CSPs with Few Alien Constraints

Peter Jonsson ✉
Department of Computer and Information Science, Linköping University, Sweden

Victor Lagerkvist ✉
Department of Computer and Information Science, Linköping University, Sweden

George Osipov ✉
Department of Computer and Information Science, Linköping University, Sweden

Abstract

The *constraint satisfaction problem* asks to decide if a set of constraints over a relational structure $\mathcal{A}$ is satisfiable (CSP($\mathcal{A}$)). We consider CSP($\mathcal{A} \cup \mathcal{B}$) where $\mathcal{A}$ is a structure and $\mathcal{B}$ is an *alien* structure, and analyse its (parameterized) complexity when at most k alien constraints are allowed. We establish connections and obtain transferable complexity results to several well-studied problems that previously escaped classification attempts. Our novel approach, utilizing logical and algebraic methods, yields an FPT versus pNP dichotomy for arbitrary finite structures and sharper dichotomies for Boolean structures and first-order reducts of $(\mathbb{N}, =)$ (equality CSPs), together with many partial results for general ω-categorical structures.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Mathematics of computing → Discrete mathematics

Keywords and phrases Constraint satisfaction, parameterized complexity, hybrid theories

Digital Object Identifier 10.4230/LIPIcs.CP.2024.15

Funding *Peter Jonsson and George Osipov*: Supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
Peter Jonsson: Partially supported by the Swedish research council under grant VR-2021-04371.
Victor Lagerkvist: Partially supported by the Swedish research council under grant VR-2022-03214.

Acknowledgements We thank the anonymous reviewers for helpful suggestions for how to improve the presentation of the paper.

1 Introduction

The *constraint satisfaction problem* over a structure $\mathcal{A}$ (CSP($\mathcal{A}$)) is the problem of verifying whether a set of constraints over $\mathcal{A}$ admits at least one solution. This problem framework is vast, and, just to name a few, include all Boolean satisfiability problems as well as k-coloring problems, and for infinite domains we may formulate both problems centrally related to model checking first-order formulas and qualitative reasoning. Notable examples where complete complexity dichotomies are known (separating tractable from NP-hard problems) include *all* finite structures [12, 24] and first-order definable relations over well-behaved base structures like $(\mathbb{N}, =)$ and $(\mathbb{Q}, <)$ [2]. While impressive mathematical achievements, these dichotomy results are still somewhat unsatisfactory from a practical perspective since we are unlikely to encounter instances which are based on *purely* tractable constraints. Could it be possible to extend the reach of these powerful theoretical results by relaxing the basic setting so that we may allow greater flexibility than purely tractable constraints while still obtaining something simpler than an arbitrary NP-hard CSP?

We consider this problem in a *hybrid* setting via problems of the form CSP($\mathcal{A} \cup \mathcal{B}$) where $\mathcal{A}$ is a "stable", tractable background structure and $\mathcal{B}$ is an *alien* structure. We focus on the case when CSP($\mathcal{A} \cup \mathcal{B}$) is NP-hard (thus, richer than a polynomial-time solvable problem) but where we have comparably few constraints from the alien structure $\mathcal{B}$. This problem is compatible with the influential framework of *parameterized complexity* which has been used with great effect to study *structurally* restricted problems (e.g., based on tree-width) but where comparably little is known when one simultaneously restricts the allowed constraints.

We begin (in Section 3) by relating the CSP problem with alien constraints to other problems, namely, (1) *model checking*, (2) the problem of checking whether a constraint in a CSP instance is *redundant*, (3) the *implication* problem and (4) the *equivalence* problem. We prove that the latter three problems are equivalent under Turing reductions and provide a general method for obtaining complexity dichotomies for all of these problems via a complexity dichotomy for the CSP problem with alien constraints. Importantly, all of these problems are well-known in their own right, but have traditionally been studied with wildly disparate tools and techniques, but by viewing them under the unifying lens of alien constraints we not only get four dichotomies for the price of one but also open the powerful toolbox based on *universal algebra*. For non-Boolean domains this is not only a simplifying aspect but an absolute necessity to obtain general results. We expand upon the algebraic approach in Section 4 and relate alien constraints to *primitive positive definitions* (pp-definitions) and the important notion of a *core*. As a second general contribution we explore the case when each relation in $\mathcal{B}$ can be defined via an *existential positive formula* over $\mathcal{A}$. This results in a general *fixed-parameter tractable* (FPT) algorithm (with respect to the number of alien constraints) applicable to both finite, and, as we demonstrate later, many natural classes of structures over infinite domains.

In the second half of the paper we attack the complexity of alien constraints more systematically. We begin with structures over finite domains where we obtain a general tractability result by combining the aforementioned FPT algorithm together with the CSP dichotomy theorem [12, 24]. In a similar vein we obtain a general hardness result based on a universal algebraic gadget. Put together this yields a general result: if $\mathcal{A} \cup \mathcal{B}$ is a core (which we may assume without loss of generality) then either $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is FPT, or $\mathrm{CSP}_{\leq p}(\mathcal{A} \cup \mathcal{B})$ is NP-hard for some $p \geq 0$, i.e., is *para-NP-hard* (pNP-hard). Thus, from a parameterized complexity view we obtain a complete dichotomy (FPT versus pNP-hardness) for finite-domain structures. However, to also obtain dichotomies for implication, equivalence, and the redundancy problem, we need sharper bounds on the parameter p. We concentrate on two special cases. We begin with Boolean structures in Section 5.2 and obtain a complete classification which e.g. states that $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is FPT if $\mathcal{A}$ is in one of the classical *Schaefer* classes, and give a precise characterization of $\mathrm{CSP}_{\leq p}(\mathcal{A} \cup \mathcal{B})$ for all relevant values of p if $\mathcal{A}$ is not Schaefer. For example, if we assume that $\mathcal{A}$ is Horn, we may thus conclude that $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is FPT for *any* alien Boolean structure $\mathcal{B}$. More generally this dichotomy is sufficiently sharp to also yield dichotomies for implication, equivalence, and redundancy. Compared to the proofs by Schnoor & Schnoor [22] for implication and Böhler [11] for equivalence, we do not use an exhaustive case analysis over Post's lattice.

In Section 6 we consider structures over infinite domains. If we assume that $\mathcal{A}$ and $\mathcal{B}$ are ω-categorical, then we manage to lift the FPT algorithm based on existential positive definability from Section 4 to the infinite setting. Another important distinction is that the notion of a core, and subsequently the common trick of singleton expansion, works differently for ω-categorical languages. Here we follow Bodirsky [2] and use the notion of a *model-complete core*, which means that all n-ary orbits are pp-definable, where an orbit is defined as the action of the automorphism group over a fixed n-ary tuple. This allows us to, for example, prove that $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is FPT whenever $\mathcal{A}$ is an ω-categorical model-complete core and $\mathrm{CSP}(\mathcal{A})$ is in P such that the orbits of the automorphism group of $\mathcal{B}$ are included in the orbits of the automorphism group of $\mathcal{A}$. This forms a cornerstone for the dichotomy for equality languages since the only remaining cases are when $\mathcal{A}$ is 0-valid (meaning that each relation contains a constant tuple) but not Horn (defined similarly to the Boolean domain), and when $\mathcal{B}$ is not 0-valid. The remaining cases are far from trivial, however, and we require the algebraic machinery from Bodirsky et al. [4] which provides a characterization of equality languages in terms of their *retraction* to finite domains. We rely on this description via a recent classification result by Osipov & Wahlström [19]. Importantly, our dichotomy result is sufficiently sharp to additionally obtain complexity dichotomies for the implication, equivalence, and redundancy problems. To the best of our knowledge, these dichotomies are the first of their kind for arbitrary equality languages.

We finish the paper with a comprehensive discussion in Section 7. Most importantly, we have opened up the possibility to systematically study not only alien constraints, but also related problems that have previously escaped complexity classifications. For future research the main open questions are whether (1) sharper results can be obtained for arbitrary finite domains and (2) which further classes of infinite domain structures should be considered.

Proofs of statements marked with $(\star)$ have been removed due to space constraints.

2 Preliminaries

We begin by introducing the basic terminology and the fundamental problems under consideration. We assume throughout the paper that the complexity classes P and NP are distinct. We let $\mathbb{Q}$ denote the rationals, $\mathbb{N} = \{0, 1, 2, \dots\}$ the natural numbers, $\mathbb{Z} = \{\dots, -2.-1, 0, 1, 2, \dots\}$ the integers, and $\mathbb{Z}_+ = \{1, 2, 3, \dots\}$ the positive integers. For every $c \in \mathbb{Z}_+$, we let $[c] = \{1, 2, \dots, c\}$.

