{k-1}. By (2), s^{k-1} acts as a multiplicative identity in G. Moreover, given any $1 \leqslant \ell < k-1$, the inverse of s^ℓ in G is simply $s^{k-1-\ell}$.

(1) $\implies$ (4): By the definition of regularity, there is some element t such that $s^2t = s$, meaning that $s \sqsubseteq s^2$.

(2) $\implies$ (1): If $k = 2$, let $t = s$. Otherwise, if $k > 2$, let $t = s^{k-2}$. Then we have $s^2t = s$.

(4) $\implies$ (2): By assumption, $s^2t = s$ for some $t \in S$. Note that this implies that $s^{k+1}t^k = s$ for all $k \geqslant 1$. As S is finite, there must be numbers $k > \ell > 1$ satisfying that $s^k = s^\ell$. Then, the following chain of identities holds

$$s = s^k t^{k-1} = s^\ell t^{k-1} = s^\ell t^{\ell-1} t^{k-\ell} = st^{k-\ell}.$$

In particular, this means that $s^{k'} = s^{k'} t^{k-\ell}$ for any $k' \geqslant 1$. Finally, it also holds that $s = s^{k-\ell+1} t^{k-\ell}$, which together with the last equality yields that $s = s^{k-\ell+1}$. Since $\ell < k$, we have $k - \ell + 1 > 1$, thus establishing (2).

□

We use the standard product (and also the power) of a semigroup (monoid, group), where the operation is defined component-wise. We use the symbol $\preceq$ for a substructure; e.g., if S is a semigroup then we write $T \preceq S$ to indicate that T is a subsemigroup of S (and similarly for monoids and groups).

Unless stated explicitly otherwise, all semigroups, monoids, and groups in this paper are finite.

---

[3] In the extended abstract of this work [40], we required that $s^2t = s$ and $st = ts$ for some $t \in S$. For a finite semigroup S, this is equivalent to requiring that $s^2t = s$ for some $t \in S$ as if this second condition holds, then $s^k = s$ for some $k > 1$ by Lemma 1 (2), implying the existence of t that commutes with s and satisfies $s^2t = s$.

[4] The usual definition of a regular element in a semigroup, which is weaker, requires that $sts = s$ for some t [33]. What we call regular is often called completely regular.

Relational structures A *relational signature* σ consists of a finite set of relation symbols R, each with a finite arity $\mathrm{ar}(R) \in \mathbb{N}$. A *relational structure* $\boldsymbol{A}$ over the signature σ, or a σ-structure, consists of a finite set A and a relation $R^{\boldsymbol{A}} \subseteq A^k$ of arity $k = \mathrm{ar}(R)$ for every $R \in \sigma$. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two σ-structures. A map $h : A \to B$ is called a *homomorphism* from $\boldsymbol{A}$ to $\boldsymbol{B}$ if h preserves all relations in $\boldsymbol{A}$; i.e., if, for every $R \in \sigma$, $h(\mathbf{x}) \in R^{\boldsymbol{B}}$ whenever $\mathbf{x} \in R^{\boldsymbol{A}}$, where h is applied component-wise. We denote the existence of a homomorphism from $\boldsymbol{A}$ to $\boldsymbol{B}$ by writing $\boldsymbol{A} \to \boldsymbol{B}$. A *template* is a pair $(\boldsymbol{A}, \boldsymbol{B})$ of relational structures such that $\boldsymbol{A} \to \boldsymbol{B}$.

A k-ary *polymorphism* of a template $(\boldsymbol{A}, \boldsymbol{B})$ over signature σ is a map $p : A^k \to B$ that preserves all relations $R^{\boldsymbol{A}}$ from $\boldsymbol{A}$ in the following sense: For any $\mathrm{ar}(R) \times k$ matrix whose columns belong to $R^{\boldsymbol{A}}$, applying p row-wise results in a tuple that belongs to $R^{\boldsymbol{B}}$. We denote by $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ the set of all polymorphisms of $(\boldsymbol{A}, \boldsymbol{B})$.[5]

The i-th coordinate of a map $p : A^k \to A$ is called *essential* if there exist $a_1, \ldots, a_k \in A$ and $a_i' \in A$ such that $p(a_1, \ldots, a_k) \neq p(a_1, \ldots, a_{i-1}, a_i', a_{i+1}, \ldots, a_k)$. A coordinate that is not essential is called *inessential*. A map $p : A^k \to A$ is called *idempotent* if $p(x, \ldots, x) = x$.

Minions A *minion* $\mathscr{M}$ is a collection of sets $\mathscr{M}(n)$, one for each positive number n, such that, for each map $\pi : n \to m$, there is a map $\pi^{\mathscr{M}} : \mathscr{M}(n) \to \mathscr{M}(m)$ satisfying (1) $\mathrm{id}_{[n]}^{\mathscr{M}} = \mathrm{id}_{\mathscr{M}(n)}$ for every $n \geqslant 10$, and (2) $\pi^{\mathscr{M}} \circ \tau^{\mathscr{M}} = (\pi \circ \tau)^{\mathscr{M}}$ for every pair of suitable maps π, τ. When the minion is clear from the context, we write $p^{(\pi)}$ for $\pi^{\mathscr{M}}(p)$. Elements $p \in \mathscr{M}(n)$ are called n-ary or as having *arity* n. Whenever $p^{(\pi)} = q$ we say that q is a *minor* of p. A *minion homomorphism* $\xi : \mathscr{M} \to \mathscr{N}$ is a collection of maps $\xi_n : \mathscr{M}(n) \to \mathscr{N}(n)$ for each $n \geqslant 1$ that preserve minor operations; that is, $\xi_m(p^{(\pi)}) = (\xi_n, (p))^{(\pi)}$ for every minor $p^{(\pi)}$, where $\pi : [n] \to [m]$.

Given a template $(\boldsymbol{A}, \boldsymbol{B})$, its set of polymorphisms $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ can be equipped with a minion structure in a natural way: The n-ary elements of $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ are just n-ary polymorphisms $p : A^n \to B$. Additionally, given an n-ary polymorphism p and a map $\pi : [n] \to [m]$, the minor $p^{(\pi)}$ is the polymorphism $q : A^m \to B$ given by $(a_1, \ldots, a_m) \mapsto p(b_1, \ldots, b_n)$, where $b_i = a_{\pi(i)}$ for each $i \in [n]$.

Given a minion $\mathscr{M}$, we define two special types of elements. An element $p \in \mathscr{M}(2k+1)$ is called *alternating* if $p^{(\pi)} = p$ for any permutation $\pi : [2m+1] \to [2m+1]$ that preserves parity, and $p^{(\pi_1)} = p^{(\pi_2)}$, where for each $i = 1, 2$ the map π_i is given by $1 \mapsto i$, $2 \mapsto i$ and $j \mapsto j$ for all $j > 2$. An element $p \in \mathscr{M}(2k+1)$ is called 2-*block-symmetric* if the set $[2k+1]$ can be partitioned into two blocks of size $k+1$ and k in such a way that $p^{(\pi)} = p$ for any map $\pi : [2m+1] \to [2m+1]$ that preserves each block

Constraint satisfaction Let $(\boldsymbol{A}, \boldsymbol{B})$ be a template with common signature σ. The *promise constraint satisfaction problem* (PCSP) with template $(\boldsymbol{A}, \boldsymbol{B})$ is the following computational problem, denoted by $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$. Given a σ-structure $\boldsymbol{X}$, output YES if $\boldsymbol{X} \to \boldsymbol{A}$ and output NO if $\boldsymbol{X} \not\to \boldsymbol{B}$. This is the decision version. In the search version, one is given a σ-structure $\boldsymbol{X}$ with the promise that $\boldsymbol{X} \to \boldsymbol{A}$; the goal is to find a homomorphism from $\boldsymbol{X}$ to $\boldsymbol{B}$ (which necessarily exists, as $\boldsymbol{X} \to \boldsymbol{A}$ and $\boldsymbol{A} \to \boldsymbol{B}$, and homomorphisms compose). It is known that the decision version polynomial-time reduces to the search version (but it is not known whether the two variants are polynomial-time equivalent) [5]. In our results, the positive (tractability) results are for the search version, whereas the hardness (intractability) results are for the decision version. We denote by $\mathrm{CSP}(\boldsymbol{A})$ the problem $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{A})$; this is the

[5] Equivalently, p is a polymorphism of $(\boldsymbol{A}, \boldsymbol{B})$ if p is a homomorphism from the k-th power of $\boldsymbol{A}$ to $\boldsymbol{B}$.

standard (non-promise) constraint satisfaction problem (CSP). For CSPs, the decision version and the search version are polynomial-time equivalent [18].

We need two existing algorithms for PCSPs, namely the AIP algorithm [5] and the strictly more powerful BLP + AIP algorithm [16]. Their power is captured by the following results.

Theorem 1 ([5]). *Let* $(\boldsymbol{A}, \boldsymbol{B})$ *be a template. Then* $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ *is solved by* AIP *if and only if* $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ *contains alternating maps of all odd arities.*

Theorem 2 ([16]). *Let* $(\boldsymbol{A}, \boldsymbol{B})$ *be a template. Then* $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ *is solved by* BLP + AIP *if and only if* $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ *contains* 2*-block-symmetric maps of all odd arities.*

3 Overview of Results

Promise equations over monoids and groups Our first and main result is a dichotomy theorem for solving promise equations over finite monoids and thus also, as a special case, over finite groups. We first define equations in the standard, non-promise setting as it is useful for mentioning previous work and for our own proofs.

An *equation* over a semigroup S is an expression of the form $x_1 \dots x_n = y_1 \dots y_m$, where each x_i, y_i is either a variable or some element from S, referred to as a *constant*. A *system of equations* over S is just a set of equations. A solution to such a system is an assignment of elements of S to the variables of the system that makes all equations hold. Equations and systems of equations are defined similarly for monoids and groups. The only difference is that for groups we allow "inverted variables" x^{-1} in the equations, which are interpreted as inverses of the elements assigned to x.

In the context of CSPs, it is common to consider only restricted "types" of equations that can then express all other equations. The following definition captures systems of equations where each equation is either of the form $x_1 x_2 = x_3$, for three variables, or $x = c$, fixing a variable to a constant. It is well known that restricting to systems of equations of this kind is without loss of generality, cf. Appendix A.

Definition 1. Let S be a semigroup and $T \leq S$ a subsemigroup. The relational structure $\mathrm{Eqn}(S, T)$ has universe S, and the following relations:

- A ternary relation $R_\times = \{(s_1, s_2, s_3) \in S^3 \mid s_1 s_2 = s_3\}$, and
- a singleton unary relation $R_t = \{t\}$ for each $t \in T$.

This template captures systems of equations of the kind described above when we allow only constants in a subsemigroup T of the ambient semigroup S. Similarly, we define the templates $\mathrm{Eqn}(M, N)$, $\mathrm{Eqn}(G, H)$ in the same way when M is a monoid and $N \leq M$ a submonoid, and when G is a group and $H \leq G$ is a subgroup. Observe that the definition of subgroup is more restrictive than the one of submonoid and this in turn is more restrictive than the notion of subsemigroup. We slightly abuse the notation and write $\mathrm{Eqn}(S, T)$ for $\mathrm{CSP}(\mathrm{Eqn}(S, T))$.

Previous works focused on problems $\mathrm{Eqn}(G) = \mathrm{Eqn}(G, G)$ and $\mathrm{Eqn}(M) = \mathrm{Eqn}(M, M)$. Given a group G, it is known that $\mathrm{Eqn}(G)$ is solvable in polynomial time (by AIP) if G is Abelian, and NP-hard otherwise [29]. Similarly, when M is a monoid, $\mathrm{Eqn}(M)$ is solvable in polynomial time if M is Abelian and it is the union of its subgroups, and NP-hard otherwise [35]. These results were shown before the Dichotomy Theorem for CSPs was proved [20, 47]. The original proofs relied on ad-hoc reductions and various notions from the theory of groups and the theory of monoids. For the sake of completeness, we present simplified proofs of those previous results as corollaries of the Dichotomy Theorem in Appendix B.

We now define promise equations.

Definition 2. Let S_1, S_2 be semigroups, and let φ be a semigroup homomorphism with $\mathrm{dom}(\varphi) \leq S_1$ and $\mathrm{Im}(\varphi) \leq S_2$. The *promise system of equations over semigroups* problem $\mathrm{PEqn}(S_1, S_2, \varphi)$ is the $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$, where $A = S_1$, $B = S_2$, and the relations are defined as follows:

- A ternary relation $R_\times^{\boldsymbol{A}} = \{(s_1, s_2, s_3) \in S_1^3 \mid s_1 s_2 = s_3\}$, and $R_\times^{\boldsymbol{B}} = \{(s_1, s_2, s_3) \in S_2^3 \mid s_1 s_2 = s_3\}$.
- For each $t \in \mathrm{dom}(\varphi)$, a unary relation given by $R_t^{\boldsymbol{A}} = \{t\}$, and $R_t^{\boldsymbol{B}} = \{\varphi(t)\}$.

