Solving promise equations over monoids and groups

Alberto Larrauri
Department of Computer Science, University of Oxford, UK

Stanislav Živný
Department of Computer Science, University of Oxford, UK

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.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Constraint and logic programming

Keywords and phrases constraint satisfaction, promise constraint satisfaction, equations, minions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.122

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full version*: `https://arxiv.org/abs/2402.08434`

Funding This research was funded in whole by UKRI EP/X024431/1. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

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 [34, 26], 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, many complexity dichotomies were established in restricted fragments of CSPs. This included conservative CSPs [19], or equations over finite algebraic structures such as semigroups, groups, and monoids [29, 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].

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

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson; Article No. 122; pp. 122:1–122:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One of the recent research directions in constraint satisfaction that has attracted a lot of attention is the area of *promise* CSPs (PCSPs) [3, 13, 5]. 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., [27, 1, 4, 17, 43, 44, 6, 2, 21, 14, 15, 24, 22], 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 is the "the algebraic approach to PCSPs" [5]. The influential paper [5] identified *minions* as an important concept. Minions generalise the notion of "a family of functions that is closed under permuting arguments, identifying arguments, and adding dummy arguments". A crucial example is the polymorphism minion of a PCSP template. Polymorphisms can be seen as high-dimensional symmetries of a PCSP template and capture the complexity of the underlying computational problem [13, 5]. Following the algebraic approach [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, 16, 22, 15].

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 14 in Section 3.

All our tractability results use solvability via the BLP + AIP algorithm [16]. In fact, tractable PCSPs corresponding to promise systems of equations over monoids are finitely tractable in the sense of [13, 1]. In the special case of promise systems of equations over groups, the affine integer programming (AIP) algorithm [13, 5] 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, 8, 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 [25, 7]. Systems of equations have been studied, e.g., over semigroups in [46], over monoids and semigroups in [35], and over arbitrary finite algebras in [39, 37, 12, 42].

The full version of this paper [40] contains all details and proofs.

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 *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 can be expressed as a product of elements in S including t. Note that $\sqsubseteq$ constitutes a preorder over any semigroup. 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. We call an element s of S *regular* if $s^2 t = s$ and $ts = st$ for some t in S.[3] 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 use the standard product (and also the power) of a semigroup (monoid, group), where the operation is defined componentwise. We use the symbol $\leq$ for a substructure; e.g., if S is a semigroup then we write $T \leq S$ to indicate that T is a subsemigroup of S (and similarly for monoids and groups).

[2] I.e., the multiplication on the LHS is in S_1, whereas the multiplication on the RHS is in S_2.
[3] The usual definition of a regular element in a semigroup just requires that $sts = s$ for some t [33].

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

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 componentwise. 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 $(\boldsymbol{A}, \boldsymbol{B})$.[4]

Minions

A *minion* $\mathcal{M}$ consists of a set $\mathcal{M}(n)$ for each positive number n, and a map $\pi^{\mathcal{M}} : \mathcal{M}(n) \to \mathcal{M}(m)$ for each map $\pi : n \to m$ satisfying (1) $\mathrm{id}_{[n]}^{\mathcal{M}} = \mathrm{id}_{\mathcal{M}(n)}$ for every $n > 0$, and (2) $\pi^{\mathcal{M}} \circ \tau^{\mathcal{M}} = (\pi \circ \tau)^{\mathcal{M}}$ for every pair of suitable maps π, τ. When the minion is clear from the context, we write $p^{(\pi)}$ for $\pi^{\mathcal{M}}(p)$. Elements $p \in \mathcal{M}(n)$ are called n-ary, and whenever $p^{(\pi)} = q$ we say that q is a *minor* of p. A *minion homomorphism* $\xi : \mathcal{M} \to \mathcal{N}$ is a map from a minion $\mathcal{M}$ to another minion $\mathcal{N}$ that preserves arities and minor operations. I.e., $\xi(p^{(\pi)}) = (\xi(p))^{(\pi)}$ for every minor $p^{(\pi)}$.