A *parameterized problem* is a subset of $\Sigma^* \times \mathbb{N}$ where Σ is the input alphabet, i.e., an instance is given by $x \in \Sigma^*$ of size n and a natural number k, and the running time of an algorithm is studied with respect to both k and n. The most favourable complexity class is FPT (*fixed-parameter tractable*), which contains all problems that can be decided in $f(k) \cdot n^{O(1)}$ time with f being some computable function. An *fpt-reduction* from a parameterized problem $L_1 \subseteq \Sigma_1^* \times \mathbb{N}$ to $L_2 \subseteq \Sigma_2^* \times \mathbb{N}$ is a function $P : \Sigma_1^* \times \mathbb{N} \to \Sigma_2^* \times \mathbb{N}$ that preserves membership (i.e., $(x, k) \in L_1 \Leftrightarrow P((x, k)) \in L_2$), is computable in $f(k) \cdot |x|^{O(1)}$ time for some computable function f, and there exists a computable function g such that for all $(x, k) \in L_1$, if $(x', k') = P((x, k))$, then $k' \leq g(k)$. It is easy to verify that if L_1 and L_2 are parameterized problems such that L_1 fpt-reduces to L_2 and L_2 is in FPT, then it follows that L_1 is in FPT, too. There are many parameterized classes with less desirable running times than FPT but we focus on pNP-hard problems: a problem is pNP-hard under fpt-reductions if it is NP-hard for some constant parameter value, implying such problems are not in FPT unless $\mathsf{P} = \mathsf{NP}$.

We continue by defining *constraint satisfaction problems.* First, a *constraint language* is a (typically finite) set of relations $\mathcal{A}$ over a universe A, and for a relation $R \in \Gamma$ we write $\mathrm{ar}(R) = k$ to denote its arity k. It is sometimes convenient to associate a constraint language with a relational signature, and thus obtaining a *relational structure*: a tuple $(A; \tau, I)$ where A is the *domain*, or *universe*, τ is a relational signature, and I is a function from σ to the set of all relations over D which assigns each relation symbol R a corresponding relation $R^{\mathcal{A}}$ over D. We write $\mathrm{ar}(R)$ for the arity of a relation R, and if $R = \emptyset$ then $\mathrm{ar}(R) = 0$. All structures in this paper are relational and we assume that they have a finite signature unless otherwise stated. Typically, we do not need to make a sharp distinction between relations and the corresponding relation symbols, so we usually simply write $(A; R_1, \dots, R_m)$, where each R_i is a relation over A, to denote a structure. We also sometimes do not make a sharp distinction between structures and sets of relations when the signature is not important. For arbitrary structures $\mathcal{A}$ and $\mathcal{A}'$ with domains A and A', we let $\mathcal{A} \cup \mathcal{A}'$ denote the structure with domain $A \cup A'$ and containing the relations in $\mathcal{A}$ and $\mathcal{A}'$.

For a constraint language (or structure) $\mathcal{A}$ an instance of the *constraint satisfaction problem* over $\mathcal{A}$ (CSP($\mathcal{A}$)) is then given by $I = (V, C)$ where V is a set of variables and C a set of constraints of the form $R(x_1, \ldots, x_k)$ where $x_1, \ldots, x_k \in V$ and $R \in \mathcal{A}$, and the question is whether there exist a function $f\colon V \to A$ that satisfies all constraints (a *solution*), i.e., $(f(x_1), \ldots, f(x_k)) \in R$ for all $R(x_1, \ldots, x_k) \in C$. The *CSP dichotomy theorem* says that all finite-domain CSPs are either in P or are NP-complete [12, 24]. Given an instance $I = (V, C)$ of CSP($\mathcal{A}$), we let Sol(I) be the set of solutions to I. We now define CSPs with alien constraints in the style of Cohen et al. [14].

$\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$

Instance: A natural number k and an instance $I = (V, C_1 \cup C_2)$ of CSP($\mathcal{A} \cup \mathcal{B}$), where (V, C_1) is an instance of CSP($\mathcal{A}$) and (V, C_2) is an instance of CSP($\mathcal{B}$) with $|C_2| \leq k$.
Question: Does there exist a satisfying assignment to I?

Throughout the paper, we assume without loss of generality that the structures $\mathcal{A}$ and $\mathcal{B}$ can be associated with disjoint signatures. The parameter in $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is the *number of alien constraints* (abbreviated #ac). We let $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ denote the $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ problem restricted to a fixed value k of parameter #ac. Note that if CSP($\mathcal{A}$) is not in P, then $\mathrm{CSP}_{\leq 0}(\mathcal{A} \cup \mathcal{B})$ is not in P; moreover, if CSP($\mathcal{A} \cup \mathcal{B}$) is in P, then $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in P. Thus, it is sensible to always require that CSP($\mathcal{A}$) is in P and CSP($\mathcal{A} \cup \mathcal{B}$) is not in P. In many natural cases (e.g., all finite-domain CSPs), CSP($\mathcal{A} \cup \mathcal{B}$) not being polynomial-time solvable implies that CSP($\mathcal{A} \cup \mathcal{B}$) is NP-hard.

A k-ary relation R is said to have a *primitive positive definition* (pp-definition) over a constraint language $\mathcal{A}$ if $R(x_1, \ldots, x_k) \equiv \exists y_1, \ldots, y_{k'} : R_1(\mathbf{x_1}) \wedge \ldots \wedge R_m(\mathbf{x_m})$ where each $R_i \in \mathcal{A} \cup \{=_A\}$ and each $\mathbf{x_i}$ is a tuple of variables over $x_1, \ldots, x_k, y_1, \ldots, y_{k'}$ matching the arity of R_i. Here, and in the sequel, $=_A$ is the equality relation over A, i.e. $\{(a, a) \mid a \in A\}$. If $\mathcal{A}$ is a constraint language, then we let $\langle \mathcal{A} \rangle$ be the inclusion-wise smallest set of relations containing $\mathcal{A}$ closed under pp-definitions.

▶ **Theorem 1** ([16]). *Let $\mathcal{A}$ and $\mathcal{B}$ be structures with the same domain. If every relation of $\mathcal{A}$ has a primitive positive definition in $\mathcal{B}$, then there is a polynomial-time reduction from CSP($\mathcal{A}$) to CSP($\mathcal{B}$).*

When working with problems of the form $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ we additionally introduce the following simplifying notation: $\langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$ denotes the set of all pp-definable relations over $\mathcal{A} \cup \mathcal{B}$ using at most k atoms from $\mathcal{B}$. We now describe the corresponding algebraic objects. An operation $f\colon D^m \to D$ is a *polymorphism* of a relation $R \subseteq D^k$ if, for any choice of m tuples $(t_{11}, \ldots, t_{1k}), \ldots, (t_{m1}, \ldots, t_{mk})$ from R, it holds that $(f(t_{11}, \ldots, t_{m1}), \ldots, f(t_{1k}, \ldots, t_{mk}))$ is in R. An *endomorphism* is a polymorphism with arity one. If f is a polymorphism of R, then we sometimes say that R is *invariant* under f. A constraint language $\mathcal{A}$ has the polymorphism f if every relation in $\mathcal{A}$ has f as a polymorphism. We let Pol($\mathcal{A}$) and End($\mathcal{A}$) denote the sets of polymorphisms and endomorphisms of $\mathcal{A}$, respectively. If F is a set of functions over D, then Inv(F) denotes the set of relations over D that are invariant under every function in F. There are close algebraic connections between the operators $\langle \cdot \rangle$, Pol($\cdot$), and Inv($\cdot$). For instance, if $\mathcal{A}$ has a finite domain (or, more generally, if $\mathcal{A}$ is ω-categorical; see below), then we have a Galois connection $\langle \mathcal{A} \rangle = \mathrm{Inv}(\mathrm{Pol}(\mathcal{A}))$ [9, Theorem 5.1].

Polymorphisms enable us to compactly describe the tractable cases of Boolean CSPs.

▶ **Theorem 2** ([21])**.** *Let $\mathcal{A}$ be a constraint language over the Boolean domain. The problem CSP($\mathcal{A}$) is decidable in polynomial time if $\mathcal{A}$ is invariant under one of the following six operations: (1) the constant unary operation 0 ($\mathcal{A}$ is 0-valid), (2) the constant unary operation 1 ($\mathcal{A}$ is 1-valid), (3) the binary min operation $\sqcap$ ($\mathcal{A}$ is Horn), (4) the binary max operation $\sqcup$ ($\mathcal{A}$ is anti-Horn), (5) the ternary majority operation $M(x, y, z) = (x \sqcap y) \sqcup (x \sqcap z) \sqcup (y \sqcap z)$ ($\mathcal{A}$ is 2-SAT), or (6) the ternary minority operation $m(x, y, z) = x \oplus y \oplus z$ where $\oplus$ is the addition operator in $GF(2)$ ($\mathcal{A}$ is affine). Otherwise, the problem CSP($\mathcal{A}$) is NP-complete.*

A Boolean constraint language that satisfies condition (3), (4), (5), or (6) is called *Schaefer*.

A finite-domain structure $\mathcal{A}$ is a *core* if every $e \in \text{End}(\mathcal{A})$ is a bijection. We let $f(R) = \{(f(t_1), \ldots, f(t_n)) \mid (t_1, \ldots, t_n) \in R\}$ when $f \colon A \to A$ and $R \in \mathcal{A}$. If $e \in \text{End}(\mathcal{A})$ has minimal range, then $e(\mathcal{A}) = \{e(R) \mid R \in \mathcal{A}\}$ is a core and this core is unique up to isomorphism. We can thus speak about *the core* $\mathcal{A}^c$ of $\mathcal{A}$. It is easy to see that CSP($\mathcal{A}$) and CSP($\mathcal{A}^c$) are equivalent under polynomial-time reductions (indeed, even log-space reductions suffice). Another useful equivalence concerns constant relations. Let $\mathcal{A}^+$ denote the structure $\mathcal{A}$ expanded by all unary singleton relations $\{(a)\}$, $a \in A$. If $\mathcal{A}$ is a core, then CSP($\mathcal{A}$) and CSP($\mathcal{A}^+$) are equivalent under polynomial-time reductions [1].