For this template to be well defined there should be a homomorphism from $\boldsymbol{A}$ to $\boldsymbol{B}$, which is equivalent to the existence of a semigroup homomorphism $\psi : S_1 \to S_2$ that extends φ.

Analogously, we also define the *promise system of equations over monoids* problem and the *promise system of equations over groups* problem by replacing semigroup-related notions with monoid-related notions and group-related notions respectively. Observe that the problem $\mathrm{Eqn}(S, T)$ described before corresponds precisely to $\mathrm{PEqn}(S, S, \mathrm{id}_S)$.

We can now state our main result.

Theorem 3 (Main)**.** *Let M_1, M_2 be monoids and φ a monoid homomorphism with $\mathrm{dom}(\varphi) \leq M_1, \mathrm{Im}(\varphi) \leq M_2$. Then $\mathrm{PEqn}(M_1, M_2, \varphi)$ is solvable in polynomial time by $\mathrm{BLP} + \mathrm{AIP}$ if and only if there is an Abelian homomorphism $\psi : M_1 \to M_2$ extending φ and $\mathrm{Im}(\psi)$ is a union of subgroups. If no such homomorphism ψ exists, then $\mathrm{PEqn}(M_1, M_2, \varphi)$ is NP-hard.*

For the special case of groups, we get a simpler tractability criterion and a simpler algorithm.

Corollary 1. *Let G_1, G_2 be groups and φ a group homomorphism with $\mathrm{dom}(\varphi) \leq G_1, \mathrm{Im}(\varphi) \leq G_2$. Then $\mathrm{PEqn}(G_1, G_2, \varphi)$ is solvable in polynomial time by AIP if and only if there is an Abelian homomorphism $\psi : G_1 \to G_2$ extending φ. If no such homomorphism ψ exists, then $\mathrm{PEqn}(G_1, G_2, \varphi)$ is NP-hard.*

As easy corollaries, Theorem 3 applies in the special case of non-promise setting.

Corollary 2. *Given two monoids $N \leq M$, $\mathrm{Eqn}(M, N)$ is solvable in polynomial time by $\mathrm{BLP} + \mathrm{AIP}$ if and only if there is an Abelian endomorphism of M extending id_N whose image is a union of subgroups. If no such endomorphism exists, then $\mathrm{Eqn}(M, N)$ is NP-hard.*

Corollary 3. *Given two groups $H \leq G$, $\mathrm{Eqn}(G, H)$ is solvable in polynomial time by AIP if and only if there is an Abelian endomorphism of G that extends id_H. If no such endomorphism exists, then $\mathrm{Eqn}(G, H)$ is NP-hard.*

Example 1. Let G be the dihedral group on four elements, and H be the symmetric group on four elements. Observe that G can be seen as a subgroup of H in a natural way: H consists of all permutations on four elements, while G contains only those that are symmetries of the square. The group G is generated by the right 90-degree rotation r and an arbitrary reflection f that leaves no element fixed. We consider two group homomorphisms φ_1, φ_2 with $\mathrm{dom}(\varphi_i) \leq G$ and $\mathrm{Im}(\varphi_i) \leq H$. The domain of both homomorphisms is the subgroup $\{e, r, r^2, r^3\} \leq G$. Then, φ_1 is given by $r \mapsto r^2$, and φ_2 is given by $r \mapsto r$. The following hold:

- $\mathrm{PEqn}(G, H, \varphi_1)$ is tractable, and solvable by AIP. However both $\mathrm{Eqn}(G, \mathrm{dom}(\varphi_1))$ and $\mathrm{Eqn}(H, \mathrm{Im}(\varphi_1))$ are NP-hard.
- $\mathrm{PEqn}(G, H, \varphi_2)$ is NP-hard.

To see the first item, observe that the group homomorphism $\psi : G \to H$ given by $r \mapsto r^2$ and $f \mapsto f$ is Abelian (its image is isomorphic to the direct product $\mathbb{Z}_2 \times \mathbb{Z}_2$) and extends φ_1. Hardness of $\mathrm{Eqn}(G, \mathrm{dom}(\varphi_1))$ is a consequence of the fact that the commutator subgroup of G is $\{e, r^2\}$, so $r^2 \in \mathrm{dom}(\varphi_1)$ is included in the kernel of any Abelian endomorphism of G. Similarly, hardness of $\mathrm{Eqn}(H, \mathrm{Im}(\varphi_1))$ follows from the fact that the commutator subgroup of H is the alternating group on four elements, and has $\mathrm{Im}(\varphi_1)$ as a subgroup.

The second item can be proved by observing that the only normal subgroup of G that does not intersect $\mathrm{dom}(\varphi_2)$ is the trivial subgroup, so any homomorphism $\psi : G \to H$ that extends φ_2 needs to be injective, and thus non-Abelian.

We say that $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is *finitely tractable* if there is $\boldsymbol{C}$ such that $\boldsymbol{A} \to \boldsymbol{C} \to \boldsymbol{B}$ and $\mathrm{CSP}(\boldsymbol{C})$ is solvable in polynomial time. The tractable cases in Theorem 3 are in fact finitely tractable, as the next result shows.

Lemma 2. *Assume that* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is in the positive part of Theorem 3; i.e., there is an Abelian homomorphism* $\psi : M_1 \to M_2$ *extending* φ *and* $\mathrm{Im}(\psi)$ *is a union of subgroups. Then,* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is finitely tractable.*

Proof. By Theorem 3 there must be some Abelian homomorphism $\psi : M_1 \to M_2$ extending φ and $\mathrm{Im}(\psi)$ is a union of subgroups. Let $M \leq M_2$ be the submonoid $\mathrm{Im}(\psi)$. By assumption M is Abelian and a union of subgroups. Let $N \leq M$ be the submonoid $\mathrm{Im}(\varphi)$. We claim that $\mathrm{Eqn}(M, N)$ is solvable in polynomial time. Indeed, consider the map id_M. This map is an Abelian endomorphism of M, whose image is a union of subgroups. Moreover, id_M extends id_N. So, by Theorem 3, $\mathrm{Eqn}(M, N)$ is solvable in polynomial time by BLP + AIP.

The idea now is that $\mathrm{Eqn}(M, N)$ can be "sandwiched" by the template $(\boldsymbol{A}, \boldsymbol{B})$ (defined in Definition 2) of $\mathrm{PEqn}(M_1, M_2, \varphi)$. To make this formal, we need to produce a template $\boldsymbol{C}$ in the same signature as $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $\mathrm{CSP}(\boldsymbol{C})$ is $\mathrm{Eqn}(M, N)$ up to relabeling some relations. The set C equals M. The relation $R_\times^{\boldsymbol{C}}$ consists of the triples $(s_1, s_2, s_3) \in M^3$ such that $s_1 s_2 = s_3$. Finally, for each $s \in \mathrm{dom}(\varphi)$, we define $R_s^{\boldsymbol{C}} = \{\varphi(s)\}$. By construction, it holds that $\boldsymbol{A} \mapsto \boldsymbol{C} \mapsto \boldsymbol{B}$: the map ψ is a homomorphism from $\boldsymbol{A}$ to $\boldsymbol{C}$, and the inclusion map is a homomorphism from $\boldsymbol{C}$ to $\boldsymbol{B}$. On the other hand, $\mathrm{CSP}(\boldsymbol{C})$ is easily seen to be equivalent to $\mathrm{Eqn}(\boldsymbol{C})$. Indeed, we can obtain $\mathrm{Eqn}(\boldsymbol{C})$ from $\boldsymbol{C}$ by relabeling each relation R_s to $R_{\varphi(s)}$ and removing duplicate relations, which does not change the complexity of the related CSP. □

The power of BLP + AIP is necessary in Theorem 3 in the sense that AIP does not suffice for all monoids, even for (non-promise) CSPs, unlike in the case of groups. Indeed, adding a fresh element to a group that serves as the monoid identity fools AIP.

Lemma 3. *Let G be an arbitrary Abelian group. Let M be the monoid resulting from adding to G a fresh element e that serves as the monoid identity. Then* $\mathrm{Eqn}(M, M)$ *is solvable by* BLP + AIP *but not by* AIP.

Proof. The fact that BLP + AIP solves the $\mathrm{Eqn}(M, M)$ follows by Theorem 3 from the fact that M is Abelian and a union of subgroups. To rule out AIP, we show that $\mathrm{Eqn}(M, M)$ has no alternating polymorphisms; this suffices by Theorem 1. We begin with the following observation. Let $p : M^n \to M$ be a polymorphism whose i-th coordinate is inessential. Consider the homomorphism $\tau_{p,i}$ that sends each element $s \in M$ to $p(e, \dots, s, \dots, e)$, where all arguments are equal to e except for the i-th one, which is equal to s. Then it must be that $\tau_{p,i}$ is constant and equal to e. Now suppose that p is a $(2n+1)$-ary alternating polymorphism. Then define $q(x_1, \dots, x_{2n}) = p(x_1, x_1, x_2, x_3, \dots, x_{2n-1})$. As p is alternating, the first coordinate is inessential in q. By our previous observation, $\tau_{q,1}$ is constant and equal to e. By definition,

$$\tau_{q,1}(s) = q(s, e, \dots, e) = p(s, s, e, \dots, e) = p(s, e, \dots, e)p(e, s, e, \dots, e) = \tau_{p,1}(s)\tau_{p,2}(s).$$

Hence $\tau_{p,1}(s)\tau_{p,2}(s) = e$ for all $s \in M$. The only way that the product of two elements equals e in M is that both elements are equal to e. Thus, both $\tau_{p,1}$ and $\tau_{p,2}$ are constant and equal to e. This means that the first and the second coordinate are inessential in p. However, as p is alternating, p is preserved under parity preserving permutations of its arguments, so the fact that its first and second coordinates are inessential means that in fact all its coordinates are inessential. However, if all coordinates of p are inessential, then p is constant, but this contradicts the fact that p must be idempotent, as singleton unary relations are in $\mathrm{Eqn}(M, M)$ and thus preserved by p. $\square$

Promise equations over semigroups As our next result, we prove that every PCSP is polynomial-time equivalent to a problem of the form $\mathrm{PEqn}(S_1, S_2, \varphi)$ over some semigroups S_1, S_2. Hence, extending our classification of promise equations beyond monoids is difficult in the sense that understanding the computational complexity of promise equations over semigroups is as hard as classifying all PCSPs. This result is analogous to the one known in the non-promise setting obtained in [35], whose proof we closely follow. One difficulty in lifting the result from [35] is the lack of constants in the promise setting. The details can be found in Section 6.

Theorem 4. *Let $(\boldsymbol{A}, \boldsymbol{B})$ be a template. Then there are semigroups S_1, S_2 and a semigroup homomorphism φ with $\mathrm{dom}(\varphi) \le S_1$ and $\mathrm{Im}(\varphi) \le S_2$ such that $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is polynomial-time equivalent to $\mathrm{PEqn}(S_1, S_2, \varphi)$.*

Monoidal minions As our third result, we investigate minions based on monoids. For PCSPs whose polymorphism minions are homomorphically equivalent to such minions, we establish a dichotomy. This is a building block in the proof of our main result, but may be interesting in its own right. In this direction, we show that for each monoidal minion $\mathscr{M}$, there are PCSP templates whose polymorphism minions are isomorphic to $\mathscr{M}$. For a finite set $[n]$, a tuple $(a_i)_{i\in[n]} \in M^n$ is called commutative if each pair of its elements commute.

Definition 3. Given an element $a \in M$ of a monoid M, the *monoidal minion* $\mathscr{M}_{M,a}$ is the one where for each $n \in \mathbb{N}$ the elements $\boldsymbol{b} \in \mathscr{M}_{M,a}(n)$ are commutative tuples $\boldsymbol{b} \in M^n$ with $\prod_{i\in[n]} b_i = a$, and where for each $m \geqslant 1$ and each $\pi : [n] \to [m]$ the minor $\boldsymbol{b}^{(\pi)}$ is the tuple $\boldsymbol{c} \in M^m$ given by $c_j = \prod_{i\in\pi^{-1}(j)} b_i$, and the empty product equals the identity element e.