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. That is, n-ary elements of $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B})$ are just n-ary polymorphisms $p : A^n \to B$. Additionally, given some n-ary polymorphism p, and some 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 $\mathcal{M}$, we define two special types of elements. An element $p \in \mathcal{M}(2k+1)$ 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 $p \in \mathcal{M}(2k+1)$ 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

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

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 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 [40].

▶ **Definition 3.** *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 abuse the notation and write $\mathrm{Eqn}(S, T)$ for $\mathrm{CSP}(\mathrm{Eqn}(S, T))$.

Previous works focused on problems $\mathrm{Eqn}(G, G)$ and $\mathrm{Eqn}(M, M)$. Given a group G, it is known that $\mathrm{Eqn}(G, G)$ is solvable in polynomial time (by AIP) if G is Abelian, and NP-hard otherwise [29]. Similarly, when M is a monoid, $\mathrm{Eqn}(M, M)$ is solvable in polynomial time if M is Abelian and it is the union of its subgroups, and NP-hard otherwise [35].

We now define promise equations.

▸ **Definition 4.** *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 $\mathrm{PEqn}(S, S, \mathrm{id}_T)$.

We can now state our main result.

▸ **Theorem 5** (Main). $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is solvable in polynomial time by BLP + AIP if there is an Abelian homomorphism $\psi : M_1 \to M_2$ extending φ and $\mathrm{Im}(\psi)$ is a union of subgroups. Otherwise,* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is* NP*-hard.*

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

▸ **Corollary 6.** $\mathrm{PEqn}(G_1, G_2, \varphi)$ *is solvable in polynomial time via AIP if there is an Abelian homomorphism $\psi : G_1 \to G_2$ extending φ. Otherwise,* $\mathrm{PEqn}(G_1, G_2, \varphi)$ *is* NP*-hard.*

As easy corollaries, Theorem 5 applies in the special case of non-promise setting.

▸ **Corollary 7.** *Given two monoids $N \leq M$,* $\mathrm{Eqn}(M, N)$ *is solvable in polynomial time by BLP + AIP if there is an Abelian endomorphism of M extending* id_N *whose image is a union of subgroups, and is* NP*-hard otherwise.*

▸ **Corollary 8.** *Given two groups $H \leq G$,* $\mathrm{Eqn}(G, H)$ *is solvable in polynomial time by AIP if there is an Abelian endomorphism of G that extends* id_H*, and is* NP*-hard otherwise.*

▸ **Example 9.** 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 homomorphism 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 via 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 $Z_2 \times Z_2$) 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 $\{e, r, r^2, r^3\} \geq \mathrm{dom}(\varphi_1)$, so $\mathrm{dom}(\varphi_1)$ 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 5 are in fact finitely tractable; for details, cf. [40].

The power of BLP + AIP is necessary in Theorem 5 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; for details, cf. [40].

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 10.** *Let $(\boldsymbol{A}, \boldsymbol{B})$ be a template. Then there are semigroups S_1, S_2 and a semigroup homomorphism φ with $\mathrm{dom}(\varphi) \leq S_1$ and $\mathrm{Im}(\varphi) \leq S_2$ 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 $\mathcal{M}$, there are PCSP templates whose polymorphism minions are isomorphic to $\mathcal{M}$. 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 11.** *Given an element $a \in M$ the* monoidal minion *$\mathcal{M}_{M,a}$ is the one where for each $n \in \mathbb{N}$ the elements $\boldsymbol{b} \in \mathcal{M}_{M,a}(n)$ are commutative tuples $\boldsymbol{b} \in M^n$ with $\prod_{i \in [n]} b_i = a$, and where for each $m > 0$ 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 12.** *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 $\mathcal{M}_{M,a}$. If a is regular in M then $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is solvable in polynomial time by BLP + AIP. Otherwise, $\mathrm{PCSP}(\boldsymbol{A}, \boldsymbol{B})$ is NP-complete.*