We will frequently consider *ω-categorical* structures. An *automorphism* of a structure $\mathcal{A}$ is a permutation α of its domain A such that both α and its inverse are homomorphisms. The set of all automorphisms of a structure $\mathcal{A}$ is denoted by Aut($\mathcal{A}$), and forms a group with respect to composition. The *orbit* of $(a_1, \ldots, a_n) \in A^n$ in Aut($\mathcal{A}$) is the set $\{(\alpha(a_1), \ldots, \alpha(a_n)) \mid \alpha \in \text{Aut}(\mathcal{A})\}$. Let Orb($\mathcal{A}$) denote the set of orbits of n-tuples in Aut($\mathcal{A}$) (for all $n \geq 1$). A structure $\mathcal{A}$ with countable domain is ω-categorical if and only if Aut($\mathcal{A}$) is *oligomorphic*, i.e., it has only finitely many orbits of n-tuples for all $n \geq 1$.

Two important classes of ω-categorical structures are *equality languages* (respectively, *temporal languages*) where each relation can be defined as the set of models of a first-order formula over $(\mathbb{N}; =)$ (respectively, $(\mathbb{Q}; <)$). Importantly, Aut($\mathcal{A}$) is the full symmetric group if $\mathcal{A}$ is an equality language. A relation in an equality language is said to be *0-valid* if it contains *any* constant tuple. This is justified since if the relation is invariant under one constant operation, then it is invariant under all constant operations.The computational complexity of CSP for equality languages was classified by Bodirsky and Kára [7, Theorem 1]: for any equality language $\mathcal{A}$, CSP($\mathcal{A}$) is solvable in polynomial time if $\mathcal{A}$ is 0-valid or invariant under a binary injective operation, and is NP-complete otherwise.

3 Applications of Alien Constraints

We will now demonstrate how alien constraints can be used for studying the complexity of CSP-related problem: Section 3.1 contains an example where we analyse the complexity of *redundancy*, *equivalence*, and *implication* problems, and we consider connections between the model checking problem and CSPs with alien constraints in Section 3.2. To relate problem complexity we use *Turing reductions*: a problem L_1 is *polynomial-time Turing reducible* to L_2 (denoted $L_1 \leq^p_T L_2$) if it can be solved in polynomial time using an oracle for L_2. Two problems L_1 and L_2 are *polynomial-time Turing equivalent* if $L_1 \leq^p_T L_2$ and $L_2 \leq^p_T L_1$.

3.1 The Redundancy Problem and its Relatives

We will now study the complexity of a family of well-known computational problems. We begin by some definitions. Let $\mathcal{A}$ denote a constraint language and assume that $I = (V, C)$ is an instance of CSP($\mathcal{A}$). We say that a constraint $c \in C$ is *redundant* in I if $\text{Sol}((V, C)) = \text{Sol}((V, C \setminus \{c\}))$. We have the following computational problems.

REDUNDANT($\mathcal{A}$)
Instance: An instance (V, C) of CSP($\mathcal{A}$) and a constraint $c \in C$.
Question: Is c redundant in (V, C)?

IMPL($\mathcal{A}$)
Instance: Two instances $(V, C_1), (V, C_2)$ of CSP($\mathcal{A}$).
Question: Does (V, C_1) imply (V, C_2), i.e., is it the case that $\mathrm{Sol}((V, C_1)) \subseteq \mathrm{Sol}((V, C_2))$?

EQUIV($\mathcal{A}$)
Instance: Two instances $(V, C_1), (V, C_2)$ of CSP($\mathcal{A}$).
Question: Is it the case that $\mathrm{Sol}((V, C_1)) = \mathrm{Sol}((V, C_2))$?

Before we start working with alien constraints, we exhibit a close connection between REDUNDANT($\cdot$), EQUIV($\cdot$), and IMPL($\cdot$).

▶ **Lemma 3.** *Let $\mathcal{A}$ be a structure. The problems* EQUIV($\mathcal{A}$)*,* IMPL($\mathcal{A}$)*, and* REDUNDANT($\mathcal{A}$) *are polynomial-time Turing equivalent.*

Proof. We show that (1) EQUIV($\mathcal{A}$) $\leq_T^p$ IMPL($\mathcal{A}$), (2) IMPL($\mathcal{A}$) $\leq_T^p$ REDUNDANT($\mathcal{A}$), and (3) REDUNDANT($\mathcal{A}$) $\leq_T^p$ EQUIV($\mathcal{A}$).

(1). Let $((V, C_1), (V, C_2))$ be an instance of EQUIV($\mathcal{A}$). We need to check whether $\mathrm{Sol}((V, C_1)) = \mathrm{Sol}((V, C_2))$. This is true if and only if the two IMPL instances $((V, C_1), (V, C_2))$ and $((V, C_2), (V, C_1))$ are yes-instances.

(2). Let $((V, C_1), (V, C_2))$ be an instance of IMPL($\mathcal{A}$). For each constraint $c \in C_2$, first check whether C_1 implies $\{c\}$ by (a) checking if $c \in C_1$, in which case C_1 trivially implies $\{c\}$, (b) if not, then check whether c is redundant in $C_1 \cup \{c\}$, in which case we answer yes, and otherwise no. If C_1 implies $\{c\}$ for every $c \in C_2$ then C_1 implies C_2 and we answer yes, and otherwise no.

(3). Let $I = ((V, C), c)$ be an instance of REDUNDANT($\mathcal{A}$). It is obvious that I is a yes-instance if and only if the instance $((V, C), (V, C \setminus \{c\}))$ is a yes-instance of EQUIV($\mathcal{A}$). ◀

Next, we show how the complexity of REDUNDANT($\mathcal{A}$) can be analysed by exploiting CSPs with alien constraints. If R is a k-ary relation over domain D, then we let $\bar{R}$ denote its *complement*, i.e. $\bar{R} = D^k \setminus R$.

▶ **Theorem 4** ($\star$). *Let $\mathcal{A}$ be a structure with domain A. If CSP($\mathcal{A}$) is not in* P*, then* REDUNDANT($\mathcal{A}$) *is not in* P*. In particular,* REDUNDANT($\mathcal{A}$) *is* NP*-hard (under polynomial-time Turing reductions) whenever CSP($\mathcal{A}$) is* NP*-hard. Otherwise,* REDUNDANT($\mathcal{A}$) *is in* P *if and only if for every relation $R \in \mathcal{A}$, $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \{\bar{R}\})$ is in* P*.*

Combining Theorem 4 with the forthcoming complexity classification of Boolean CSPs with alien constraints (Theorem 14) shows that Boolean REDUNDANT($\mathcal{A}$) is in P if and only if $\mathcal{A}$ is Schaefer. We have not found this result in the literature but we view it as folklore since it follows from other classification results (start from [11] or [22] and transfer the results to REDUNDANT($\mathcal{A}$) with the aid of Lemma 3). However, we claim that our proof is very different when compared to the proofs in [11] and [22]): Böhler et al. use a lengthy case analysis while Schnoor & Schnoor in addition uses the so-called weak base method, which scales poorly since not much is known about this construction for non-Boolean domains. We do not claim that our proof is superior, but we do not see how to generalize the classifications by Böhler et al. and Schnoor & Schnoor to larger (in particular infinite) domains since they are fundamentally based on Post's classification of Boolean clones. Such a generalization, on the other hand, is indeed possible with our approach. We demonstrate in Section 6.2

that we can obtain a full understanding of the complexity of CSPs with alien constraints for equality languages. This result carries over to $\textsc{Redundant}(\cdot)$ via Theorem 4, implying that we have a full complexity classification of $\textsc{Redundant}(\cdot)$ for equality languages. This result can immediately be transferred to $\textsc{Impl}(\cdot)$ and $\textsc{Equiv}(\cdot)$ by Lemma 3.

3.2 Model Checking

We follow [18] and view the *model checking* problem as follows: given a logic $\mathscr{L}$, a structure $\mathcal{A}$, and a sentence ϕ of $\mathscr{L}$, decide whether $\mathcal{A} \models \phi$. The main motivation for this problem is its connection to databases [23]. From the CSP perspective, we consider a slightly reformulated version: given an instance $I = (V, C)$ of $\mathrm{CSP}(\mathcal{A})$ and a formula ϕ with free variables in V, we ask if there is a tuple in $\mathrm{Sol}(I)$ that satisfies ϕ. If ϕ can be expressed as an instance I' of $\mathrm{CSP}(\mathcal{B})$ for some structure $\mathcal{B}$, then this is the same thing as if asking whether $I \cup I'$ has a solution or not. In the model-checking setting, we want to check whether ϕ is true in all solutions of I. If $\neg\phi$ can be expressed as an instance I' of $\mathrm{CSP}(\mathcal{B})$ for some structure $\mathcal{B}$, then we are done: every solution to I satisfies ϕ if and only if $\mathrm{CSP}(I \cup I')$ is not satisfiable, and this clarifies the connection with CSPs with alien constraints. For instance, one may view $\textsc{Impl}(\mathcal{A})$ (and consequently the underlying $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \bar{R})$ problems by Lemma 3 and Theorem 4) as the model checking problem restricted to queries that are $\mathcal{A}$-sentences constructed using the operators $\forall$ and $\vee$. Naturally, one wants the ability to use more complex queries such as (1) queries extended with other relations, i.e. queries constructed over an expanded structure, or (2) queries that are built using other logical connectives.