Theorem 5. *Let M be a finite monoid and let $a \in M$. Consider a template $(\boldsymbol{A}, \boldsymbol{B})$ with $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ homomorphically equivalent to $\mathscr{M}_{M,a}$. Then $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is solvable in polynomial time by $\mathrm{BLP} + \mathrm{AIP}$ if and only if a is regular in M. If a is not regular, then $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is* NP*-hard.*

Next, we show that there are templates whose polymorphism minions are of the considered type (up to isomorphism).

Theorem 6. *Let M be a monoid, and $a \in M$ an arbitrary element. Then the template $(\boldsymbol{A}, \boldsymbol{B})$ described below satisfies that $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B}) \simeq \mathscr{M}_{M,a}$.*[6] *The signature σ of $\boldsymbol{A}$ and $\boldsymbol{B}$ contains three relation symbols: a ternary symbol R, and two unary ones C_0, C_1. We define $A = \{0, 1\}$, $R^{\boldsymbol{A}} = \{(1,0,0), (0,1,0), (0,0,1)\}$, $C_0^{\boldsymbol{A}} = \{0\}$ and $C_1^{\boldsymbol{A}} = \{1\}$. The universe B of $\boldsymbol{B}$ is $\mathscr{M}_{M,a}(2)$. We define $R^{\boldsymbol{B}}$ as the set of triples in $(\mathscr{M}_{M,a}(2))^3$ of the form $((c_1, c_2c_3), (c_2, c_1c_3), (c_3, c_1c_2))$, where $c_1, c_2, c_3 \in M$ commute pairwise, and $c_1c_2c_3 = a$. Finally, the unary relations $C_0^{\boldsymbol{B}}$ and $C_1^{\boldsymbol{B}}$ are the singleton sets containing the tuples (e, a) and (a, e) respectively.*[7]

Finally, we remark that monoidal minions are natural objects of study, as they include other relevant previously studied minions.

Remark 1. Consider the Abelian monoid $M = \{0, 1, \epsilon\}$, whose multiplicative identity is 0, and where $1 \cdot 1 = 1 \cdot \epsilon = \epsilon \cdot \epsilon = \epsilon$. The elements of $\mathcal{M}_{M,1}$ are tuples with all zero entries except for a single 1 entry. Hence $\mathcal{M}_{M,1}$, corresponds to the so-called trivial minion $\mathscr{T}$ consisting of all projections (also known as dictators) on a two-element set. This minion represents the hardness boundary for CSPs, in the sense that a CSP is NP-hard if and only if its polymorphism minion maps homomorphically to $\mathscr{T}$ [18, 47].

Another example of a monoidal minion is the one capturing the power of arc consistency from [24]. In fact, every *linear* minion (in the sense of [23]) is a union of monoidal minions.[8]

If we allow infinite monoids to be considered, then monoidal minions include important minions that capture solvability via relevant algorithms. Consider the monoid $M = \{(r, z) \in \mathbb{Q} \times \mathbb{Z} \mid r \in [0, 1],$ and $r = 0$ implies $z = 0\}$, where the binary operation is given by coordinate-wise addition, and the identity is $(0, 0)$. Then $\mathscr{M}_{M,(1,1)}$ is precisely the minion $\mathscr{M}_{\mathrm{BLP+AIP}}$ described in [16], which expresses the power of BLP + AIP. Similarly, the minions described in [5] to capture the power of BLP and AIP are monoidal minions as well.

4 Monoidal Minions: Proof of Theorem 5

Solvability by BLP + AIP We show both directions. First we prove that a being regular implies that $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is solvable by BLP + AIP. We use the characterisation of the power of BLP + AIP from Theorem 2 for the tractability part of Theorem 5. Observe that if there is a minion homomorphism $\xi : \mathscr{M}_{M,a} \to \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ and $p \in \mathscr{M}_{M,a}$ is a $(2i+1)$-ary 2-block-symmetric element, then so is $\xi(p)$. Hence, showing that $\mathscr{M}_{M,a}$ has 2-block-symmetric elements of all odd arities proves that $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is solvable in polynomial time by BLP + AIP. By Lemma 1 (2), $a^j = a$ for some $j > 1$. Let $b = a^{j-2}$, where $a^0 = e$. Then $a^2b = a$, and $ab = ba$. For each $i \geqslant 1$ consider the $(2i+1)$-ary element of $\mathscr{M}_{M,a}$ consisting of $i+1$ consecutive a's followed by i consecutive b's. To see that this this is indeed an element of $\mathscr{M}_{M,a}$, observe that a and b commute, and $a^{i+1}b^i = a$ follows from $a^2b = a$. This tuple is 2-block-symmetric, with the blocks corresponding to a's and b's (of sizes $i+1$ and i, respectively).

[6]We use $\simeq$ to denote the isomorphism relation, i.e., the existence of a bijection between the minions that preserves arities and minor operations.

[7]The map $f : A \to B$ given by $0 \mapsto (e, a)$ and $1 \mapsto (a, e)$ is a homomorphism from $\boldsymbol{A}$ to $\boldsymbol{B}$. The structure $\boldsymbol{A}$ corresponds to the “1-in-3” template, where both constants are added, and $\boldsymbol{B}$ is the so-called “free structure” [5] of $\mathscr{M}_{M,a}$ generated by $\boldsymbol{A}$.

[8]We thank Lorenzo Ciardo for this observation.

Now we prove that if PCSP($\boldsymbol{A}, \boldsymbol{B}$) is solvable by BLP+AIP, then a must be regular. If PCSP($\boldsymbol{A}, \boldsymbol{B}$), then Pol($\boldsymbol{A}, \boldsymbol{B}$) has a $(2i+1)$-ary 2-block symmetric polymorphism p_i for each $i \geqslant 1$. As Pol($\boldsymbol{A}, \boldsymbol{B}$) is homomorphically equivalent to $\mathscr{M}_{M,a}$, we conclude there must be a $(2i+1)$-ary 2-block symmetric element $\boldsymbol{c_i}$ for each $i \geqslant 1$. Observe that $\boldsymbol{c_i}$ is a tuple in M^{2i+1} of the form $(\alpha_i, \ldots, \alpha_i, \beta_i, \ldots, \beta_i)$, where the first $i+1$ elements are equal to some $\alpha_i \in M$, and the last i elements are equal to some $\beta_i \in M$. By the definition of $\mathscr{M}_{M,a}$ it must hold that $\alpha_i^{i+1}\beta_i^i = a$, and that $\alpha_i\beta_i = \beta_i\alpha_i$. As M is finite, there must be a pair $(\alpha, \beta) \in M^2$ that appears infinitely often in the sequence $(\alpha_i, \beta_i)_{i \geqslant 1}$. Then, there must be two indices $j > 2i+1$, with $i \geqslant 1$, satisfying $(\alpha, \beta) = (\alpha_i, \beta_i) = (\alpha_j, \beta_j)$. The following chain of identities holds

$$a = \alpha^{i+1}\beta^i = \alpha^{j+1}\beta^j = \left(\alpha^{i+1}\beta^j\right)^2 \left(\alpha^{j-2i-1}\beta^{j-2i}\right) = a^2 \left(\alpha^{j-2i-1}\beta^{j-2i}\right).$$

This shows that a is regular.

NP-hardness We prove the intractability part of Theorem 5 (as well as other hardness results later in this paper) using the following result.

Theorem 7 ([5])**.** *Let $\mathscr{M} = \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$, and let $K, L \geqslant 1$ be any fixed integers. Suppose that $\mathscr{M}$ satisfies the following conditions:*

1. $\mathscr{M} = \bigcup_{\ell \in [L]} \mathscr{M}_\ell$*;*
2. *for each $\ell \in [L]$, there is a map $p \mapsto \mathcal{I}_\ell(p)$ that sends each $p \in \mathscr{M}_\ell$ to a set of its coordinates $\mathcal{I}_\ell(p)$ of size at most K;*
3. *for each $\ell \in [L]$ and for each minor $p^{(\pi)} = q$, where $p, q \in \mathscr{M}_\ell$, $\pi(\mathcal{I}_\ell(p)) \bigcap \mathcal{I}_\ell(q) \neq \varnothing$.*

Then PCSP($\boldsymbol{A}, \boldsymbol{B}$) *is* NP*-complete.*

Given a template $(\boldsymbol{A}, \boldsymbol{B})$, if there is a minion homomorphism $\xi : \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B}) \to \mathscr{M}_{M,a}$ and $\mathscr{M}_{M,a}$ satisfies the conditions in Theorem 7, so does Pol($\boldsymbol{A}, \boldsymbol{B}$). Indeed, if $\mathscr{M}_{M,a} = \bigcup_{\ell \in [L]} \mathscr{M}_\ell$, then we can write $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B}) = \bigcup_{\ell \in [L]} \xi^{-1}(\mathscr{M}_\ell)$. Additionally, if the map $\mathcal{I}_\ell$ witnesses the condition in the theorem for $\mathscr{M}_\ell$, then the map $\mathcal{I}'_\ell$ given by $p \mapsto \mathcal{I}_\ell(\xi(p))$ witnesses the same condition for $\xi^{-1}(\mathscr{M}_\ell)$. Hence, we show the hardness part of Theorem 5 by proving that $\mathscr{M}_{M,a}$ satisfies the assumptions in Theorem 7 when $a \in M$ is not regular.

For a monoid M, we we write $a \sqsubsetneq b$ when $a \sqsubseteq b$ holds but $b \sqsubseteq a$ does not. We use the following simple observation.

Observation 1. *Let M be a monoid and $a, b, c \in M$ three elements that commute pairwise. Suppose that $abc \sqsubsetneq ab$. Then $ac \sqsubsetneq a$.*

Proof. We prove the contrapositive. Suppose that $a \sqsubseteq ac$. That is, there is some $d \in M$ that satisfies $acd = a$. We have $(abc)d = (bac)d = b(acd) = ba = ab$, proving that $ab \sqsubseteq abc$. □

Assume that a is not regular. That is, that $a^2b \neq a$ for every $b \in M$. Let $\boldsymbol{b} \in \mathscr{M}_{M,a}(n)$ for some number $n \geqslant 1$. Using the fact that the elements b_i commute pairwise one can deduce that $\prod_{i \in I} b_i \sqsubseteq \prod_{j \in J} b_j$ for all $J \subseteq I \subseteq [n]$. A coordinate $j \in [n]$ is called *relevant* in $\boldsymbol{b}$ if $a \sqsubsetneq \prod_{i \in [n] \setminus \{j\}} b_i$. Consider the map $\mathcal{I}$ that assigns to each $\boldsymbol{b} \in \mathscr{M}_{M,a}$ its set of relevant coordinates. Claims 1 through 3 proved below establish the required assumptions in Theorem 7 with $L = 1$ and $K = |M|$, thus showing NP-hardness of PCSP($\boldsymbol{A}, \boldsymbol{B}$). Throughout the proof we adopt the convention that empty products over a monoid equal the identity element.

Claim 1: $\boldsymbol{b}$ has at most $|M|$ relevant coordinates. Let $\{i_1, \ldots, i_h\} \subseteq [n]$ be the set of relevant coordinates of $\boldsymbol{b}$. Given $k \in [h]$ we define

$$c_k = \prod_{j \in [k-1]} b_{i_j}, \quad \text{and} \quad d_k = \prod_{j \in [n] \setminus \{i_1, \ldots, i_k\}} b_j.$$

The following hold: (1) $a = d_k c_k b_{i_k}$, (2) b_{i_k}, c_k and d_k commute pairwise, and (3) as i_k is a relevant coordinate, it holds that $d_k c_k b_{i_k} \subsetneq d_k c_k$. Applying Observation 1, we obtain that $c_k b_{i_k} \subsetneq c_k$. Expanding the definition of c_k this means that

$$\prod_{j \in [k]} b_{i_j} \subsetneq \prod_{j \in [k-1]} b_{i_j}.$$

This holds for all $k \in [h]$, so in particular the products $\prod_{j \in [k]} b_{i_j}$ must be pairwise different and the number h of relevant coordinates is at most $|M|$, proving the claim.

Claim 2: Minors preserve relevant coordinates. Let $\boldsymbol{c} = \boldsymbol{b}^{(\pi)}$, where $\pi : [n] \to [m]$ is a map and let $i \in [n]$ be a relevant coordinate of $\boldsymbol{b}$. We want to show that $j = \pi(i)$ is a relevant coordinate of $\boldsymbol{c}$. Indeed, if that were not the case, using the equality $\prod_{k \in [n] \setminus \pi^{-1}(j)} b_k = \prod_{\ell \in [m] \setminus \{j\}} c_\ell$, we would have that

$$\prod_{k \in [n] \setminus \pi^{-1}(j)} b_k \sqsubseteq a.$$

Using this together with the fact that $\prod_{k \in [n] \setminus \{i\}} b_k \sqsubseteq \prod_{k \in [n] \setminus \pi^{-1}(j)} b_k$, where $i \in \pi^{-1}(j)$, shows that

$$\prod_{k \in [n] \setminus \{i\}} b_k \sqsubseteq a,$$

thus contradicting the fact that i was a relevant coordinate of $\boldsymbol{b}$.