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

▸ **Theorem 13.** *Let M be a monoid, and $a \in M$ an arbitrary element. Then the template $(\boldsymbol{A}, \boldsymbol{B})$ described below satisfies that $\mathrm{Pol}(\boldsymbol{A}, \boldsymbol{B}) \simeq \mathcal{M}_{M,a}$.*[5] *The signature σ of $\boldsymbol{A}$ and*

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

$\boldsymbol{B}$ *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* $\mathcal{M}_{M,a}(2)$*. We define* $R^{\boldsymbol{B}}$ *as the set of triples in* $(\mathcal{M}_{M,a}(2))^3$ *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.*[6]

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

▸ Remark 14. 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 $\mathcal{T}$ 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 $\mathcal{T}$ [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.[7]

If we allow infinite monoids to be considered, then monoidal minions include important minions that capture solvability via relevant algorithms. Consider the monoid $M = \{(r,z) \in \mathbb{Q}\times\mathbb{Z} \mid r \in [0,1], \text{ and } r = 0 \text{ implies } z = 0\}$, where the binary operation is given by coordinate-wise addition, and the identity is $(0,0)$. Then $\mathcal{M}_{M,(1,1)}$ is precisely the minion $\mathcal{M}_{\text{BLP+AIP}}$ 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 12

Tractability We use the characterisation of the power of BLP + AIP from Theorem 2 for the tractability part of Theorem 12. Observe that if there is a minion homomorphism $\xi : \mathcal{M}_{M,a} \to \text{Pol}(\boldsymbol{A},\boldsymbol{B})$ and $p \in \mathcal{M}_{M,a}$ is a $(2k+1)$-ary 2-block-symmetric element, then so is $\xi(p)$. Hence, showing that $\mathcal{M}_{M,a}$ has 2-block-symmetric elements of all arities proves that PCSP$(\boldsymbol{A},\boldsymbol{B})$ is solvable in polynomial time via BLP + AIP. Let $b \in M$ witness that a is regular. For each $k > 0$ consider the $(2k+1)$-ary element of $\mathcal{M}_{M,a}$ consisting of $k+1$ consecutive a's followed by k consecutive b's. To see that this this is indeed an element of $\mathcal{M}_{M,a}$ we need to check that $a^{k+1}b^k = a$. This follows from the assumption that b witnesses that a is regular and using $a^{k+1}b^k = a(ba)^k$. This tuple is 2-block-symmetric, with the blocks corresponding to a and b (of sizes k and $k+1$, respectively).

Intractability We prove the intractability part of Theorem 12 (as well as other hardness results later in this paper) using the following result.

▸ **Theorem 15** ([5]). *Let* $\mathcal{M} = \text{Pol}(\boldsymbol{A},\boldsymbol{B})$*, and let* $K, L > 0$ *be any fixed integers. Suppose that* M *satisfies the following condition:*

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

---

[6] 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 $\mathcal{M}_{M,a}$ generated by $\boldsymbol{A}$.

[7] We thank Lorenzo Ciardo for this observation.

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

Given a template $(\boldsymbol{A},\boldsymbol{B})$, if there is a minion homomorphism $\xi : \mathrm{Pol}(\boldsymbol{A},\boldsymbol{B}) \to \mathcal{M}_{M,a}$ and $\mathcal{M}_{M,a}$ satisfies the condition in the previous theorem, so does $\mathrm{Pol}(\boldsymbol{A},\boldsymbol{B})$. Indeed, if $\mathcal{M}_{M,a} = \bigcup_{\ell\in[L]} \mathcal{M}_\ell$, then we can write $\mathrm{Pol}(\boldsymbol{A},\boldsymbol{B}) = \bigcup_{\ell\in[L]} \xi^{-1}(\mathcal{M}_\ell)$. Additionally, if the map $\mathcal{I}_\ell$ witnesses the condition in the theorem for $\mathcal{M}_\ell$, then the map $\mathcal{I}'_\ell$ given by $p \mapsto \mathcal{I}_\ell(\xi(p))$ witnesses the same condition for $\xi^{-1}(\mathcal{M}_\ell)$. Hence, we show the hardness part of Theorem 12 by proving that $\mathcal{M}_{M,a}$ satisfies the assumptions in Theorem 15 when $a \in M$ is not regular.