In both cases, it makes sense to study the fixed-parameter tractability of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ with parameter $\#\mathsf{ac}$ since the query is typically much smaller than the structure $\mathcal{A}$. The connection is quite obvious in the first case (one may view $\#\mathsf{ac}$ as measuring how "complex" the given query is) while it is more hidden in the second case. Let us therefore consider the negation operator. From a logical perspective, one may view a constraint $\bar{R}(x_1, \ldots, x_k)$ as the formula $\neg R(x_1, \ldots, x_k)$. Needless to say, the relation $\bar{R}$ is often not pp-definable in a structure $\mathcal{A}$ containing R but it may be existential positive definable in $\mathcal{A}$. Assume that the preconditions of the example hold and that $\mathrm{CSP}(\mathcal{A})$ is in P. We know that $\bar{R}$ has an existential positive definition in $\mathcal{A}$ for every $R \in \mathcal{A}$. Let $\bar{\mathcal{A}} = \{\bar{R} \mid R \in \mathcal{A}\}$ and consider the problem $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \bar{\mathcal{A}})$. The forthcoming Theorem 15 is applicable so this problem is in FPT parameterized by $\#\mathsf{ac}$. Now, the corresponding model checking problem is to decide if $\mathcal{A} \models \phi$ where ϕ is an $\mathcal{A}$-sentence constructed using the operators $\forall$ and $\vee$ and where we are additionally allowed to use negated relations $\neg R(x_1, \ldots, x_m)$. It follows that this problem is in FPT parameterized by the number of negated relations.

4 General Tools for Alien Constraints

We analyze the complexity of $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$, starting in Section 4.1 by investigating which of the classic algebraic tools are applicable to the alien constraint setting, and continuing in Section 4.2 by presenting a general FPT result. We will use these observations for proving various results but also for obtaining a better understanding of alien constraints.

4.1 Alien Constraints and Algebra

First, we have a straightforward generalization of Theorem 1 in the alien constraint setting.

▶ **Theorem 5** ($\star$). *Let $\mathcal{A}$ and $\mathcal{B}$ be two structures with disjoint signatures. There exists a polynomial time many-one reduction f from $\mathrm{CSP}_{\leq}(\mathcal{A}^* \cup \mathcal{B}^*)$ to $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ for any finite $\mathcal{A}^* \subseteq \langle \mathcal{A} \rangle$ and $\mathcal{B}^* \subseteq \langle \mathcal{A} \cup \mathcal{B} \rangle$. If $I = (V, C, k)$ is an instance of $\mathrm{CSP}_{\leq}(\mathcal{A}^* \cup \mathcal{B}^*)$ and $f(I) = (V', C', k')$, then k' only depends on k, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{B}^*$, so f is an fpt-reduction.*

This claim is, naturally, in general not true for $\mathrm{CSP}_{\leq k}(\mathcal{A}^* \cup \mathcal{B})$ for finite $\mathcal{A}^* \subseteq \langle \mathcal{A} \cup \mathcal{B} \rangle$. The idea underlying Theorem 5 can be used in many different ways and we give one example.

▶ **Proposition 6.** *If $\mathcal{A}, \mathcal{B}$ are structures and $R \in \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq 1}$, then $CSP_{\leq k}(\mathcal{A} \cup (\mathcal{B} \cup \{R\}))$ is polynomial-time reducible to $CSP_{\leq k}(\mathcal{A} \cup \mathcal{B})$.*

We proceed by relating $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ to the important idea of reducing to a core (recall Section 2). Recall that $\mathcal{A}^c$ denotes the (unique up to isomorphism) core of a finite-domain structure $\mathcal{A}$. For two structures $\mathcal{A} \cup \mathcal{B}$ we similarly write $(\mathcal{A} \cup \mathcal{B})^c$ for the core. Specifically, if $e \in \mathrm{End}(\mathcal{A} \cup \mathcal{B})$ has minimal range, then the core consists of $\{e(R) \mid R \in \mathcal{A}\} \cup \{e(R) \mid R \in \mathcal{B}\}$ of the same signature as $\mathcal{A}$ and $\mathcal{B}$, and the problem $\mathrm{CSP}_{\leq}((\mathcal{A} \cup \mathcal{B})^c)$ is thus well-defined.

▶ **Theorem 7** ($\star$)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two structures over a finite universe A. Then $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ and $CSP_{\leq}((\mathcal{A} \cup \mathcal{B})^c)$ are interreducible under both polynomial-time and fpt reductions.*

In general, it is *not* possible to reduce from $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ to $\mathrm{CSP}_{\leq k}(\mathcal{A}^c \cup \mathcal{B})$ or from $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ to $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B}^c)$. This can be seen as follows. Consider the Boolean relation $R(x_1, x_2, x_3) \equiv x_1 = x_2 \vee x_2 = x_3$, and let $\mathcal{A} = \{R\}$, $\mathcal{B} = \{\neq\}$. Then, $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is NP-hard (see e.g. Exercise 3.24 in [13]) so $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is pNP-hard. However, $\mathcal{A}$ is 0-valid, so $\mathcal{A}^c = \{\{(0,0,0)\}\}$, implying that $\mathrm{CSP}_{\leq}(\mathcal{A}^c \cup \mathcal{B})$ is in P.

4.2 Fixed-Parameter Tractability

We present an algorithm in this section that underlies many of our fixed-parameter tractability results and it is based on a particular notion of definability. The *existential fragment* of first-order logic consists of formulas that only use the operations negation, conjunction, disjunction, and existential quantification, while the *existential positive* fragment additionally disallows negation. We emphasize that it is required that the equality relation is allowed in existential (positive) definitions. We can view existential positive in a different way that is easier to use in our algorithm. Let $\mathcal{A}$ be a structure with domain A and assume that $R \subseteq A^m$ is defined via a existential positive definition over $\mathcal{A}$, i.e., $R(x_1, \ldots, x_m) \equiv \exists y_1, \ldots, y_n \colon \ \phi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ where ϕ is a quantifier-free existential positive $\mathcal{A}$-formula. Since ϕ can be written in disjunctive normal form without introducing negation or quantifiers, it follows that R is a finite union of relations in $\langle \mathcal{A} \rangle$.

▶ **Theorem 8.** *Assume the following.*

1. *$\mathcal{A}, \mathcal{B}$ are structures with the same domain A,*
2. *every relation in $\mathcal{B}$ is existential positive definable in $\mathcal{A}$, and*
3. *$CSP(\mathcal{A})$ is in P.*

Then $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in FPT *parameterized by* #ac.

Proof. Assume $\mathcal{B} = \{A; B_1, \ldots, B_m\}$. Condition 2. implies that B_i, $i \in [m]$, is a finite union of relations $B_i = R_i^1 \cup \cdots \cup R_i^{c_i}$ where $R_i^1, \ldots, R_i^{c_i}$ are in $\langle \mathcal{A} \rangle$. Let the structure $\mathcal{A}^*$ contain the relations in $\mathcal{A} \cup \{R_i^j \mid i \in [m] \text{ and } j \in [c_i]\}$. Clearly, $\mathcal{A}^*$ has a finite signature and the problem $\mathrm{CSP}(\mathcal{A}^*)$ is in P by Theorem 1 since every relation in $\mathcal{A}^*$ is a member of $\langle \mathcal{A} \rangle$. Let $b = \max\{c_i \mid i \in [m]\}$.

Let $((V, C), k)$ denote an arbitrary instance of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B}))$. The satisfiability of (V, C) can be checked via the following procedure. If C contains no $\mathcal{B}$-constraint, then check the satisfiability of (V, C) with the polynomial-time algorithm for $\mathrm{CSP}(\mathcal{A})$. Otherwise, pick one constraint $c = B_i(x_1, \ldots, x_q)$ with $B_i \in \mathcal{B}$ and check recursively the satisfiability of the following instances:

$$(V, (C \setminus \{c\}) \cup \{R_i^1(x_1, \ldots, x_q)\}), \ldots, (V, (C \setminus \{c\}) \cup \{R_i^{c_i}(x_1, \ldots, x_q)\}).$$

If at least one of the instances is satisfiable, then answer "yes" and otherwise "no". This is clearly a correct algorithm for $\mathrm{CSP}_{\le}(\mathcal{A} \cup \mathcal{B})$.

We continue with the complexity analysis. Note that the leaves in the computation tree produced by the algorithm are $\mathrm{CSP}(\mathcal{A}^*)$ instances and they are consequently solvable in polynomial time. The depth of the computation tree is at most k (since (V, C) contains at most k $\mathcal{B}$-constraints) and each node has at most b children. Thus, the problem can be solved in $b^k \cdot \mathrm{poly}(|I|)$ time. We conclude that $\mathrm{CSP}_{\le}(\mathcal{A} \cup \mathcal{B})$ is in FPT parameterized by #ac since b is a fixed constant that only depends on the structures $\mathcal{A}$ and $\mathcal{B}$. ◀

5 Finite-Domain Languages

This section is devoted to CSPs over finite domains. We begin in Section 5.1 by studying how the definability of constants affect the complexity of finite-domain CSPs with alien constraints, and we use this as a cornerstone for a parameterized FPT versus pNP dichotomy result for of $\mathrm{CSP}_{\le}(\mathcal{A} \cup \mathcal{B})$. We show a sharper result for Boolean structures in Section 5.2.