Claim 3: $\boldsymbol{b}$ has at least one relevant coordinate. Suppose otherwise for the sake of contradiction. Then for each $i \in [n]$ there is an element $c_i \in M$ such that $ac_i = \prod_{i \in [n] \setminus \{j\}} b_i$. Let $c = \prod_{i \in [n]} c_i$. One can check that that $a^2 c = a$, contradicting our assumption that a was not regular. Indeed,

$$\begin{aligned}
a^2 c &= \left(\prod_{i=1}^{n} b_i\right)(ac_1)\left(\prod_{i=2}^{n} c_i\right) = \left(\prod_{i=1}^{n} b_i\right)\left(\prod_{i \in [n] \setminus \{1\}} b_i\right)\left(\prod_{i=2}^{n} c_i\right) \\
&= \left(\prod_{i=2}^{n} b_i\right)(ac_2)\left(\prod_{i=3}^{n} c_i\right) = \left(\prod_{i=2}^{n} b_i\right)\left(\prod_{i \in [n] \setminus \{2\}} b_i\right)\left(\prod_{i=3}^{n} c_i\right) \\
&= \left(\prod_{i=3}^{n} b_i\right)(ac_3)\left(\prod_{i=4}^{n} c_i\right) = \left(\prod_{i=3}^{n} b_i\right)\left(\prod_{i \in [n] \setminus \{3\}} b_i\right)\left(\prod_{i=4}^{n} c_i\right) \\
&= \cdots = \left(\prod_{i=n}^{n} b_i\right)\left(\prod_{i \in [n] \setminus \{n\}} b_i\right) = a.
\end{aligned}$$

Here we have repeatedly used the fact that the elements b_i commute pairwise and in particular they commute with $a = \prod_{i \in [n]} b_i$.

5 Equations Over Monoids and Groups: Proofs of Theorem 3 and Corollary 1

We begin with a simple characterisation of the polymorphisms of promise equation templates.

Lemma 4. *Consider a template* $\mathrm{PEqn}(Z_1, Z_2, \varphi)$ *of promise equations over semigroups, monoids, or groups, respectively. A map* $p : Z_1^n \to Z_2$ *is a polymorphism of* $\mathrm{PEqn}(Z_1, Z_2, \varphi)$ *if and only if* p *is a semigroup, monoid, or group homomorphism, respectively, and* $p(s, s, \ldots, s) = \varphi(s)$ *for all* $s \in \mathrm{dom}(\varphi)$*.*

Proof. We show the statement for the semigroup, monoid and the group case. The semigroup case is straightforward: p is a polymorphism if and only if it preserves $R_\times$ and R_s for all $s \in \mathrm{dom}(\varphi)$. Preserving $R_\times$ is equivalent to preserving the product operation from Z_1^n to Z_2, and preserving R_s means that $p(s, \ldots, s) = \varphi(s)$.

The monoid case follows in the same way. Using the same reasoning we obtain that p preserves the product operation from Z_1^n to Z_2, and that $p(s, \ldots, s) = \varphi(s)$ for all $s \in \mathrm{dom}(\varphi)$. The only additional requirement is that $p(e_{Z_1}, \ldots, e_{Z_1}) = e_{Z_2}$. This follows from the facts that $e_{Z_1} \in \mathrm{dom}(\varphi)$, and φ is a monoid homomorphism, so it must preserve identity elements. This means that $p(e_{Z_1}, \ldots, e_{Z_1}) = \varphi(e_{Z_1}) = e_{Z_2}$.

Finally, the group case is shown as the monoid case using that preserving inverse elements is just a consequence of preserving the product operation and preserving identity elements. □

Let us discuss some key properties of polymorphisms that will be used in the proof of Theorem 5. Given an n-ary polymorphism p of $\mathrm{PEqn}(M_1, M_2, \varphi)$, we define $\mathcal{N}(p)$ as the submonoid $\{p(s, \ldots, s) \mid s \in M_1\} \preceq M_2$. Given $i \in [n]$, we also define the submonoid $\mathcal{N}(p, i) \preceq M_2$ as

$$\{p(s_1, \ldots, s_n) \mid s_i \in M_1, \text{ and } s_j = e \text{ when } j \neq i\}.$$

We give some facts about these submonoids that follow directly from the definitions.

Observation 2. *Let* M_1, M_2 *be monoids and* φ *a monoid homomorphism with* $\mathrm{dom}(\varphi) \preceq M_1, \mathrm{Im}(\varphi) \preceq M_2$*. Let* p *be a* n*-ary polymorphism of* $\mathrm{PEqn}(M_1, M_2, \varphi)$*. Then the following statements hold:*

1. *The map* $\phi : \prod_{i \in [n]} \mathcal{N}(p, i) \to M_2$ *given by* $(s_1, \ldots s_n) \mapsto \prod_{i \in [n]} s_i$ *is a monoid homomorphism. In particular, given* $1 \leqslant i < j \leqslant n$*, any two elements* $t_1 \in \mathcal{N}(p, i)$*,* $t_2 \in \mathcal{N}(p, j)$ *commute.*

2. *If* $\mathcal{N}(p, i) = \mathcal{N}(p, j)$ *for some* $i \neq j \in [n]$ *then* $\mathcal{N}(p, i)$ *is Abelian.*

3. *The submonoid* $\mathcal{N}(p)$ *is contained in* $\mathrm{Im}(\phi)$*, where* ϕ *is as defined in Item 1. In particular, if* $\mathcal{N}(p)$ *is not Abelian, some* $\mathcal{N}(p, i)$ *must be non-Abelian.*

We are ready to prove our main result.

Theorem 3 (Main)**.** *Let* M_1*,* M_2 *be monoids and* φ *a monoid homomorphism with* $\mathrm{dom}(\varphi) \preceq M_1, \mathrm{Im}(\varphi) \preceq M_2$*. Then* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is solvable in polynomial time by* BLP + AIP *if and only if there is an Abelian homomorphism* $\psi : M_1 \to M_2$ *extending* φ *and* $\mathrm{Im}(\psi)$ *is a union of subgroups. If no such homomorphism* ψ *exists, then* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is* NP*-hard.*

Proof. First we show that the existence of such homomorphism ψ is equivalent to solvability by BLP + AIP. We prove both implications. Suppose that such homomorphism ψ exists. As $\mathrm{Im}(\psi)$ is a union of subgroups, by Lemma 1 there is some number $k > 1$ such that $s^k = s$ for all $s \in \mathrm{Im}(\psi)$. Let $n \geqslant 1$ be arbitrary. Consider the map $p : M_1^{2n+1} \to M_2$ given by

$$(s_i)_{i\in[2n+1]} \mapsto \left(\prod_{i\in[n+1]} \psi(s_i)\right)\left(\prod_{i\in[n]} \psi(s_{i+n+1})^{k-2}\right),$$

where the convention is that the zero-th power of an element equals the identity of the monoid. We claim that p is a 2-block-symmetric polymorphism of $\mathrm{PEqn}(M_1, M_2, \varphi)$ with the first block consisting of the first $n + 1$ coordinates, and the second block consisting of the rest. The fact that p is a 2-block-symmetric map with the blocks as claimed follows from the fact that ψ is Abelian. To complete the argument, we show that p is a polymorphism of $\mathrm{PEqn}(M_1, M_2, \varphi)$ using the characterisation from Lemma 4. First, observe that the fact that ψ is Abelian implies that p is a monoid homomorphism. Indeed,

$$\begin{aligned}
&p(s_1, \ldots, s_{2n+1})p(t_1, \ldots, t_{2n+1}) \\
&\qquad = \left(\prod_{i\in[n+1]} \psi(s_i)\psi(t_i)\right)\left(\prod_{i\in[n]} \psi(s_{i+n+1})^{k-1}\psi(t_{i+n+1})^{k-1}\right) \\
&\qquad = p(s_1t_1, \ldots, s_{2n+1}t_{2n+1}),
\end{aligned}$$

so p preserves products. Now we only need to prove that $p(s, \ldots, s) = \varphi(s)$ for all $s \in \mathrm{dom}(\varphi)$ in order to show that p is a polymorphism. To see that this holds, observe that

$$p(s, \ldots, s) = \psi(s)^{n(k-1)+1} = \psi(s) = \varphi(s),$$

where the last equality uses the fact that ψ extends φ. This completes the proof of the first implication via Theorem 2.

In the other direction, suppose that $\mathrm{PEqn}(M_1, M_2, \varphi)$ is solvable by BLP + AIP. That is, by Theorem 2, there is a 2-block-symmetric polymorphisms p_i of $\mathrm{PEqn}(M_1, M_2, \phi)$ of arity $2i + 1$ for each $i \geqslant 1$. For each i, we define three homomorphisms $\alpha_i, \beta_i, \gamma_i$ from M_1 to M_2. Given $a \in M_1$, we define $\alpha_i(a)$ as the element $p_i(\boldsymbol{a})$, where $a_1 = a$ and $a_j = e$ for all $j \neq 1$. Similarly, $\beta_i(a)$ is the element $p_i(\boldsymbol{a}')$, where $a'_{i+2} = a$, and $a'_j = e$ for all $j \neq i + 2$. This way, given an arbitrary $\boldsymbol{b} \in M_1^{2i+1}$, it holds that

$$p_i(\boldsymbol{b}) = \left(\prod_{j=1}^{i+1} \alpha_i(b_j)\right)\left(\prod_{j=i+2}^{2i+1} \beta_i(b_j)\right).$$

Finally, given $a \in M_1$, the element $\gamma_i(a)$ equals $p_i(a, a, \ldots, a)$, so, $\gamma_i(a) = \alpha_i(a)^{i+1}\beta_i(a)^i$. As the number of possible triples $(\alpha_i, \beta_i, \gamma_i)$ is finite, there is a choice (α, β, γ) that appears infinitely often in the family $(\alpha_i, \beta_i, \gamma_i)_{i=0}^{\infty}$. Let i, j be such that $(\alpha, \beta, \gamma) = (\alpha_i, \beta_i, \gamma_i) = (\alpha_j, \beta_j, \gamma_j)$ and $j \geqslant 2i + 1$. We claim that γ has all the properties of the map Ψ in the statement. We need to check that (I) γ extends φ, (II) γ has Abelian image, and (III) $\mathrm{Im}(\gamma)$ is a union of subgroups. Property (I) follows from the fact that p_i is a polymorphism. To show property (II), we first need to make some observations about $\mathrm{Im}(\alpha)$ and $\mathrm{Im}(\beta)$. By definition, $\mathrm{Im}(\alpha) = \mathcal{N}(p_j, 1)$ and $\mathrm{Im}(\beta) = \mathcal{N}(p_j, j + 2)$. By Item 1 in Observation 2, this implies that $ab = ba$ for all

$a \in \mathrm{Im}(\alpha), b \in \mathrm{Im}(\beta)$. Using the fact that p_j is 2-block symmetric and $j \geqslant 2$, we can deduce that $\mathcal{N}(p_j, 1) = \mathcal{N}(p_j, 2)$ and $\mathcal{N}(p_j, j+2) = \mathcal{N}(p_j, j+3)$. By Item 2 in Observation 2, this implies that both $\mathrm{Im}(\alpha)$ and $\mathrm{Im}(\beta)$ are Abelian monoids. Having shown these properties of $\mathrm{Im}(\alpha)$ and $\mathrm{Im}(\beta)$ we are ready to show (II). We need to prove that $\gamma(a)\gamma(b) = \gamma(b)\gamma(a)$ for all $a, b \in M_1$. Indeed,

$$\gamma(a)\gamma(b) = \left(\alpha(a^{j+1})\beta(a^j)\right)\left(\alpha(b^{j+1})\beta(b^j)\right) = \gamma(b)\gamma(a),$$

where the second equality uses the fact that all terms in the second expression commute due to our observations about $\mathrm{Im}(\alpha)$ and $\mathrm{Im}(\beta)$. Finally, let us show that (III) holds. By Lemma 1, we just need to show that for each $a \in M_1$ there is some $b \in M_2$ satisfying $\gamma(a)^2 b = \gamma(a)$. By our choice of i and j, for all $a \in M_1$ it holds that

$$\begin{aligned}\gamma(a) &= \alpha(a^{i+1})\beta(a^i) = \alpha(a^{j+1})\beta(a^j) \\ &= \alpha(a^{i+1})^2\alpha(a^{j-2i-1})\beta(a^i)^2\beta(a^{j-2i}) = \gamma^2(a)\left(\alpha^{j-2i-1}(a)\beta^{j-2i}(a)\right),\end{aligned}$$

where the fourth equality uses that all the terms in the fourth expression commute by our observations about $\mathrm{Im}(\alpha)$ and $\mathrm{Im}(\beta)$.