For a monoid M, we define a refinement $\sqsubseteq^{\mathrm{A}}$ of the preorder $\sqsubseteq$ introduced in Section 2. In detail, we write $a \sqsubseteq^{\mathrm{A}} b$ for $a, b \in M$ if there is a third element $c \in M$ that commutes with b such that $bc = a$. We put $a \sim^{\mathrm{A}} b$ when both $a \sqsubseteq^{\mathrm{A}} b$ and $b \sqsubseteq^{\mathrm{A}} a$ hold, and $a \sqsubsetneq^{\mathrm{A}} b$ when $a \sqsubseteq^{\mathrm{A}} b$ holds but $b \sqsubseteq^{\mathrm{A}} a$ does not. We use the following simple observation.

▶ **Observation 16.** *Let M be a monoid and $a, b, c \in M$ be three elements that commute pairwise. Suppose that $abc \sqsubsetneq^{\mathrm{A}} ab$. Then $ac \sqsubsetneq^{\mathrm{A}} a$.*

Proof. We prove the contrapositive. Suppose that $a \sqsubseteq^{\mathrm{A}} ac$. That is, there is some $d \in M$ that commutes with ac and satisfies $acd = a$. We have $dabc = (dac)b = (acd)b = ab$ $abcd = c(abd) = ca = ac$, and thus $dabc = abcd = ab$, proving that $abc \sqsupseteq^{\mathrm{A}} ab$. ◀

Assume that a is not regular. That is, that $a^2 b \neq a$ for every $b \in M$ that commutes with a. Let $\boldsymbol{b} \in \mathcal{M}_{M,a}(n)$ for some number $n > 0$. A coordinate $j \in [n]$ is called *relevant* in $\boldsymbol{b}$ if $\prod_{i\in[m]\setminus j} b_i \sqsupsetneq^{\mathrm{A}} \prod_{i\in[n]} b_i$. Consider the map $\mathcal{I}$ that assigns to each $\boldsymbol{b} \in \mathcal{M}_{M,a}$ its set of relevant coordinates. Claims 1 through 3 proved below establish the required assumptions in Theorem 15 with $L = 1$ and $K = |m|$, thus showing NP-hardness of PCSP$(\boldsymbol{A},\boldsymbol{B})$.

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

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

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 $d_k c_k b_{i_k} \sqsubsetneq^{\mathrm{A}} d_k c_k$. Applying Observation 16, we obtain that $c_k b_{i_k} \sqsubsetneq^{\mathrm{A}} c_k$. Expanding the definition of c_k this means that

$$\prod_{j\in[k]} b_{i_j} \sqsubsetneq^{\mathrm{A}} \prod_{j\in[k-1]} b_{i_j}.$$

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 was not the case we would have that

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

However, $i \in \pi^{-1}(j)$, so we know that $\prod_{k\in[n]\setminus\{i\}} b_k \sqsubseteq^{\mathrm{A}} \prod_{k\in[n]\setminus\pi^{-1}(j)} b_k$. Putting this together with the previous identity shows that

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

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

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

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

5 Equations Over Monoids and Groups: Proofs of Theorem 5 and Corollary 6

We need a simple characterisation of the polymorphisms of promise equation templates, and various characterisations of regularity; both are proved in [40].

▸ **Lemma 17.** *Consider a template* $\mathrm{PEqn}(Z_1, Z_2, \varphi)$ *of promise equations over semigroups/monoids/groups. 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/group homomorphism and* $p(s, s, \dots, s) = \varphi(s)$ *for all* $s \in \mathrm{dom}(\varphi)$.

▸ **Lemma 18.** *Let* M *be a monoid and* $s \in M$. *Then the following are equivalent:*
1. s *is regular,*
2. $s^k = s$ *for some* $k > 1$,
3. s *belongs to a subgroup of* M,
4. $s \sqsubseteq s^2$.