5.1 Parameterized Dichotomy

We begin with a simplifying result. For a finite set A, let $\mathcal{C}_A$ be the structure whose relations are the constants over A.

▶ **Lemma 9** ($\star$). *Let $\mathcal{A}$ be a structure over a domain A. For every $\mathcal{C} \subseteq \mathcal{C}_A$, $CSP(\mathcal{A} \cup \mathcal{C})$ is polynomial-time reducible to $CSP_{\le|\mathcal{C}|}(\mathcal{A} \cup \mathcal{C})$.*

Lemma 9 together with the basic algebraic results from Section 4.1 allows us to prove the following result that combines a more easily formulated fixed-parameter result (compared to Theorem 8) with a powerful hardness result.

▶ **Theorem 10** ($\star$). *Let $\mathcal{A}, \mathcal{B}$ be structures with finite domain D. Assume that $\mathcal{A} \cup \mathcal{B}$ is a core. If $CSP(\mathcal{A} \cup \mathcal{C}_A)$ is in* P*, then $CSP_{\le}(\mathcal{A} \cup \mathcal{B})$ is in* FPT *with parameter* #ac*. Otherwise, $CSP_{\le p}(\mathcal{A} \cup \mathcal{B})$ is* NP*-hard for some p that only depends on the structures $\mathcal{A}$ and $\mathcal{B}$.*

Proof. We provide a short proof sketch. Using the dichotomy of finite domain CSPs [12, 24], we first assume $\mathrm{CSP}(\mathcal{A} \cup \mathcal{C}_D)$ is in P. One can prove that every tuple over D is pp-definable over $\mathcal{A} \cup \mathcal{C}_D$ and then that each relation in $\mathcal{B}$ is existential positive definable over $\mathcal{A} \cup \mathcal{C}_D$. We can now apply Theorem 8, and $\mathrm{CSP}_{\le}(\mathcal{A} \cup \mathcal{B})$ is in FPT.

For the NP-hard case, we assume $\mathrm{CSP}(\mathcal{A} \cup \mathcal{C}_D)$ is NP-hard and construct a polynomial-time reduction from $\mathrm{CSP}(\mathcal{A} \cup \mathcal{C}_D)$ to $\mathrm{CSP}_{\le p}(\mathcal{A} \cup \mathcal{B})$. We use the endomorphisms of $\mathcal{A} \cup \mathcal{B}$ to construct a pp-definable relation E which allow us to simulate the constant relations, and a reduction to $\mathrm{CSP}_{\le 1}(\mathcal{A} \cup \{E\})$ to establish the claim via Lemma 9 and Theorem 5. ◀

Theorem 10 has broad applicability. Let us, for instance, consider a structure $\mathcal{A}$ with finite domain A and containing a finite number of relations from $\mathrm{Inv}(f)$ where $f \colon A^m \to A$ is idempotent ($f : A^m \to D$ is *idempotent* if $f(a, \dots, a) = a$ for all $a \in A$.) If $\mathrm{CSP}(\mathcal{A})$ is in P, then $\mathrm{CSP}(\mathcal{A} \cup \mathcal{C}_A)$ is in P since constant relations are invariant under f. Hence, $\mathrm{CSP}_{\le}(\mathcal{A} \cup \mathcal{B})$ is in FPT parameterized by #ac for *every* finite structure $\mathcal{B}$ with domain A by Theorem 10. Idempotent functions that give rise to polynomial-time solvable CSPs are fundamental and well-studied in the literature; see e.g. the survey by Barto et al. [1].

Via Theorem 7 we obtain the following parameterized complexity dichotomy separating problems in FPT from pNP-hard problems.

▶ **Corollary 11.** *Let $\mathcal{A}, \mathcal{B}$ be structures over the finite domain A. Then, $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ is either in* FPT *or* pNP*-hard (in parameter* #ac*).*

Proof. Let $e \in \mathrm{End}(\mathcal{A} \cup \mathcal{B})$ have minimal range and let $\mathcal{A}' = \{e(R) \mid R \in \mathcal{A}\}$ and $\mathcal{B}' = \{R \mid R \in \mathcal{B}\}$ be the two components of the core $(\mathcal{A} \cup \mathcal{B})^c$, and let $A' = \{e(a) \mid a \in A\}$ be the resulting domain. The problems $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ and $\mathrm{CSP}_{\leq}(\mathcal{A}' \cup \mathcal{B}')$ are fpt-interreducible by Theorem 7. The problem $\mathrm{CSP}(\mathcal{A}' \cup \mathcal{C}_{A'})$ is either in P or is NP-hard by the CSP dichotomy theorem [12, 24]. In the first case, $\mathrm{CSP}_{\leq}(\mathcal{A}' \cup \mathcal{B}')$ (and thus $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$) is in FPT with parameter #ac. Otherwise, $\mathrm{CSP}_{\leq}(\mathcal{A}' \cup \mathcal{B}')$ is pNP-hard, and the fpt-reduction from $\mathrm{CSP}_{\leq}(\mathcal{A}' \cup \mathcal{B}')$ to $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ establishes pNP-hardness for the latter. ◀

Corollary 11 must be used with caution: it does not imply that $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is NP-hard and results such as Theorem 4 may not be applicable. This encourages the refinement of coarse complexity results based on Theorem 10. We use Boolean relations as an example of this in the next section.

5.2 Classification of Boolean Languages

We present a complexity classification of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ when $\mathcal{A}$ and $\mathcal{B}$ are Boolean structures (Theorem 14). We begin with two auxiliary results and we define relations $c_0 = \{(0)\}$ and $c_1 = \{(1)\}$.

▶ **Lemma 12** ($\star$)**.** *Let $\mathcal{A}$ be a Boolean structure where $c_0 \in \langle \mathcal{A} \rangle$. If an n-ary Boolean $R \neq \emptyset$ is not 0-valid then $c_1 \in \langle \mathcal{A} \cup \{R\} \rangle_{\leq 1}$.*

We say that a Boolean relation R is *invariant under complement* if it is invariant under the operation $\{0 \mapsto 1, 1 \mapsto 0\}$. This is equivalent to $(t_1, \ldots, t_k) \in R$ if and only if $(1 - t_1, \ldots, 1 - t_k) \in R$.

▶ **Lemma 13** ($\star$)**.** *Let $\mathcal{A}$ be a Boolean structure with finite signature. If $\mathcal{A}$ is invariant under complement, then $CSP(\mathcal{A} \cup \{c_0, c_1\})$ is polynomial-time reducible to $CSP_{\leq 1}(\mathcal{A} \cup \{\neq\})$.*

We are now ready for analysing the complexity of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ when $\mathcal{A}$ and $\mathcal{B}$ are Boolean structures. We use a simplifying concept: a *0/1*-pair (R_0, R_1) contains two Boolean relations where R_0 is 0-valid but not 1-valid and R_1 is 1-valid but not 0-valid.

▶ **Theorem 14.** *Let $\mathcal{A}$ and $\mathcal{B}$ be Boolean structures such that $CSP(\mathcal{A})$ is in* P *and $CSP(\mathcal{A} \cup \mathcal{B})$ is* NP*-hard. Then the following holds.*

1. *If $\mathcal{A}$ is Schaefer, then $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in* FPT *with parameter* #ac*.*
2. *If (i) $\mathcal{A}$ is not Schaefer, (ii) $\mathcal{A}$ is both 0- and 1-valid, (iii) $\mathcal{B}$ contains a* 0/1*-pair, and (iv) $\mathcal{B}$ is 0- or 1-valid, then $CSP_{\leq 2}(\mathcal{A} \cup \mathcal{B})$ is* NP*-hard and $CSP_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is in* P*.*
3. *Otherwise, $CSP_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is* NP*-hard.*

Proof. Assume $\mathcal{A}$ is Schaefer and let $\mathcal{A}^+ = \mathcal{A} \cup \{c_0, c_1\}$. The structure $\mathcal{A}^+$ is clearly a core and $\mathcal{A}^+ \cup \mathcal{B}$ is a core, too. The problem $\mathrm{CSP}(\mathcal{A}^+)$ is in P by Theorem 2 so Theorem 10 implies that $\mathrm{CSP}_{\leq}(\mathcal{A}^+ \cup \mathcal{B})$ (and naturally $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$) is in FPT parameterized by #ac. Since $\mathrm{CSP}(\mathcal{A})$ is in P, we know from Theorem 2 that $\mathcal{A}$ is 0-valid, 1-valid or Schaefer. We assume henceforth that $\mathcal{A}$ is 0-valid and not Schaefer; the other case is analogous. If $\mathcal{B}$ is 0-valid, then $\mathrm{CSP}(\mathcal{A} \cup \mathcal{B})$ is trivially in P and this is ruled out by our initial assumptions. We assume henceforth that $\mathcal{B}$ is not 0-valid and consider two cases depending on whether c_0 is pp-definable in $\mathcal{A}$ or not.

Case 1. c_0 is pp-definable in $\mathcal{A}$. We know that $\mathrm{CSP}(\mathcal{A} \cup \{c_0, c_1\})$ is NP-hard by Theorem 2 since $\mathcal{A}$ is not Schaefer. We can thus assume that $\mathrm{CSP}(\mathcal{A} \cup \{c_1\})$ is NP-hard. Lemma 9 implies that $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \{c_1\})$ is NP-hard. The relation c_1 is in $\langle \mathcal{A} \cup \mathcal{B} \rangle_{\mathcal{B} \leq 1}$ by Lemma 12 so we conclude that $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is NP-hard.