Finally, let us prove the second part of the theorem. We show that $\mathrm{PEqn}(M_1, M_2, \varphi)$ is NP-hard assuming there is no Abelian homomorphism $\psi : M_1 \to M_2$ extending φ whose image is a union of subgroups. Let $\mathscr{M}$ be the polymorphism minion of $\mathrm{PEqn}(M_1, M_2, \varphi)$. Given a polymorphism $p \in \mathscr{M}$, we define $\mathcal{N}(p)$ as the submonoid $\{p(s, \dots, s) \mid s \in M_1\} \preceq M_2$. Observe that by assumption, for a given polymorphism p it holds that the monoid $\mathcal{N}(p)$ is non-Abelian or that $\mathcal{N}(p)$ is not a union of subgroups. Define Ω as the set of monoid homomorphisms $\psi : M_1 \to M_2$ for which $\mathrm{Im}(\psi)$ is not a union of subgroups. By Lemma 1, this happens precisely when $\mathrm{Im}(\psi)$ contains some non-regular element $a \in M_2$. Let $L = |\Omega| + 1$, and let $K = \max(|M_2|, |\{N \preceq M_2 \mid N \text{ is non-Abelian }\}|)$. We use Theorem 7 with the constants L, K to show NP-hardness. We define the following subminions of $\mathscr{M}$.

$$\mathscr{M}_{\mathrm{A}} = \{p \in \mathscr{M}, \mid \mathcal{N}(p) \text{ is not Abelian}\},$$

and given any monoid homomorphism $\psi \in \Omega$ we set

$$\mathscr{M}_\psi = \{p \in \mathscr{M}, \mid p(s, \dots, s) = \psi(s) \text{ for all } s \in M_1\}.$$

By the previous observation it holds that

$$\mathscr{M} = \mathscr{M}_{\mathrm{A}} \bigcup_{\psi \in \Omega} \mathscr{M}_\psi.$$

We give selection functions $\mathcal{I}$ for each of these sub-minions satisfying the assumptions of Theorem 7. Let p be any n-ary polymorphism in $\mathscr{M}_{\mathrm{A}}$. Given $i \in [n]$ we define $\mathcal{N}(p, i) \preceq M_2$ as the submonoid

$$\{p(s_1, \dots, s_n) \mid s_i \in M_1, \text{ and } s_j = e \text{ when } j \neq i\}.$$

We give some facts about these submonoids.

Given an n-ary polymorphism $p \in \mathscr{M}_{\mathrm{A}}$, we define $\mathcal{I}_{\mathrm{A}}(p) \subseteq [n]$ as the set of coordinates i for which $\mathcal{N}(p, i)$ is non-Abelian. We claim that $\mathcal{I}_{\mathrm{A}}$ satisfies the assumptions of Theorem 7. Indeed, given some n-ary p:

- $\mathcal{I}_{\mathrm{A}}(p)$ is non empty by Item 3 in Observation 2.
- $|\mathcal{I}_{\mathrm{A}}(p)| \leqslant K$. Otherwise it would be that $\mathcal{N}(p, i) = \mathcal{N}(p, j)$ for some different $i, j \in \mathcal{I}_{\mathrm{A}}(p)$, contradicting the fact that $\mathcal{N}(p, i)$ is non-Abelian (by Item 2 in Observation 2).
- Suppose that $p = q^{(\pi)}$ for some m-ary q and some $\pi : [m] \to [n]$. Let $i \in \mathcal{I}_{\mathrm{A}}(p)$, then

$$\mathcal{N}(p, i) \subseteq \left\{ \prod_{j \in \pi^{-1}(i)} s_j \mid s_j \in \mathcal{N}(s, j) \text{ for all } j \in \pi^{-1}(i) \right\}.$$

  As $\mathcal{N}(p, i)$ is non-Abelian, some submonoid $\mathcal{N}(q, j)$ with $j \in \pi^{-1}(i)$ must be non-Abelian as well. This means that $\mathcal{I}_{\mathrm{A}}(p) \subseteq \pi(\mathcal{I}_{\mathrm{A}}(q))$.

Now consider an arbitrary homomorphism $\psi \in \Omega$ for which $\mathscr{M}_\psi$ is non-empty. We define a selection function $\mathcal{I}_\psi$ satisfying the assumptions of Theorem 7. Let $t \in \mathrm{Im}(\psi)$ be a non-regular element, and let $s \in M_1$ be such that $\psi(s) = t$. Let $\mathscr{M}_{M_2,t}$ be the monoidal minion defined in Definition 3. Consider the map $\xi : \mathscr{M}_\psi \to \mathscr{M}_{M_2,t}$ that sends any n-ary polymorphism $p \in \mathscr{M}_\psi$ to the tuple $(r_1, \dots, r_n) \in \mathscr{M}_{M_2,t}(n)$ where for each $i \in [n]$

$$r_i = p(s_1, \dots, s_n), \qquad \text{where } s_i = s, \text{ and } s_j = e \text{ for all } j \neq i\ .$$

Observe that this is a well-defined minion homomorphism from $\mathscr{M}_\psi$ to $\mathscr{M}_{M_2,t}$. Indeed, first note that $(r_1, \dots, r_n)$ belongs to the second minion. This holds because $r_1 r_2 \dots r_n = p(s, \dots, s) = \psi(s) = t$, and, for each $i \in [n]$, the element r_i belongs to $\mathcal{N}(p, i)$, so the r_i's commute pairwise by Item 1 in Observation 2. One can also check that ξ preserves minors.

From the proof of Theorem 5 there is some selection function $\mathcal{I}$ on $\mathscr{M}_{M_2,t}$ satisfying the hypotheses of Theorem 7 for some constant $K' = |M_2| \leqslant K$ and $L = 1$. Thus, we can define $\mathcal{I}_\psi$ on $\mathscr{M}_\psi$ simply by setting $\mathcal{I}_\psi(p) = \mathcal{I}(\xi(p))$ for each polymorphism $p \in \mathscr{M}_\psi$.

Hence, have defined selection functions $\mathcal{I}_{\mathrm{A}}$ and $\mathcal{I}_\psi$ for each $\psi \in \Omega$ that satisfy the requirements of Theorem 7, showing that $\mathrm{PEqn}(M_1, M_2, \varphi)$ is NP-hard. $\square$

Corollary 1. *Let G_1, G_2 be groups and φ a group homomorphism with $\mathrm{dom}(\varphi) \preceq G_1, \mathrm{Im}(\varphi) \preceq G_2$. Then $\mathrm{PEqn}(G_1, G_2, \varphi)$ is solvable in polynomial time by* AIP *if and only if there is an Abelian homomorphism $\psi : G_1 \to G_2$ extending φ. If no such homomorphism ψ exists, then $\mathrm{PEqn}(G_1, G_2, \varphi)$ is* NP*-hard.*

Proof. We prove both directions. The hardness case follows from Theorem 3. Indeed, $\mathrm{PEqn}(G_1, G_2, \varphi)$ is a template of promise equations over monoids (where the monoids just happen to be groups). Suppose that there is no Abelian group homomorphism $\psi : G_1 \to G_2$ that extends φ. Observe that a monoid homomorphism between two groups must also be a group homomorphism, so there is no Abelian monoid homomorphism $\psi : G_1 \to G_2$ that extends φ. Thus, by Theorem 3, $\mathrm{PEqn}(G_1, G_2, \varphi)$ is NP-hard.

In the other direction, suppose that such a ψ exists. We show that $\mathrm{PEqn}(G_1, G_2, \varphi)$ is solved by AIP using Theorem 1. Let n be any odd arity and let $p : G_1^n \to G_2$ be the map given by $p(g_1, \dots, g_n) \mapsto \prod_{i \in [n]} t_i$, where $t_i = \psi(g_i)$ for every odd i, and $t_i = \psi(g_i)^{-1}$ for every even i. Then p is an alternating polymorphism of $\mathrm{PEqn}(G_1, G_2, \varphi)$. $\square$

6 Equations over Semigroups: Proof of Theorem 4

A *digraph* $\boldsymbol{D}$ is a relational structure whose signature consists of a single binary relation symbol E.

We follow closely the ideas from [35, Theorem 7]. That result states that every CSP is polynomial-time equivalent to a problem of the form $\mathrm{Eqn}(S, S)$ for some semigroup S. Their proof uses the fact that every CSP is polynomial-time equivalent to another CSP whose template is a digraph $\boldsymbol{D}$ with all singleton unary relations [26]. The fact that they consider these unary relations on $\boldsymbol{D}$ yields equations in $\mathrm{Eqn}(S, S)$ where all constants are allowed. For PCSPs, however, this is our starting point.

Theorem 8 ([13]). *For every template $(\boldsymbol{A}_1, \boldsymbol{A}_2)$ there is a template $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ of digraphs such that $\mathrm{PCSP}(\boldsymbol{A}_1, \boldsymbol{A}_2)$ is polynomial-time equivalent to $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$.*

The fact that we lack singleton unary relations in the templates $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ is the main obstacle for applying the techniques from [35]. We overcome this by extending our digraphs with an additional edge joining two fresh distinguished vertices. The relational signature σ^+ contains one binary relation symbol E, and two unary relation symbols P, Q. Given a digraph $\boldsymbol{D}$, we write $\boldsymbol{D}^+$ for the σ^+ structure defined by $D^+ = D \cup \{p, q\}$, where p and q are fresh vertices, $E^{\boldsymbol{D}^+} = E^{\boldsymbol{D}} \cup \{(p, q)\}$, $P^{\boldsymbol{D}^+} = \{p\}$, and $Q^{\boldsymbol{D}^+} = \{q\}$.

Lemma 5. *Let $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ be a template of digraphs. Then $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ is polynomial-time Turing-equivalent to $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$.*

Proof. We give polynomial-time Turing reductions in both directions. First, we reduce from $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ to $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$. We consider two cases. Suppose that $E^{\boldsymbol{D}_2}$ is empty. Then $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ amounts to deciding whether a given instance $\boldsymbol{I}$ has an edge or not, which takes polynomial time. Otherwise, assume that $E^{\boldsymbol{D}_2}$ is non-empty. Then our reduction takes any instance $\boldsymbol{I}$ of $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ and considers it as an instance of $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$ where the unary relations are empty. Clearly, if $\boldsymbol{I}$ maps homomorphically to $\boldsymbol{D}_1$ then it also maps homomorphically to $\boldsymbol{D}_1^+$ using the same homomorphism. Otherwise, if $\boldsymbol{I}$ does not map homomorphically to $\boldsymbol{D}_2$ then it cannot map homomorphically to $\boldsymbol{D}_2^+$. Indeed, to see this observe that the digraph resulting from of $\boldsymbol{D}_2^+$ (by forgetting about the P, Q relations) maps homomorphically to $\boldsymbol{D}_2$: it suffices to send the edge (p, q) to an arbitrary edge in $E^{\boldsymbol{D}_2}$, which is non-empty by assumption.

Now we describe a polynomial-time reduction from $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$ to $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$. The reduction considers an instance $\boldsymbol{I}$ of $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$ and checks in polynomial time whether every connected component of $\boldsymbol{I}$ that intersects $P^{\boldsymbol{I}}$ or $Q^{\boldsymbol{I}}$ maps homomorphically to the edge structure $\boldsymbol{W}$ with $W = \{p, q\}$, $E^{\boldsymbol{W}} = \{(p, q)\}$, $P^{\boldsymbol{W}} = \{p\}$, and $Q^{\boldsymbol{W}} = \{q\}$. If this is not the case, $\boldsymbol{I}$ is rejected. Otherwise, we remove from $\boldsymbol{I}$ the components that intersect $P^{\boldsymbol{I}}$ or $Q^{\boldsymbol{I}}$. Next, we check in polynomial time whether each remaining component of $\boldsymbol{I}$ can be mapped homomorphically to $\boldsymbol{W}$, and remove the ones that do. The resulting instance $\boldsymbol{I}'$ is equivalent to the original $\boldsymbol{I}$ in the sense that $\boldsymbol{I}$ maps to $\boldsymbol{D}_i^+$ if and only if $\boldsymbol{I}'$ does so as well. Furthermore, observe that a homomorphism from $\boldsymbol{I}'$ to $\boldsymbol{D}_i^+$ cannot include p and q in its image, as there are no components in $\boldsymbol{I}'$ that map homomorphically to $\boldsymbol{W}$. This means that $\boldsymbol{I}'$ maps to $\boldsymbol{D}_i^+$ if and only if it maps to $\boldsymbol{D}_i$. Hence, as the last step in our reduction we simply use $\boldsymbol{I}'$ as an instance of $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$. □

A semigroup S is a *right-normal band* if $ss = s$ for all $s \in S$ and $rst = srt$ for all $r, s, t \in S$. Recall that we write $s \sim r$ if $s \sqsubseteq r$ and $r \sqsubseteq s$ hold. It follows from the definitions that the quotient $\widehat{S} = S/\sim$ inherits the semigroup structure from S. Moreover, $\widehat{S}$ is a *semilattice*, meaning that it is an Abelian semigroup where every element is idempotent. Given an instance $\boldsymbol{I}$ of $\mathrm{Eqn}(S, S)$ we denote by $\widehat{\boldsymbol{I}}$ the corresponding instance over $\widehat{S}$, where every constant s is substituted by its $\sim$ class $\hat{s}$.