With these two lemmas, we can now prove our main result.

▸ **Theorem 5** (Main). $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is solvable in polynomial time by BLP + AIP if there is an Abelian homomorphism* $\psi : M_1 \to M_2$ *extending* φ *and* $\mathrm{Im}(\psi)$ *is a union of subgroups. Otherwise,* $\mathrm{PEqn}(M_1, M_2, \varphi)$ *is* NP*-hard.*

Proof. We prove both implications. Suppose that such homomorphism ψ exists. As $\mathrm{Im}(\psi)$ is a union of subgroups, by Lemma 18 there is some number $k > 1$ such that $s^k = s$ for all $s \in \mathrm{Im}(\psi)$. Let $n > 0$ 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 17. First, observe that the fact that ψ is Abelian implies that p is a monoid homomorphism. Indeed,

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

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

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

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, 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 $\mathcal{M}$ be the polymorphism minion of $\mathrm{PEqn}(M_1, M_2, \varphi)$. Given a polymorphism $p \in \mathcal{M}$, 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 18, 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 15 with the constants L, K to show NP-hardness. We define the following subminions of $\mathcal{M}$.

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

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

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

By the previous observation it holds that

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

We give selection functions $\mathcal{I}$ for each of these sub-minions satisfying the assumptions of Theorem 15. Suppose that $\mathcal{M}_{\mathrm{A}}$ is not empty. Otherwise we are done defining $\mathcal{I}_{\mathrm{A}}$. Let p be any n-ary polymorphism in $\mathcal{M}_{\mathrm{A}}$. 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.

Fact 1: The map $\phi : \prod_{i\in[n]} \mathcal{N}(p, i) \to M_2$ given by $(s_1, \dots s_n) \mapsto \prod_{i\in[n]} s_i$ 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.

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

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

Given an n-ary polymorphism $p \in \mathcal{M}_{\mathrm{A}}$, 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 15. Indeed, given some n-ary p:

- $\mathcal{I}_{\mathrm{A}}(p)$ is non empty by Fact 3.
- $|\mathcal{I}_{\mathrm{A}}(p)| \leqslant K$. 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 Fact 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 $\mathcal{M}_\psi$ is non-empty. We define a selection function $\mathcal{I}_\psi$ satisfying the assumptions of Theorem 15. Let $t \in \mathrm{Im}(\psi)$ be a non-regular element, and let $s \in M_1$ be such that $\psi(s) = t$. Let $\mathcal{M}_{M_2,t}$ be the monoidal minion defined in Definition 11. Consider the map $\xi : \mathcal{M}_\psi \to \mathcal{M}_{M_2,a}$ that sends any n-ary polymorphism $p \in \mathcal{M}_\psi$ to the tuple $(r_1, \dots, r_n) \in \mathcal{M}_{M_2,a}(n)$ 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 $\mathcal{M}_\psi$ to $\mathcal{M}_{M_2,t}$. Indeed, first observe 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 Fact 1 above. One can also check that ξ preserves minors.

From the proof of Theorem 12 there is some selection function $\mathcal{I}$ on $\mathcal{M}_{M_2,t}$ satisfying the hypotheses of Theorem 15 for some constant $K' = |M_2| \leqslant K$ and $L = 1$. Thus, we can define $\mathcal{I}_\psi$ on $\mathcal{M}_\psi$ simply by setting $\mathcal{I}_\psi(p) = \mathcal{I}(\xi(p))$ for each polymorphism $p \in \mathcal{M}_\psi$.