Case 2. c_0 is not pp-definable in $\mathcal{A}$. This implies that every relation in $\mathcal{A}$ is simultaneously 0- and 1-valid. To see this, assume to the contrary that $\mathcal{A}$ contains a relation that is not 1-valid. Then, $x = 0 \Leftrightarrow R(x, \ldots, x)$ and c_0 is pp-definable in $\mathcal{A}$. This implies that $\mathcal{B}$ contains (a) a relation that is not invariant under any constant operation or (b) every relation is closed under a constant operation and $\mathcal{B}$ contains a 0/1-pair. Note that if (a) and (b) does not hold, then $\mathcal{B}$ is invariant under a constant operation and $\mathrm{CSP}(\mathcal{A} \cup \mathcal{B})$ is trivially in P.

Case 2(a). There is a a relation R in $\mathcal{B}$ that is not invariant under any constant operation, i.e. $(0, \ldots, 0) \notin R$ and $(1, \ldots, 1) \notin R$. The relation R has arity $a \geq 2$. Let t be the tuple in R that contains the maximal number b of 0:s. Clearly, $b < a$. We assume that the arguments are permuted so that t begins with b 0:s and continues with $a - b$ 1:s. Consider the pp-defintion

$$S(x, y) \equiv R(\underbrace{x, \ldots, x}_{b \text{ occ.}}, \underbrace{y, \ldots, y}_{a-b \text{ occ.}}).$$

There are two possibilities: either $S(x, y) \Leftrightarrow x = 0 \wedge y = 1$ or $S(x, y) \Leftrightarrow x \neq y$. In the first case we are done since $\mathrm{CSP}(\mathcal{A} \cup \{c_0, c_1\})$ is NP-hard (recall that $\mathcal{A}$ is not Schaefer) and $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is easily seen to be NP-hard by Lemma 9. Let us consider the second case. If $\mathcal{A}$ is invariant under complement, then $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is NP-hard by Lemma 13. If $\mathcal{A}$ is not invariant under complement, then we claim that c_0 and c_1 can be pp-defined with the aid of $\neq$. Arbitrarily choose a relation T in $\mathcal{A}$ that contains a tuple $t = (t_1, \ldots, t_a)$ such that $(1 - t_1, \ldots, 1 - t_a) \notin T$ – note that t cannot be a constant tuple since both $(0, \ldots, 0)$ and $(1, \ldots, 1)$ are in T. Assume that t contains b 0:s and that the arguments are permuted so that t begins with b 0:s followed by $a - b$ 1:s. Consider the pp-definition

$$U(x, y) \equiv x \neq y \wedge T(\underbrace{x, \ldots, x}_{b \text{ occ.}}, \underbrace{y, \ldots, y}_{a-b \text{ occ.}}).$$

The relation U contains the single tuple $(0, 1)$. We know that $\mathrm{CSP}(\mathcal{A} \cup \{c_0, c_1\})$ is NP-hard (recall that $\mathcal{A}$ is not Schaefer) and Lemma 9 implies that $\mathrm{CSP}_{\leq 2}(\mathcal{A} \cup \{c_0, c_1\})$ is NP-hard, too. It is now easy to see that $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is NP-hard via the definition of U.

Case 2(b). Every relation in $\mathcal{B}$ is closed under at least one constant operation and $\mathcal{B}$ contains a 0/1-pair (R_0, R_1). Since $\mathcal{A}$ is both 0- and 1-valid, it follows that $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ is in P. The constant relations c_0 and c_1 are pp-definable in $\{R_0, R_1\}$ since $x = 0 \Leftrightarrow R_0(x, \ldots, x)$ and $x = 1 \Leftrightarrow R_1(x, \ldots, x)$. This implies with the aid of Lemma 9 that $\mathrm{CSP}_{\leq 2}(\mathcal{A} \cup \mathcal{B})$ is NP-hard since $\mathcal{A}$ is not Schaefer. ◀

Theorem 14 carries over to Boolean REDUNDANT(·), EQUIV(·) and IMPL(·) by Lemma 3 combined with Theorem 4, so these problems are in P if and only if $\mathcal{A}$ is Schaefer (case 2. in Theorem 14 is not applicable when analysing these problems since it requires $|\mathcal{B}| \geq 2$). Otherwise, they are NP-complete under polynomial-time Turing reductions. The *meta-problem* for Boolean CSPs with alien constraints is decidable, i.e., there is an algorithm that decides for Boolean structures $\mathcal{A}, \mathcal{B}$ whether $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in case 1., 2., or 3. of Theorem 14. This is obvious since we have polymorphism descriptions of the Schaefer languages.

6 Infinite-Domain Languages

We focus on infinite-domain CSPs in this section. We begin Section 6.1 by discussing certain problems when CSPs with alien constraints are generalized to infinite domains. Our conclusion is that restricting ourselves to ω-categorical structures is a viable first step: ω-categorical structures constitute a rich class of CSPs and we can generalize at least some of the machinery from Section 5 to this setting. We demonstrate this in Section 6.2 where we obtain a complete complexity classification for equality languages.

6.1 Orbits and Infinite-Domain CSPs

It is not straightforward to tranfer the results in Section 5 to the infinite-domain regime. First, let us consider Theorem 8. In contrast to finite domains, relations in $\mathcal{B}$ may not be finite unions of relations in $\langle \mathcal{A} \rangle$ or, equivalently, not being definable with an existential positive formula. Second, let us consider Theorem 10: the proof is based on structures expanded with symbols for each domain value and this leads to problematic structures with infinite signatures. The proof is also based on the assumption that CSPs are either polynomial-time solvable or NP-complete, and this is no longer true [5]. It is thus necessary to restrict our attention to some class of structures with sufficiently pleasant properties. A natural choice is ω-categorical structures that allows us to reformulate Theorem 8 as follows.

▶ **Theorem 15** ($\star$). *Assume the following.*

1. *$\mathcal{A}, \mathcal{B}$ are structures with the same countable (not necessarily infinite) domain A,*
2. *$\mathcal{A}$ and $\mathcal{B}$ are ω-categorical,*
3. *every relation in* $\text{Orb}(\mathcal{B})$ *is existential primitive definable in $\langle \mathcal{A} \rangle$, and*
4. *CSP($\mathcal{A}$) is in* P

Then CSP$_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in FPT *parameterized by* #ac.

▶ **Example 16.** Results related to Theorem 15 have been presented in the literature. Recall that RCC5 and RCC8 are spatial formalism with binary relations that are disjunctions of certain basic relations [20]. Li et al. [17] prove that if $\mathcal{A}$ is a polynomial-time solvable RCC5 or RCC8 constraint language containing all basic relations, then Redundant($\mathcal{A}$) is in P. This immediately follows from combining Theorem 4 and Theorem 15 since RCC5 and RCC8 can be represented by ω-categorical constraint languages [3, 10] and every RCC5/RCC8 relation is existential primitive definable in the structure of basic relations by definition. This result can be generalized to a much larger class of relations in the case of RCC5 since the orbits of k-tuples are pp-definable in the structure of basic relations [6, Proposition 35].

A general hardness result based on the principles behind Theorem 10 does not seem possible in the infinite-domain setting, even for ω-categorical structures. The hardness proof in Theorem 10 utilizes variables given fixed values and a direct generalization would lead to groups of variables that together form an orbit of an n-tuple. Such gadgets behave very differently from variables given fixed values: in particular, they do not admit a result similar to Lemma 9. Thus, hardness results needs to be constructed in other ways.

We know from Section 4.1 that $\text{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ and $\text{CSP}_{\leq}((\mathcal{A} \cup \mathcal{B})^c)$ are the same when $\mathcal{A}$ and $\mathcal{B}$ has the same finite domain. We now consider a generalisation of cores to infinite domains from Bodirsky [2]: an ω-categorical structure $\mathcal{A}$ with countable domain is a *model-complete core* if every relation in $\text{Orb}(\mathcal{A})$ is pp-definable in $\mathcal{A}$. There is an obvious infinite-domain analogue of Theorem 7: if $\mathcal{A}' \cup \mathcal{B}'$ is the model-complete core of $\mathcal{A} \cup \mathcal{B}$ (where $\mathcal{A}, \mathcal{B}$ are ω-categorical structures over a countable domain A), then $\text{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ polynomial-time reduces to $\text{CSP}_{\leq}(\mathcal{A}' \cup \mathcal{B}')$. Model-complete cores share many other properties with cores, too. With this said, it is interesting to understand model-complete cores in the context of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$, simply because they are so well-studied and exhibit useful properties. We merely touch upon this subject by making an observation that we use in Section 6.2.