We need two lemmas from [35] and a simple observation.

Lemma 6 ([35])**.** *Let S be a semilattice. Then* $\mathrm{Eqn}(S, S)$ *can be solved in polynomial time. Moreover, if an instance $\boldsymbol{I}$ has a solution, it also has a unique minimal one (with respect to the $\sqsubseteq$ preorder) that can be obtained in polynomial time.*

Lemma 7 ([35])**.** *Let S be a right-normal band. Then an instance $\boldsymbol{I}$ of* $\mathrm{Eqn}(S, S)$ *is solvable if it has a solution f satisfying $f(x) \in \hat{s}_x$, for all $x \in I$, where the map $x \mapsto \hat{s}_x$ is the minimal solution of $\widehat{\boldsymbol{I}}$ in* $\mathrm{Eqn}(\widehat{S}, \widehat{S})$.

Observation 3. *Let S be a right-normal band, and let $s, s', t \in S$ be three arbitrary elements with $s \sim s'$. Then $st = s't$.*

Proof. As $s \sim s'$, it must hold that $s = s'r'$ and $s' = sr$ for some $r, r' \in S$. Thus, $st = s'r't = srr't$, and $s't = srt = s'r'rt = srr'rt = srr't$, where the last equality holds since S is a right-normal band. □

Let $\boldsymbol{D}$ be a digraph. We define a semigroup S_D related to $\boldsymbol{D}$ in a similar fashion as [35]. The main difference is that we need to "plant" a special subsemigroup S_W inside S_D that is used later as the set of constants in our promise equations. The semigroup $S = S_D$ is a right-normal band. It has the following $\sim$-classes: $V^{\mathrm{L}}, V^{\mathrm{R}}, V^{\mathrm{LC}}, V^{\mathrm{LR}}, V^{\mathrm{CR}}, E^{\mathrm{C}}, 0$, described as follows. Given $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LC}, \mathrm{LR}, \mathrm{CR}\}$ the class $V^{\square}$ is a copy of $D \cup \{p, q\}$. That is, $V^{\square} = \{v^{\square} \mid v \in D\} \cup \{p^{\square}, q^{\square}\}$. The class E^{C} is a copy of $E^D \cup \{(p, q)\}$, meaning that $E^{\mathrm{C}} = \{(u, v)^{\mathrm{C}} \mid (u, v) \in E^D\} \cup \{(p, q)^{\mathrm{C}}\}$. The letters L, R, and C stand for *left, right, and center*, respectively. Finally, the class 0 contains a single element 0. By Observation 3, in a right-normal band T it must hold that $st = s't$ for all $s, s', t \in T$ with $s \sim s'$. Hence, given a $\sim$-class $C \subseteq T$ and an element t we abuse the notation and write Ct to denote the product of an arbitrary element from C with t. The product operation in S is given by the following rules:

$$
\begin{aligned}
V^{\mathrm{R}} v^{\mathrm{L}} = V^{\mathrm{L}} v^{\mathrm{R}} = V^{\mathrm{LR}} v^{\mathrm{R}} = V^{\mathrm{LR}} v^{\mathrm{L}} = V^{\mathrm{L}} v^{\mathrm{LR}} = V^{\mathrm{R}} v^{\mathrm{LR}} &= v^{\mathrm{LR}} \\
V^{\mathrm{L}} v^{\mathrm{LC}} = V^{\mathrm{LC}} v^{\mathrm{L}} = E^{\mathrm{C}} v^{\mathrm{L}} = E^{\mathrm{C}} v^{\mathrm{LC}} &= v^{\mathrm{LC}} \\
V^{\mathrm{R}} v^{\mathrm{CR}} = V^{\mathrm{CR}} v^{\mathrm{R}} = E^{\mathrm{C}} v^{\mathrm{R}} = E^{\mathrm{C}} v^{\mathrm{CR}} &= v^{\mathrm{CR}},
\end{aligned}
$$

where v is an arbitrary element in $D \cup \{p, q\}$. Additionally,

$$
V^{\mathrm{L}}(u, v)^{\mathrm{C}} = V^{\mathrm{LC}}(u, v)^{\mathrm{C}} = u^{\mathrm{LC}}, \quad \text{and} \quad V^{\mathrm{R}}(u, v)^{\mathrm{C}} = V^{\mathrm{CR}}(u, v)^{\mathrm{C}} = v^{\mathrm{CR}},
$$

where (u, v) belongs to $E^D \cup \{(p, q)\}$. Finally, all other products not described above have 0 as their result.

Let us give some intuition about the semigroup S_D. Our goal is to encode the incidence structure of the digraph D in a semigroup. The construction S_D achieves this as follows. The rule $V^{\mathrm{L}}(u, v)^{\mathrm{C}} = u^{\mathrm{LC}}$ states that multiplying any L-copy of a vertex w^{L} with an edge $(u, v)^{\mathrm{C}}$ results in the LC-copy of the edge's source u^{LC}. Similarly, we can extract information about an edge's target by multiplying with an R-copy of a vertex using the rule $V^{\mathrm{R}}(u,v)^{\mathrm{C}} = v^{\mathrm{CR}}$. Finally, the identities $V^{\mathrm{R}}v^{\mathrm{L}} = V^{\mathrm{L}}v^{\mathrm{R}} = v^{\mathrm{LR}}$ show that an L-copy of a vertex v^{L} and an R-copy of a vertex u^{R} commute if and only if they are copies of the same vertex (i.e., $v = u$).

We define the subsemigroup $S_W \leq S_D$ as the one containing the elements $0, (p,q)^{\mathrm{C}}, p^{\square}, q^{\square}$ for $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LC}, \mathrm{LR}, \mathrm{CR}\}$. Observe that for any digraph D, the quotient $\widehat{S_D} = S_D/\sim$ is isomorphic to $\widehat{S_W} = S_W/\sim$. This semilattice is depicted in Figure 1.

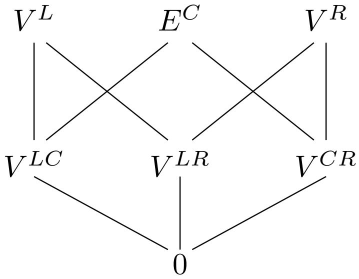

Figure 1: The semilattice $\widehat{S_D}$, where lines indicate the order.

Lemma 8. *There is a polynomial-time algorithm Φ that takes as an input a σ^+-structure $\boldsymbol{I}$ and outputs a system of equations $\Phi(\boldsymbol{I})$ with constants in S_W satisfying that, for any digraph $\boldsymbol{D}$, $\boldsymbol{I}$ maps into $\boldsymbol{D}^+$ if and only if $\Phi(\boldsymbol{I})$ has a solution over S_D.*

Proof. This follows the first reduction in [35, Theorem 7] while making sure that all constants remain in S_W. We construct the system $\Phi(\boldsymbol{I})$. For every vertex $v \in I$ we include variables $v^{\mathrm{L}}, v^{\mathrm{R}}$. For each $\square \in \{\mathrm{L}, \mathrm{R}\}$ we include the constraint $v^{\square} \in V^{\square}$, which is a shorthand for the equations $p^{\square}v^{\square} = v^{\square}$ and $v^{\square}p^{\square} = p^{\square}$. We also include the equation $p^{\mathrm{LR}}v^{\mathrm{L}} = p^{\mathrm{LR}}v^{\mathrm{R}}$. If $v \in P^{\boldsymbol{I}}$ we include the constraints $v^{\square} = p^{\square}$ for $\square \in \{\mathrm{L}, \mathrm{R}\}$. Similarly, if $v \in Q^{\boldsymbol{I}}$, then we include the constraints $v^{\square} = q^{\square}$. For each edge $(u,v) \in E^{\boldsymbol{I}}$ we include a variable $(u,v)^{\mathrm{C}}$ in $\Phi(\boldsymbol{I})$, together with the constraint $(u,v)^{\mathrm{C}} \in E^{\mathrm{C}}$, which is a shorthand for the equations $(u,v)^{\mathrm{C}}(p,q)^{\mathrm{C}} = (p,q)^{\mathrm{C}}$ and $(p,q)^{\mathrm{C}}(u,v)^{\mathrm{C}} = (u,v)^{\mathrm{C}}$. Finally, we also add the equations $p^{\mathrm{LC}}(u,v)^{\mathrm{C}} = p^{\mathrm{LC}}u^{\mathrm{L}}$ and $p^{\mathrm{CR}}(u,v)^{\mathrm{C}} = p^{\mathrm{CR}}v^{\mathrm{R}}$. The resulting system $\Phi(\boldsymbol{I})$ satisfies the statement of the theorem. $\square$

Lemma 9. *There is a polynomial-time algorithm Ψ that takes as an input a system of equations $\boldsymbol{X}$ with constants in S_W and produces one of the following outcomes:*

(I) It outputs a σ^+-structure $\Psi(\boldsymbol{X})$ that maps into $\boldsymbol{D}^+$ for a digraph $\boldsymbol{D}$ if and only if $\boldsymbol{X}$ has a solution over S_D, or

(II) it rejects $\boldsymbol{X}$ and $\boldsymbol{X}$ has no solution over S_D for any digraph $\boldsymbol{D}$.

Proof. We describe the algorithm Ψ. This algorithm is meant to transform the system $\boldsymbol{X}$ into a system of the form $\Phi(\boldsymbol{I})$, for the algorithm Φ given in Lemma 8 and some σ^+-structure $\boldsymbol{I}$. This time we follow the second reduction in [35, Theorem 7] while making sure that all constants in $\boldsymbol{X}$ remain in S_W throughout all the transformations.

Without loss of generality, we may assume that every equation in $\boldsymbol{X}$ is initially of the form $x_1x_2 = x_3$, for some variables x_1, x_2, x_3, or of the form $x = s$, for some variable x and some element $s \in S_W$. Consider the system $\widehat{\boldsymbol{X}}$ with constants in $\widehat{S_W} = S_W/\sim$. By Lemma 6 we can find a minimal solution of $\widehat{\boldsymbol{X}}$ in polynomial time. If such a solution does not exist, then the system $\boldsymbol{X}$ is not satisfiable over S_D for any digraph $\boldsymbol{D}$, and the algorithm Ψ just rejects it. Otherwise, suppose that the system $\widehat{\boldsymbol{X}}$ has some minimal solution. This solution maps each variable $x \in X$ to a $\sim$-class C_x of S_W. Consider an arbitrary digraph $\boldsymbol{D}$. Using the observation that $\widehat{S_W} \simeq \widehat{S_D}$ and Lemma 7, we deduce that $\boldsymbol{X}$ has a solution over S_D if and only if it has a solution where the value of each variable $x \in X$ belongs to the class C_x. Given a class C_x, we define the constant $c_x \in S_W$ as

- $p^{\square}$ if C_x is the class $V^{\square}$ for $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LC}, \mathrm{LR}, \mathrm{CR}\}$,
- $(p, q)^{\mathrm{C}}$ if $C_x = E^{\mathrm{C}}$, or
- 0 if $C_x = 0$.