Hence, have defined selection functions $\mathcal{I}_{\mathrm{A}}$ and $\mathcal{I}_\psi$ for each $\psi \in \Omega$ that satisfy the requirements of Theorem 15, showing that $\mathrm{PEqn}(M_1, M_2, \varphi)$ is NP-hard. ◂

▸ **Corollary 6.** $\mathrm{PEqn}(G_1, G_2, \varphi)$ *is solvable in polynomial time via AIP if there is an Abelian homomorphism* $\psi : G_1 \to G_2$ *extending* φ*. Otherwise,* $\mathrm{PEqn}(G_1, G_2, \varphi)$ *is* NP*-hard.*

Proof. We prove both directions. The hardness case follows from Theorem 5. 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 5, $\mathrm{PEqn}(G_1, G_2, \varphi)$ 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 $\mathrm{PEqn}(G_1, G_2, \varphi)$. ◂

6 Equations over Semigroups: Proof of Theorem 10

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

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 19** ([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 20.** *Let $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ be a template of digraphs. Then $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ is polynomial-time 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 $\mathrm{PCSP}(\boldsymbol{D}_1^+, \boldsymbol{D}_2^+)$. 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 removes the ones that do. If the resulting structure $\boldsymbol{I}'$ is empty, then our reduction accepts $\boldsymbol{I}$. Otherwise, observe that 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 is easy to see 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 21** ([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 22** ([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 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 23.** *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 $s \sim s'$ and S is right-normal, 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 $E^{\mathrm{C}} = \{(u, v)^{\mathrm{C}} \mid (u, v) \in E^D\} \cup \{(p, q)^{\mathrm{C}}\}$. Finally, the class 0 contains a single element 0. By Observation 23, 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.

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 $\widehat{S_W} = S_W/\sim$.

▸ **Lemma 24.** *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

$v^{\mathrm{L}}, v^{\mathrm{R}}, v^{\mathrm{LR}}$. For each $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LR}\}$ 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 equations $p^{\mathrm{LR}} v^{\mathrm{L}} = v^{\mathrm{LR}}$ and $p^{\mathrm{LR}} v^{\mathrm{R}} = v^{\mathrm{LR}}$. If $v \in P^{\boldsymbol{I}}$ we include all constraints $v^\square = p^\square$ for $\square \in \{\mathrm{L}, \mathrm{R}, \mathrm{LR}\}$. Similarly, if $v \in Q^{\boldsymbol{I}}$, then we include the constraints of the form $v^\square = q^\square$. 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. ◂

▸ **Lemma 25.** *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 any 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 24 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 $x_1 x_2 = x_3$, 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 21 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 22, 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 23 it holds that $c_{x_1} c_{x_2} x_1 = c_{x_3} x_1$.

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 $c_1x = c_1y$, 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 $x \in V^{\mathrm{CR}}$.

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.

Consider an equation of the form $cx = cy$, where both x, y are constrained to be in c's $\sim$-class. $(p, q)^{\mathrm{C}}x = (p, q)^{\mathrm{C}}y$, both 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}. Identifying variables and adding dummy variables if necessary we can assume the following hold:

- 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 $p^{\mathrm{CR}}x = p^{\mathrm{CR}}x_{\mathrm{R}}$.
- There are no two variables $x, y \in X$ constrained by $x, y \in E^{\mathrm{C}}$ with $x_{\mathrm{L}} = y_{\mathrm{L}}$ and $x_{\mathrm{R}} = y_{\mathrm{L}}$.
- 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 24. ◂

▸ **Corollary 26.** *Let $(\boldsymbol{D}_1, \boldsymbol{D}_2)$ be a template of digraphs. Then* $\mathrm{PCSP}(\boldsymbol{D}_1, \boldsymbol{D}_2)$ *is polynomial-time 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 20. Observe that algorithm Φ given in Lemma 25 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 24 is a polynomial-time Turing reduction in the other direction. ◂

Corollary 26 and Theorem 19 establish Theorem 10.

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). *Information and Computation*, 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 Victor Dalmau and Jakub Opršal. Local consistency as a reduction between constraint satisfaction problems. Technical report, 2023. arXiv:2301.05084.

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. *International Journal of Algebra and Computation*, 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. Technical report, 2024. arXiv:2402.08434.

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 Zivný. Boolean symmetric vs. functional PCSP dichotomy. In *Proc. 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'23)*, 2023. `doi:10.1109/LICS56636.2023.10175746`.

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`.