▶ **Lemma 17** ($\star$). *Let $\mathcal{A}$ and $\mathcal{B}$ denote ω-categorical structures with a countable domain A. Assume that $\mathcal{A}$ is a model-complete core and CSP($\mathcal{A}$) is in* P. *Then, $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in* FPT *parameterized by* #ac *for every structure $\mathcal{B}$ such that $\mathrm{Orb}(\mathcal{B}) \subseteq \mathrm{Orb}(\mathcal{A})$.*

6.2 Classification of Equality Languages

We present a complexity classification of $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ for equality languages $\mathcal{A}$, $\mathcal{B}$. Essentially, there are two interesting cases: when $\mathcal{A}$ is Horn, and when $\mathcal{A}$ is 0-valid and not Horn. In the former case, $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in FPT parameterized by #ac, while in the second case it is pNP-hard. It turns out that the ability to pp-define the arity-c disequality relation, where c depends only on $\mathcal{A}$, using at most k alien constraints, determines the complexity. A dichotomy for Redundant($\cdot$), Impl($\cdot$), and Equiv($\cdot$) follows: these problems are either in P or NP-hard under polynomial-time Turing reductions.

Recall that CSP($\mathcal{A}$) for a finite equality constraint language $\mathcal{A}$ is in P if $\mathcal{A}$ is 0-valid or preserved by a binary injective operation, and NP-hard otherwise, and that the automorphism group for equality languages is the symmetric group Σ on $\mathbb{N}$, i.e. the set of permutations on $\mathbb{N}$. It is easy to see that an orbit of a k-tuple $(a_1, \ldots, a_k)$ is pp-definable in $\{=, \neq\}$. For instance, the orbit of $(0,0,1,2)$ is defined by $O(x_1, x_2, x_3, x_4) \equiv x_1 = x_2 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$. Observe that $\neq$ is invariant under every binary injective operation, so if $\mathcal{A}$ is Horn, then $\neq \in \langle \mathcal{A} \rangle$ and every orbit of n-tuples under Σ is pp-definable in $\mathcal{A}$. Thus, $\mathcal{A}$ is a model-complete core as pointed out in Section 6.1. Lemma 17 now implies the following.

▶ **Corollary 18.** *Let $\mathcal{A}$ and $\mathcal{B}$ be equality languages. If $\mathcal{A}$ is Horn, then $CSP_{\leq}(\mathcal{A} \cup \mathcal{B})$ is in* FPT *parameterized by* #ac*.*

Thus, we need to classify the complexity of $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ for every k, where $\mathcal{A}$ is 0-valid and not Horn, and $\mathcal{B}$ is not 0-valid. We will rely on results about the complexity of singleton expansions of equality constraint languages. Let $\mathcal{A}$ be a constraint language over the domain $\mathbb{N}$. By $\mathcal{A}^+_c$ we denote the expansion of $\mathcal{A}$ with c singleton relations, i.e. $\mathcal{A}^+_c = \mathcal{A} \cup \{\{1\}, \ldots, \{c\}\}$. The complexity of $\mathrm{CSP}(\mathcal{A}^+_c)$ for equality constraint languages $\mathcal{A}$ and all constants c was classified by Osipov & Wahlström [19, Section 7], building on the detailed study of polymorphisms of equality constraint languages by Bodirsky et al. [4].

The connection between $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ and $\mathrm{CSP}(\mathcal{A}^+_c)$ is the following. In one direction, we can augment every instance of CSP($\mathcal{A}$) with c fresh variables $z_1, \ldots, z_c$ and, assuming k is large enough and $\mathcal{B}$ is not 0-valid, use $\mathcal{B}$-constraints to ensure that $z_1, \ldots, z_c$ attain distinct values in every satisfying assignment. Given that $\mathcal{A}$ is invariant under every permutation of $\mathbb{N}$, we can now treat $z_1, \ldots, z_c$ as constants, e.g. as $1, \ldots, c$, and transfer hardness results from the singleton expansion to our problem. In the other direction, if the relation $\mathrm{NEQ}_{c+1} \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$, then every satisfiable instance of $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ has a solution with range $[c]$, and $\mathcal{A}^+_c$ is tractable: indeed, a satisfiable instance without such a solution would be a pp-definition of $\mathrm{NEQ}_{c'}$ for some $c' > c$. These connections are formalized in Lemmas 23 and 24. We will leverage the following hardness result.

▶ **Lemma 19** (Follows from Theorem 54 in [19]). *Let $\mathcal{A}$ be a finite equality language. If $\mathcal{A}$ is not Horn, then $CSP(\mathcal{A}^+_c)$ is* NP*-hard for some $c = c(\mathcal{A})$.*

Our main tool for studying singleton expansions are *retractions*.

▶ **Definition 20.** *Let $\mathcal{A}$ be an equality language. An operation $f\colon \mathbb{N} \to [c]$ is a* retraction of $\mathcal{A}$ to $[c]$ *if f is an endomorphism of $\mathcal{A}$ where $f(i) = i$ for all $i \in [c]$. If $\mathcal{A}$ admits a retraction f to $[c]$, then we say that $\mathcal{A}$* retracts to $[c]$, *and $\mathcal{A}_f$ is* a retract (of $\mathcal{A}$ to $[c]$).

We obtain a useful characterization of retracts.

▶ **Lemma 21.** *Let $\mathcal{A}$ be an equality language and f be a retraction from $\mathcal{A}$ to $[c]$. Then $f(R) = R \cap [c]^{\mathrm{ar}(R)}$ for all $R \in \mathcal{A}$.*

Proof. First, observe that $f(R) \subseteq R \cap [c]^{\mathrm{ar}(R)}$: indeed, f is an endomorphism, so $f(R) \subseteq R$, and $f(R) \subseteq [c]^{\mathrm{ar}(R)}$ because the range of f is $[c]$. Moreover, we have $R \cap [c]^{\mathrm{ar}(R)} \subseteq f(R)$ because f is constant on $[c]$, so it preserves every tuple in $[c]^{\mathrm{ar}(R)}$. ◀

The finite-domain language $\{R \cap [c]^{\mathrm{ar}(R)} : R \in \mathcal{A}\}$ is called a *c-slice of $\mathcal{A}$* in [19, Section 7]. Lemma 21 shows that a c-slice of $\mathcal{A}$ is the retract $\mathcal{A}_f$ under any retraction f from $\mathcal{A}$ to $[c]$. Note that the definition of the c-slice does not depend on f, so we can talk about *the retract of $\mathcal{A}$ to* $[c]$. We will use this fact implicitly when transferring results from Theorem 57 in [19].

▶ **Lemma 22** (Follows from Theorem 57 in [19])**.** *Let $\mathcal{A}$ be an equality language that is 0-valid and not Horn, and let c be a positive integer. Then exactly one of the following holds:*
- *$\mathcal{A}$ does not retract to $[c]$, and CSP($\mathcal{A}_c^+$) is* NP*-hard.*
- *$\mathcal{A}$ retracts to $[c]$, and CSP($\mathcal{A}_c^+$) is* NP*-hard for all $c \geq 2$.*
- *$\mathcal{A}$ retracts to $[c]$, and both CSP(Δ_c^+) for the retract Δ and CSP($\mathcal{A}_c^+$) are in* P*.*

Let $\mathrm{NEQ}_r = \{(t_1, \ldots, t_r) \in \mathbb{N}^r : |\{t_1, \ldots, t_r\}| = r\}$, i.e. the relation that contains every tuple of arity r with all entries distinct.

▶ **Lemma 23** ($\star$)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be equality languages and $c \in \mathbb{Z}_+$. If $\mathrm{NEQ}_{c+1} \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$, then every satisfiable instance of CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) has a solution whose range is in $[c]$.*

▶ **Lemma 24** ($\star$)**.** *Let $\mathcal{A}$, $\mathcal{B}$ be two equality constraint languages, and let $c \in \mathbb{Z}_+$ be an integer. CSP($\mathcal{A}_c^+$) is polynomial-time reducible to CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) whenever $\mathrm{NEQ}_c \in \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$.*

We are ready to present the classification.

▶ **Theorem 25.** *Let $\mathcal{A}$ and $\mathcal{B}$ be equality languages such that CSP($\mathcal{A}$) is in* P *and CSP($\mathcal{A} \cup \mathcal{B}$) is* NP*-hard.*
1. *If $\mathcal{A}$ is Horn, CSP$_{\leq}$($\mathcal{A} \cup \mathcal{B}$) is in* FPT *parameterized by* #ac.
2. *If $\mathcal{A}$ is not Horn, CSP$_{\leq}$($\mathcal{A} \cup \mathcal{B}$) is* pNP*-hard parameterized by* #ac*. Moreover, there exists an integer $c = c(\mathcal{A})$ such that CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) is in* P *whenever $\mathrm{NEQ}_c \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$, and is* NP*-hard otherwise.*

Proof. CSP($\mathcal{A}$) is in P so $\mathcal{A}$ is Horn or 0-valid. If $\mathcal{A}$ is Horn, then Corollary 18 applies, proving part 1 of the theorem. Suppose $\mathcal{A}$ is 0-valid and not Horn. By applying Lemma 19 to $\mathcal{A}$, we infer that there is a minimum positive integer c such that CSP($\mathcal{A}_c^+$) is NP-hard. Since $\mathcal{A}$ is 0-valid, we have $c \geq 2$. Using Lemma 24, we can reduce CSP($\mathcal{A}_c^+$) to CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) in polynomial time whenever $\mathrm{NEQ}_c \in \langle \mathcal{A} \cup \mathcal{B} \rangle_{\mathcal{B} \leq k}$, proving that the latter problem is NP-hard. Observe that $\mathcal{B}$ is not 0-valid because CSP($\mathcal{A} \cup \mathcal{B}$) is NP-hard, so $\neq \in \langle \mathcal{B} \rangle$ and $\mathrm{NEQ}_c \in \langle \mathcal{A} \cup \mathcal{B} \rangle_{\mathcal{B} \leq k}$ for some finite $k \leq \binom{c}{2}$. This show the pNP-hardness result in part 2.