For each variable $x \in X$ we add the equations $c_x x = x$ and $x c_x = c_x$. These equations are equivalent to the constraint that $x \in C_x$ (and we use $x \in C_x$ as a shorthand for those equations), so the resulting system is satisfiable over a semigroup S_D if and only if the original one was. Additionally, once every variable x is constrained to take values inside C_x, we can replace every equation of the form $x_1 x_2 = x_3$ in $\boldsymbol{X}$ with the equation $c_{x_3} x_2 = c_{x_3} x_3$ to yield an equivalent system. Indeed, it must hold that $c_{x_i} x_i = x_i$, so the equation $x_1 x_2 = x_3$ is equivalent to $c_{x_1} x_1 c_{x_2} x_2 = c_{x_3} x_3$. Not only that, but S_D is a normal band and $x_1 c_{x_1} = c_{x_1}$, so last equation is equivalent to $c_{x_1} c_{x_2} x_2 = c_{x_3} x_3$. Finally, the classes $C_{x_1}, C_{x_2}, C_{x_3}$ were part of a solution to $\widehat{\boldsymbol{X}}$, so it must be that $c_{x_1} c_{x_2} \sim c_{x_3}$, and by Observation 3 it holds that $c_{x_1} c_{x_2} c_{x_1} = c_{x_3} c_{x_1}$.

Every resulting equation of the form $0 x_1 = 0 x_2$ is trivially satisfied and can be discarded. Consider a variable $x \in X$ whose corresponding class C_x is 0. As we have removed every equation of the form $0 x_1 = 0 x_2$, x can only appear in constraints of the form $x \in 0$, and $x = 0$. These are trivially satisfiable by any assignment that maps x to 0, so we can remove the variable x and all equations containing it.

We are left with a system $\boldsymbol{X}$ where each variable is bound to a class $V^{\square}$ for $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LC}, \mathrm{LR}, \mathrm{CR}\}$ or E^{C}. Consider a variable $x \in X$ bound to the class V^{LC}. Suppose this variable appears in some equation of the form $c_1 x = c_1 y$, and consider the class C of c_1. By construction, it must be that $C \sqsupseteq V^{\mathrm{LC}}$ in $\widehat{S_W}$. However, we have removed all equations containing 0, so the only possibility left is that $C = V^{\mathrm{LC}}$. Suppose that we replace the requirement $x \in V^{\mathrm{LC}}$ with $x \in V^{\mathrm{L}}$ and every equation of the form $x = v^{\mathrm{LC}}$, where $v^{\mathrm{LC}} \in S_W$ is a constant, with $x = v^{\mathrm{L}}$. We claim the system $\boldsymbol{X}$ remains equivalent after these changes. Indeed, this results from the observation that $V^{\mathrm{LC}} v^{\mathrm{L}} = V^{\mathrm{LC}} v^{\mathrm{LC}}$ in any semigroup S_D for any vertex $v \in D^+$. By the same logic we can also replace any requirement of the kind $x \in V^{\mathrm{LR}}$ or $x \in V^{\mathrm{CR}}$ with $x \in V^{\mathrm{R}}$.

Consider any equation of the form $x = (u, v)^{\mathrm{C}}$ for a constant $(u, v)^{\mathrm{C}}$. This equation is equivalent to the constraints $p^{\mathrm{LC}} x = p^{\mathrm{LC}} y$, $p^{\mathrm{CR}} x = p^{\mathrm{CR}} z$, $y = u^{\mathrm{L}}$ and $z = v^{\mathrm{R}}$, where y and z are fresh variables, further restricted to $y \in V^{\mathrm{L}}, z \in V^{\mathrm{R}}$. Hence, we can substitute in X the original equation with these constraints to obtain an equivalent system.

Consider an equation of the form $cx = cy$, where both x, y are constrained to be in c's $\sim$-class. This equation holds if and only if $x = y$. Hence, we may remove this equation and identify both variables x, y together.

This far we have obtained a system $\boldsymbol{X}$ where each variable is bound to either $V^{\mathrm{L}}, V^{\mathrm{R}}$ or E^{C}, and the only constants are among $p^{\mathrm{L}}, p^{\mathrm{R}}, q^{\mathrm{L}}$, q^{R}. After identifying variables and adding dummy variables if necessary we can assume the following:

- For each variable $x \in X$ constrained by $x \in E^{\mathrm{C}}$ there is exactly one variable x_{L} constrained by $x_{\mathrm{L}} \in V^{\mathrm{L}}$ in an equation of the form $p^{\mathrm{LC}} x = p^{\mathrm{LC}} x_{\mathrm{L}}$, and exactly one variable x_{R} constrained by $x_{\mathrm{R}} \in V^{\mathrm{R}}$ that appears in an equation of the form $p^{\mathrm{CR}} x = p^{\mathrm{CR}} x_{\mathrm{R}}$.

- For any variable x_{L} constrained by $x_{\mathrm{L}} \in V^{\mathrm{L}}$, there is exactly one variable x_{R} constrained by $x_{\mathrm{R}} \in V^{\mathrm{R}}$ that appears in an equation of the form $p^{\mathrm{LR}} x_{\mathrm{L}} = p^{\mathrm{LR}} x_{\mathrm{R}}$. The same remains true after swapping R and L.
- No two different variables $x, y \in X$ constrained by $x, y \in E^{\mathrm{C}}$ satisfy $x_{\mathrm{L}} = y_{\mathrm{L}}$ and $x_{\mathrm{R}} = y_{\mathrm{R}}$.
- Not considering equations that are part of the constraints $x \in C$ for some $\sim$-class C, each equation is of the form (i) $p^{\mathrm{LR}} x = p^{\mathrm{LR}} y$ with $x \in V^{\mathrm{L}}$ and $y \in V^{\mathrm{R}}$, (ii) $p^{\mathrm{LC}} x = p^{\mathrm{LC}} x_{\mathrm{L}}$ or $p^{\mathrm{CR}} x = p^{\mathrm{CR}} x_{\mathrm{R}}$ for some $x \in E^{\mathrm{C}}$, or (iii) $x = p^{\square}$ or $x = q^{\square}$ for $\square \in \{\mathrm{L}, \mathrm{R}\}$.

One can see that such a system corresponds to $\Phi(\boldsymbol{I})$ for some σ^{+}-structure $\boldsymbol{I}$ that can be built in polynomial time. Then Ψ returns $\boldsymbol{I}$, which satisfies our requirements by Lemma 8. $\square$

Corollary 4. *Let $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ be a template of digraphs. Then $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ is polynomial-time Turing-equivalent to $\mathrm{PEqn}(S_{D_1}, S_{D_2}, \varphi)$, where $\varphi = \mathrm{id}_{S_W}$.*

Proof. We show that $\mathrm{PEqn}(S_{D_1}, S_{D_2}, \varphi)$ is polynomial-time equivalent to $\mathrm{PCSP}(\boldsymbol{D}_1^{+}, \boldsymbol{D}_2^{+})$, which suffices by Lemma 5. Observe that algorithm Φ given in Lemma 8 is a polynomial-time Turing reduction from $\mathrm{PCSP}(\boldsymbol{D}_1^{+}, \boldsymbol{D}_2^{+})$ to $\mathrm{PEqn}(S_{D_1}, S_{D_2}, \varphi)$, and algorithm Ψ, given in Lemma 9 is a polynomial-time Turing reduction in the other direction. $\square$

Corollary 4 and Theorem 8 establish Theorem 4.

7 Explicit Templates: Proof of Theorem 6

We describe a bijective minion homomorphism $\xi : \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B}) \to \mathscr{M}_{M,a}$. Let us introduce some notation before the start. We identify the powerset $2^{[n]}$ with the set of tuples $\{0,1\}^n$ by associating each set $S \subseteq [n]$ to the n-tuple whose i-th entry is one if and only if $i \in S$ (i.e., the characteristic vector of S). Thus, we see a n-ary polymorphism $p \in \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ as a map from $2^{[n]}$ to $\mathscr{M}_{M,a}(2)$. Following this convention, three sets $X_1, X_2, X_3 \subseteq [n]$ belong to the relation $R^{\boldsymbol{A}^n}$ if and only if they are a partition of $[n]$. Similarly, the unary relation $C_0^{\boldsymbol{A}^n}$ contains only the empty set, and $C_1^{\boldsymbol{A}^n}$ contains only the whole set $[n]$.

Let us carry on with the description of ξ. Given a n-ary polymorphism $p \in \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$, ξ maps it to the tuple $\boldsymbol{b}_p = (b_{p,1}, \ldots, b_{p,n}) \in \mathscr{M}_{M,a}$ defined as follows. For each $i \in [n]$, let $(c_{i,1}, c_{i,2}) = p(\{i\})$. Then we set $b_{p,i} = c_{i,1}$. In order to prove that ξ is a bijective minion homomorphism we need to show that (I) $\boldsymbol{b}_p$ is an element of $\mathscr{M}_{M,a}$ for all polymorphisms p, (II) that ξ preserves minor operations, and (III) that ξ is a bijection.

Before moving on with the rest of the proof, we recall the definition of $R^{\boldsymbol{B}}$:

$$((r_1, s_1), (r_2, s_2), (r_3, s_3)) \in R^{\boldsymbol{B}} \text{ if and only if}$$
$$r_1, r_2, r_3 \text{ commute pairwise, and } s_1 = r_2 r_3, \quad s_2 = r_3 r_1, \quad s_3 = r_1 r_2. \tag{1}$$

Below we prove some claims about the map ξ and polymorphisms of $(\boldsymbol{A}, \boldsymbol{B})$ that are used to show (I), (II), and (III). Fix some n-ary polymorphism p.

Claim 1: $b_{p,i}$ and $b_{p,j}$ commute in M for any $1 \leqslant i < j \leqslant n$. Fix different indices $i, j \in [n]$. Clearly, the sets $\{i\}, \{j\}, [n] \backslash \{i, j\}$ form a partition of $[n]$, so $(p(\{i\}), p(\{j\}), p([n] \backslash \{i, j\})$ must belong to $R^{\boldsymbol{B}}$. Thus, there are elements $r_1, r_2, r_3 \in M$ witnessing Equation (1). In particular, $b_{p,i} = r_1$ and $b_{p,j} = r_2$, so $b_{p,i}$ and $b_{p,j}$ commute, establishing the claim.

Claim 2: $p([n]) = (a, e)$, and $p(\varnothing) = (e, a)$. In general, let $(c_1, c_2) = p(X)$ for some $X \subseteq [n]$. Then $p([n]\backslash X) = (c_2, c_1)$. The facts that $p([n]) = (a, e)$, and $p(\varnothing) = (e, a)$ follow from the requirements that p must preserve C_1 and C_0 respectively. Let us show the second part of the claim. Observe that $(X, [n]\backslash X, \varnothing) \in R^{\boldsymbol{A}^n}$ so the image of this triple through p belongs to $R^{\boldsymbol{B}}$. Hence, there are elements $r_1, r_2, r_3 \in M$ witnessing Equation (1). It also holds that $p(\varnothing) = (e, a)$, so $r_3 = e$, and $p(X) = (r_1, r_2)$, $p([n]\backslash X) = (r_2, r_1)$, as desired.

Claim 3: Let $(b_1, b_2) = p(X)$ $(c_1, c_2) = p(Y)$ $(d_1, d_2) = p(X \cup Y)$ for two disjoint sets $X, Y \subseteq [n]$. Then $d_1 = b_1c_1$. Indeed, consider the set $Z = [n]\backslash(X \cup Y)$. By Claim 2 it holds that $p(Z) = (d_2, d_1)$. Moreover, it holds that $(X, Y, Z) \in R^{\boldsymbol{A}^n}$, so using the characterisation given in Equation (1) we obtain that $d_1 = b_1c_1$, as desired.

Claim 4: $\prod_{i=1^n} b_{p,i} = a$ Observe that $[n]$ is the disjoint union of the singleton sets $\{i\}$ for each $i \in [n]$. Using Claim 3 iteratively, we obtain that $\prod_{i=1^n} b_{p,i}$ equals the first element of the pair $p([n]) = (a, e)$, proving the claim.

Now we move on to proving (I), (II), and (III). Claims 1 and 4 show that for any n-ary polymorphism $p \in \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$, the tuple $(b_{p,i})_{i=0}^n$ is an element of $\mathscr{M}_{M,a}$, as stated in (I). As for fact (II), consider some n-ary polymorphism $p \in \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$, a map $\pi : [n] \to [m]$, and the minor $q = p^{(\pi)}$. We will now show that the fact that ξ preserves minors is equivalent to $b_{q,i} = \prod_{j \in \pi^{-1}(i)} b_{p,j}$ for all $i \in [m]$. Indeed, by definition, $b_{q,i}$ is the first element of the pair $q(\{i\}) = p(\pi^{-1}(i))$. However, expressing $\pi^{-1}(i)$ as a disjoint union of singletons and using Claim 3 we obtain that the first element of $p(\pi^{-1}(i))$ is the product of the first elements of the pairs $p(\{j\})$ for each $j \in \pi^{-1}(i)$, as we wanted to show.