To complete the proof of part 2, it suffices to show that we can solve CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) in polynomial time whenever $\mathrm{NEQ}_c \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\mathcal{B} \leq k}$. To this end, observe that, by the choice of c, if $c' < c$, then CSP($\mathcal{A}_{c'}^+$) is in P. Then, by Lemma 22, $\mathcal{A}$ retracts to the finite domain $[c']$, and the retract Δ is such that CSP($\Delta_{c'}^+$) is in P. We will use the algorithm for CSP($\Delta_{c'}^+$) in our algorithm for CSP$_{\leq k}$($\mathcal{A} \cup \mathcal{B}$) that works for all k such that $\mathrm{NEQ}_c \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\leq k}$.

Let I be an instance of $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$. Since $\mathrm{NEQ}_c \notin \langle \mathcal{A} \cup \mathcal{B} \rangle_{\mathcal{B} \leq k}$, Lemma 23 implies that I is satisfiable if and only if it admits a satisfying assignment with range $[c-1]$. Let X be the set of variables in I that occur in the scopes of the alien constraints. Note that $|X| \in O(k)$. Enumerate all assignments $\alpha : X \to [c-1]$, and check if it satisfies all $\mathcal{B}$-constraints in I. If not, reject it, otherwise remove the $\mathcal{B}$-constraints and add unary constraints $x = \alpha(x)$ for all $x \in X$ instead. This leads to an instance of $\mathrm{CSP}(\Delta^+_{c-1})$, which is solvable in polynomial time. If we obtain a satisfiable instance for some α, then accept I, and otherwise reject it. Correctness follows by Lemma 23 and the fact that the algorithm considers all assignments from X to $[c]$. We make $2^{O(k)}$ calls to the algorithm for $\mathrm{CSP}(\Delta^+_{c-1})$, where k is a fixed constant, and each call runs in polynomial time. This completes the proof. ◀

Theorem 14 implies that $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ is pNP-hard if and only if $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ is NP-hard for some k, and it is in FPT parameterized by #ac otherwise. Theorem 25 now implies a dichotomy for Redundant(·), Impl(·), and Equiv(·) over finite equality languages.

▶ **Theorem 26** (⋆). *Let $\mathcal{A}$ be a finite equality language. Then Redundant($\mathcal{A}$), Impl($\mathcal{A}$), and Equiv($\mathcal{A}$) are either in* P *or* NP*-hard (under polynomial-time Turing reductions).*

Algebraically characterizing the exact borderline between tractable and hard cases of the problem seems difficult. In particular, given a 0-valid non-Horn equality language $\mathcal{A}$, answering whether $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \bar{\mathcal{A}})$ is in P, i.e. whether $\mathrm{NEQ}_c \in \langle \mathcal{A} \cup \bar{R} \rangle_{\leq 1}$ for some $R \in \mathcal{A}$ and large enough c, requires a deeper understanding of such languages. However, one can show that the answer to this, and even a more general question is decidable.

▶ **Proposition 27** (⋆). *There is an algorithm that takes two equality constraint languages $\mathcal{A}$ and $\mathcal{B}$ and outputs minimum $k \in \mathbb{N} \cup \{\infty\}$ such that $\mathrm{CSP}_{\leq k}(\mathcal{A} \cup \mathcal{B})$ is* NP*-hard.*

7 Discussion

We have focused on structures with finite signatures in this paper. This is common in the CSP literature since relational structures with infinite signature cause vexatious representational issues. It may, though, be interesting to look at structures with infinite signatures, too. Zhuk [25] observes that the complexity of the following problem is open: given a system of linear equations mod 2 and a single linear equation mod 24, find a satisfying assignment over the domain $\{0, 1\}$. The equations have unbounded arity so this problem can be viewed as a $\mathrm{CSP}_{\leq 1}(\mathcal{A} \cup \mathcal{B})$ problem where $\mathcal{A}, \mathcal{B}$ have infinite signatures. This question is thus not directly answered by Theorem 14. Second, let us also remark that when considering $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$, we have assumed that both $\mathcal{A}$ and $\mathcal{B}$ are taken from some nice "superstructure". For example, in the equality language case we assume that both structures are first-order reducts of $(\mathbb{N}; =)$. One could choose structures more freely and, for example, let $\mathcal{A}$ be an equality language and $\mathcal{B}$ a finite-domain language. This calls for modifications of the underlying theory since (for instance) the algorithm that Theorem 8 is based on breaks down.

For finite domains we obtained a *coarse* parameterized dichotomy for $\mathrm{CSP}_{\leq}(\mathcal{A} \cup \mathcal{B})$ separating FPT from pNP-hardness. Sharper results providing the exact borderline between P and NP-hardness for the pNP-hard cases are required for classifying implication, equivalence, and redundancy. Via Theorem 7 and Theorem 10 the interesting case is when $\mathrm{CSP}(\mathcal{A})$ is in P, $\mathcal{A} \cup \mathcal{B}$ is core but $\mathcal{A}$ is not core. This question may be of independent algebraic interest and could be useful for other problems where the core property is not as straightforward as in the CSP case. For example, in *surjective* CSP we require the solution to be surjective, and this problem is generally hardest to analyze when the template is not a core [8].

Any complexity classification of the first-order reducts of a structure includes by necessity a classification of equality CSPs. Thus, our equality language classification lay the foundation for studying first-order reducts of more expressive structures. A natural step is to study *temporal languages*, i.e. first-order reducts of $(\mathbb{Q}; <)$. Our classification of equality constraint languages relies on the work in [4] via [19], who studied the clones of polymorphisms of equality constraint languages in more detail. One important result, due to Haddad & Rosenberg [15], is that after excluding several easy cases, every equality constraint language we end up with is only closed under operations with range $[c]$ for some constant c. Then, pp-defining the relation NEQ_{c+1} brings us into pNP-hard territory. Similar characterizations of the polymorphisms for reducts of other infinite structures, e.g. $(\mathbb{Q}; <)$, would imply corresponding pNP-hardness results, and this appear to be a manageable way forward.

References

1 Libor Barto, Andrei A. Krokhin, and Ross Willard. Polymorphisms, and how to use them. In *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

2 Manuel Bodirsky. *Complexity of Infinite-Domain Constraint Satisfaction*. Cambridge University Press, 2021.

3 Manuel Bodirsky and Hubie Chen. Qualitative temporal and spatial reasoning revisited. *Journal of Logic and Computation*, 19(6):1359–1383, 2009.

4 Manuel Bodirsky, Hubie Chen, and Michael Pinsker. The reducts of equality up to primitive positive interdefinability. *The Journal of Symbolic Logic*, 75(4):1249–1292, 2010. doi:10.2178/jsl/1286198146.

5 Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, pages 184–196, 2008.

6 Manuel Bodirsky and Peter Jonsson. A model-theoretic view on qualitative constraint reasoning. *Journal of Artificial Intelligence Research*, 58:339–385, 2017.

7 Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 43(2):136–158, 2008.

8 Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems - a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012.

9 Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. In *Proc. 17th International Workshop on Computer Science Logic (CSL-2003)*, pages 44–57, 2003.

10 Manuel Bodirsky and Stefan Wölfl. RCC8 is polynomial on networks of bounded treewidth. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, pages 756–761, 2011.

11 Elmar Böhler, Edith Hemaspaandra, Steffen Reith, and Heribert Vollmer. Equivalence and isomorphism for Boolean constraint satisfaction. In *Proc. 16th International Workshop on Computer Science Logic (CSL-2002)*, pages 412–426, 2002.

12 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS-2017)*, pages 319–330, 2017.

13 Hubie Chen. A rendezvous of logic, complexity, and algebra. *ACM SIGACT News*, 37(4):85–114, 2006.

14 David A. Cohen, Peter Jeavons, Peter Jonsson, and Manolis Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.

15 Lucien Haddad and Ivo G. Rosenberg. Finite clones containing all permutations. *Canadian Journal of Mathematics*, 46(5):951–970, 1994.

16 Peter G. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

17 Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both. On redundant topological constraints. *Artificial Intelligence*, 225:51–76, 2015.

18 Florent R. Madelaine and Barnaby Martin. On the complexity of the model checking problem. *SIAM Journal on Computing*, 47(3):769–797, 2018.

19 George Osipov and Magnus Wahlström. Parameterized complexity of equality MinCSP. *arXiv preprint arXiv:2305.11131*, 2023. This is the report version of a paper that appears in Proc. 31st Annual European Symposium on Algorithms (ESA-2023), pp. 86:1-86:17.

20 David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proc. 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR-1992)*, pages 165–176, 1992.

21 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on Theory of Computing (STOC-1978)*, pages 216–226, 1978.

22 Henning Schnoor and Ilka Schnoor. Partial polymorphisms and constraint satisfaction problems. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, pages 229–254. Springer, 2008.

23 Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proc. 14th Annual ACM Symposium on Theory of Computing (STOC-1982)*, pages 137–146, 1982.

24 Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM*, 67(5):30:1–30:78, 2020.

25 Dmitriy Zhuk. Constraint satisfaction problem: what makes the problem easy. In *Proc. International Conference of Mathematicians 2022 (ICM-2022)*, pages 1530–1553, 2022.