So far we have established that ξ is indeed a minion homomorphism, following (I) and (II). Lastly, we prove that ξ is a bijection, as stated in (III). To see that ξ is surjective, consider a tuple $(b_1, \ldots, b_n) \in \mathscr{M}_{M,a}(n)$. Then the map $p : 2^n \mapsto \mathscr{M}_{M,a}(n)$ given by

$$X \mapsto (\prod_{i \in X} b_i, \prod_{i \in [n]\backslash X} b_i)$$

is a polymorphism of $(\boldsymbol{A}, \boldsymbol{B})$. Not only that but $\xi(p) = (b_1, \ldots, b_n)$. In the other direction, to prove that ξ is injective, we show that an n-ary polymorphism $p \in \mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ is completely determined by $(b_{p,1}, \ldots, b_{p,n})$. Consider an arbitrary set $X \subseteq [n]$, and let $p(X) = (c, d)$. Expressing X as a disjoint union of singletons and using Claim 3 we obtain that $c = \prod_{i \in X} b_{p,i}$. Additionally, by Claim 2 we know that $p([n]\backslash X) = (d, c)$ and using the same argument as before we obtain that $d = \prod_{i \in [n]\backslash X} b_{p,i}$. Thus, $p(X)$ is completely determined by the tuple $(b_{p,1}, \ldots, b_{p,n})$, showing that ξ is injective.

Acknowledgements

We thank the reviewers of the extended abstract [40] and in particular of this full version for comments, suggestions, and spotting several typos and mistakes.

A Reduction to Special Equations

By definition, instances of $\mathrm{Eqn}(S, T)$ can be seen as systems of equations over S where all constants belong to T. Any system of equations can be transformed into an equivalent system by adding new variables and breaking larger equations into smaller ones until every equation is of the form $x_1x_2 = x_3$, for three variables x_1, x_2, x_3, or of the form $x = t$ for a variable x and some constant $t \in T$. For instance, the equation

$$x_1c_1x_2c_2 \dots x_kc_k = c_{k+1},$$

where all $c_1, \dots, c_{k+1} \in T$ can be transformed into the system

$$\begin{aligned} x_1 y &= x, \qquad y = c_1, \\ xc_2 \dots x_k c_k &= c_{k+1}, \end{aligned}$$

where x, y are fresh variables. Applying these steps in succession yields a system where every equation is of the desired form.

When considering equations over groups the same idea works, but one needs to take into account inverted variables x^{-1}. Given a system of equations over a group G with constants in a subgroup $H \leq G$, we can substitute any instance of the inverted variable x^{-1} with instances of a fresh variable y after adding the equations $xy = z$, $z = e$ to the system, where z is another fresh variable.

B Dichotomies For Equations over Monoids and Groups

In this section we classify the complexity of $\mathrm{Eqn}(G)$ for a group G and $\mathrm{Eqn}(M)$ for a monoid M as corollaries of the Dichotomy Theorem for CSPs. These results where obtained previously in [29] and [35]. We begin by stating the Dichotomy Theorem. A n-ary map $p : A^n \to A$ is called *cyclic* if $p(a_1, a_2, \dots, a_n) = p(a_n, a_1, \dots, a_{n-1})$ for all $a_1, \dots, a_n \in A$.

Theorem 9 ([18, 20, 47])**.** *Let $\boldsymbol{A}$ be a finite relational structure. Then $\mathrm{CSP}(\boldsymbol{A})$ is tractable if $\mathrm{Pol}(\boldsymbol{A})$ contains a cyclic polymorphism of arity at least 2. Otherwise $\mathrm{CSP}(\boldsymbol{A})$ is NP-hard.*

Theorem 10. *Let G be a group. Then $\mathrm{Eqn}(G)$ has a cyclic polymorphism p of arity at least 2 if and only if G is Abelian.*

Proof. Suppose that G is Abelian. Let $n \geqslant 2$ be such that $g^n = g$ for all $g \in G$. Then the n-ary map $p : G^n \to G$ given by $(g_1, \dots, g_n) \mapsto \prod_{i=1}^n g_i$ is a cyclic polymorphism of $\mathrm{Eqn}(G)$.

In the other direction, suppose that $p : G^n \to G$ is a cyclic polymorphism of $\mathrm{Eqn}(G)$ and $n \geqslant 2$. Let $g_1, g_2 \in G$. As p is idempotent, we have that $g_1 = p(g_1, g_1, \dots, g_1)$. Using the fact that p is a group homomorphism and is cyclic, we obtain that

$$p(g_1, g_1, \dots, g_1) = p(g_1, e, \dots, e)p(e, g_1, \dots, e) \cdots p(e, e, \dots, g_1) = p(g_1^n, e, \dots, e).$$

Similarly, we can obtain that $g_2 = p(e, g_2^n, \dots, e)$. This way,

$$g_1g_2 = p(g_1^n, g_2^n, \dots, e) = g_2g_1.$$

As our initial choice of g_1, g_2 was arbitrary, this proves that G is Abelian. $\square$

Theorem 11. *Let M be a monoid. Then $\mathrm{Eqn}(M)$ has a cyclic polymorphism p of arity at least 2 if and only if M is Abelian and regular.*

Proof. In one direction, if M is regular then, by the second item in Lemma 1 there is some $n \geqslant 2$ satisfying $g^n = g$ for all $g \in M$. This way, we can define a cyclic polymorphism of $\mathrm{Eqn}(M)$ exactly as in the proof of Theorem 10.

In the other direction, let $p : M^n \to M$ be a n-ary cyclic polymorphism of $\mathrm{Eqn}(M)$, where $n \geqslant 2$. By the same arguments as in the the proof of Theorem 10, M must be Abelian. Let us show now that M is regular. Let $\psi : M \to M$ be the homomorphism given by $g \mapsto p(g, e, \ldots, e)$. Using the fact that p is cyclic and idempotent it must hold that $g = \psi(g)^n$. This shows that ψ is a bijection and that $\psi(g) \sqsupseteq g$ for all $g \in M$. As M is finite, the only way this is possible is that $\psi(g) \sim g$ for all g. However, from $\psi(g)^n = g$ and $n \geqslant 2$ we deduce that $\psi(g) \sim \psi(g)^2$, so $\psi(g)$ is a regular element by the fourth item in Lemma 1. This holds for an arbitrary g, so every element in M is regular. □

References

[1] Kristina Asimi and Libor Barto. Finitely tractable promise constraint satisfaction problems. In *Proc. 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21)*, volume 202 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. arXiv:2010.04618, doi:10.4230/LIPIcs.MFCS.2021.11.

[2] Albert Atserias and Víctor Dalmau. Promise constraint satisfaction and width. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1129–1153, 2022. arXiv:2107.05886, doi:10.1137/1.9781611977073.48.

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

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

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

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

[7] Amey Bhangale and Subhash Khot. Optimal Inapproximability of Satisfiable k-LIN over Non-Abelian Groups. In *Proc. 53rd Annual ACM Symposium on Theory of Computing (STOC'21)*, pages 1615–1628. ACM, 2021. arXiv:2009.02815, doi:10.1145/3406325.3451003.

[8] Amey Bhangale, Subhash Khot, and Dor Minzer. On Approximability of Satisfiable k-CSPs: II. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*, pages 632–642. ACM, 2023. doi:10.1145/3564246.3585120.

[9] Amey Bhangale, Subhash Khot, and Dor Minzer. On Approximability of Satisfiable k-CSPs: III. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*, pages 643–655. ACM, 2023. doi:10.1145/3564246.3585121.

[10] Manuel Bodirsky. *Complexity of infinite-domain constraint satisfaction*, volume 52. Cambridge University Press, 2021.

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

[12] Manuel Bodirsky and Thomas Quinn-Gregson. Solving equation systems in ω-categorical algebras. *J. Math. Log.*, 21(3), 2021. doi:10.1142/S0219061321500203.

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

[14] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. Conditional Dichotomy of Boolean Ordered Promise CSPs. *TheoretiCS*, 2, 2023. arXiv:2102.11854, doi:10.46298/theoretics.23.2.

[15] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. SDPs and robust satisfiability of promise CSP. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*, pages 609–622. ACM, 2023. arXiv:2211.08373, doi:10.1145/3564246.3585180.

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

[17] Alex Brandts and Stanislav Živný. Beyond PCSP(1-in-3,NAE). *Inf. Comput.*, 2022. arXiv:2104.12800, doi:10.1016/j.ic.2022.104954.

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

[19] Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24:1–24:66, 2011. doi:10.1145/1970398.1970400.

[20] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330, 2017. arXiv:1703.03021, doi:10.1109/FOCS.2017.37.

[21] Lorenzo Ciardo and Stanislav Živný. Approximate graph colouring and the hollow shadow. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*, pages 623–631. ACM, 2023. arXiv:2211.03168, doi:10.1145/3564246.3585112.

[22] Lorenzo Ciardo and Stanislav Živný. CLAP: A new algorithm for promise CSPs. *SIAM J. Comput.*, 52(1):1–37, 2023. arXiv:2107.05018, doi:10.1137/22M1476435.

[23] Lorenzo Ciardo and Stanislav Živný. Hierarchies of minion tests for PCSPs through tensors. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*, pages 568–580, 2023. arXiv:2207.02277, doi:10.1137/1.9781611977554.ch25.

[24] Víctor Dalmau and Jakub Opršal. Local consistency as a reduction between constraint satisfaction problems. In *Proc. 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'24)*, pages 29:1–29:15. ACM, 2024. arXiv:2301.05084, doi:10.1145/3661814.3662068.

[25] Lars Engebretsen, Jonas Holmerin, and Alexander Russell. Inapproximability results for equations over finite groups. *Theor. Comput. Sci.*, 312(1):17–45, 2004. doi:10.1016/S0304-3975(03)00401-8.

[26] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

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

[28] M. R. Garey and David S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976. doi:10.1145/321921.321926.

[29] Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Inf. Comput.*, 178(1):253–262, 2002. doi:10.1006/INCO.2002.3173.

[30] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1–24, 2007. doi:10.1145/1206035.1206036.

[31] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

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

[33] John M Howie. *Fundamentals of semigroup theory*. Oxford University Press, 1995.

[34] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997. doi:10.1145/263867.263489.

[35] Ondřej Klíma, Pascal Tesson, and Denis Thérien. Dichotomies in the complexity of solving systems of equations over finite semigroups. *Theory Comput. Syst.*, 40(3):263–297, 2007. doi:10.1007/S00224-005-1279-2.

[36] Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61(2):302–332, 2000. doi:10.1006/jcss.2000.1713.

[37] Michael Kompatscher. The equation solvability problem over supernilpotent algebras with Mal'cev term. *Int. J. Algebra Comput.*, 28(06):1005–1015, 2018. doi:10.1142/S0218196718500443.

[38] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. Topology and adjunction in promise constraint satisfaction. *SIAM J. Comput.*, 52(1):37–79, 2023. arXiv:2003.11351, doi:10.1137/20M1378223.

[39] Benoît Larose and László Zádori. Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. *Int. J. Algebra Comput.*, 16(3):563–582, 2006. doi:10.1142/S0218196706003116.

[40] Alberto Larrauri and Stanislav Živný. Solving promise equations over monoids and groups. In *Proc. 51st International Colloquium on Automata, Languages, and Programming (ICALP'24)*, volume 297 of *LIPIcs*, pages 146:1–146:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.146.

[41] Dániel Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM*, 60(6), 2013. Article No. 42. arXiv:0911.0801, doi:10.1145/2535926.

[42] Peter Mayr. On the complexity dichotomy for the satisfiability of systems of term equations over finite algebras. In *Proc. 48th International Symposium on Mathematical Foundations of Computer Science (MFCS'23)*, volume 272 of *LIPIcs*, pages 66:1–66:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.MFCS.2023.66.

[43] Tamio-Vesa Nakajima and Stanislav Živný. Linearly ordered colourings of hypergraphs. *ACM Trans. Comput. Theory*, 13(3–4), 2022. arXiv:2204.05628, doi:10.1145/3570909.

[44] Tamio-Vesa Nakajima and Stanislav Živný. On the complexity of symmetric vs. functional PCSPs. *ACM Trans. Algorithms*, 20(4):33:1–33:29, 2024. arXiv:2210.03343, doi:10.1145/3673655.

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

[46] Steve Seif and Csaba Szabó. Algebra complexity problems involving graph homomorphism, semigroups and the constraint satisfaction problem. *J. Complex.*, 19(2):153–160, 2003. doi:10.1016/S0885-064X(02)00027-4.

[47] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020. arXiv:1704.01914, doi:10.1145/3402029.