Complexity of Inverse Constraint Problems and a Dichotomy for the Inverse Satisfiability Problem*

Victor Lagerkvist†[1] and Biman Roy‡[2]

[1]Department of Computer and Information Science, Linköping University, Linköping, Sweden
[2]Independent researcher, Drabantgatan 5, Linköping, 58214, Östergötland, Sweden

Abstract

The *inverse satisfiability problem* over a set of relations Γ (Inv-SAT(Γ)) is the problem of deciding whether a relation R can be defined as the set of models of a SAT(Γ) instance. Kavvadias and Sideri (SIAM Journal on Computing, 28(1), 1998) obtained a dichotomy between P and co-NP-complete for finite Γ containing the two constant Boolean relations. However, for arbitrary constraint languages the complexity has been wide open, and in this article we finally prove a complete dichotomy theorem for finite languages. Kavvadias and Sideri's techniques are not applicable and we have to turn to the more recent algebraic approach based on *partial polymorphisms*. We also study the complexity of the inverse *constraint satisfaction problem* prove a general hardness result, which in particular resolves the complexity of INVERSE k-COLOURING, mentioned as an open problem in Chen (Computational Complexity, 17(1), 2008).

1 Introduction

A *constraint language* is a set of Boolean relations. The *parameterized satisfiability problem* over a constraint language Γ (SAT(Γ)) is the computational decision problem of determining whether a conjunctive formula over Γ is satisfiable. In a seminal paper by Schaefer it was proven that SAT(Γ) is always either tractable, i.e., polynomial-time solvable, or is NP-complete [24]; a property that should not be taken for granted in light of the NP-intermediate problems constructed by Ladner [16]. In this article we will study the computational complexity of the *inverse satisfiability problem* over a constraint language Γ (Inv-SAT(Γ)), which, as the name suggests, is the exact opposite of SAT(Γ). Hence, instead of a SAT(Γ) instance, we are given a relation R, and the question is then to determine if there exists an instance of SAT(Γ) with precisely R as its sets of models. In

*A preliminary version of this article appeared in Proceedings of the 37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017).

†victor.lagerkvist@liu.se
‡cse.biman@gmail.com

fact, for every problem in NP there exists a corresponding inverse problem, and we refer the reader to Chen [8] for a survey on this topic. Contrary to SAT(Γ), Inv-SAT(Γ) is in general co-NP-complete, and its computational complexity was studied by Kavvadias and Sideri [15]. While a complete dichotomy theorem was not obtained, Kavvadias and Sideri proved that for finite constraint languages Γ containing the constant relations $\{(0)\}$ and $\{(1)\}$, Inv-SAT(Γ) is always either tractable or co-NP-complete. We will strengthen this result and give a complete dichotomy theorem for Inv-SAT(Γ) for finite constraint languages, and thus solve a long-standing open problem. At a first glance, the condition that Γ contains the constant relations might only look like a minor technical difficulty, but there are several reasons why Inv-SAT(Γ) has previously escaped a complete complexity classification. First, for SAT and its multi-valued generalization CSP, it is known that the introduction of constant relations does not affect the complexity of the problem, provided that the constraint language satisfies the algebraic property of being a core. No such property is known to hold (a priori) for Inv-SAT(Γ), essentially making the cases that are normally trivial the most difficult to handle. Second, and perhaps most importantly, the majority of dichotomies for CSP and for Boolean problems parameterized by constraint languages, have been obtained via the so-called *algebraic approach*. For a thorough survey of this approach we refer the reader to Creignou et al. [10] and to Barto [2]. In short, the algebraic approach allows us to relate the complexity of a problem parameterized by a set of relations Γ to properties of the *polymorphisms* of Γ ($\mathrm{Pol}(\Gamma)$), which we may think of as a collection of functions preserving the structure of the relations in Γ. The main applicability of this connection is that sets of polymorphisms are well-studied and are in fact completely determined in the Boolean domain [21]. Hence, instead of directly reasoning by properties of constraint languages, it is possible to prove complexity results by exploiting properties of well-known polymorphisms. The Inv-SAT(Γ) problem, however, is fundamentally incompatible with polymorphisms, and instead we turn to the more refined concept of *partial polymorphisms* of Γ ($\mathrm{pPol}(\Gamma)$). The necessary algebraic background for these concepts will be formally defined in Section 2, and at the moment the most important observation is that there is a strong connection between $\mathrm{pPol}(\Gamma)$ and the expressive power of Γ with respect to conjunctive formulas called *quantifier-free primitive positive definitions* (qfpp-definitions). Since a SAT(Γ) instance consists of a conjunctive formula over Γ, we would therefore expect a correspondence between the complexity of Inv-SAT(Γ) and $\mathrm{pPol}(\Gamma)$. Unfortunately, this is not as straightforward as one might be led to believe, and we have only been able to find such a connection for certain classes of constraint languages Γ. For example, as we will prove, Inv-SAT(Γ) is polynomial-time reducible to Inv-SAT(Δ) if $\mathrm{pPol}(\Delta) \subseteq \mathrm{pPol}(\Gamma)$ but there does not exist Γ' such that $\mathrm{pPol}(\Delta) \subset \mathrm{pPol}(\Gamma') \subset \mathrm{pPol}(\Gamma)$.

Despite these complicating factors we are able to leverage partial polymorphisms in the context of Inv-SAT(Γ). This is accomplished by using algebraic techniques developed by Schnoor and Schnoor [25] and Lagerkvist [17], allowing us to classify the constraint languages under consideration according to their expressive power, in an extremely fine-grained way. These expressibility results turn out to be vital when we prove our dichotomy theorem for Inv-SAT(Γ) in Section 3. More precisely, our dichotomy result states that Inv-SAT(Γ) is co-NP-complete for finite Γ if the polymorphisms of Γ can be generated by a set of unary Boolean operations — a property which in the literature is also sometimes called *non-Schaefer*. This complexity classification in fact exactly coincides with the complexity of enumerating the solutions of SAT(Γ) with polynomial delay [11]. Using our dichotomy we can for example say that INVERSE 1-IN-k-SAT and INVERSE NAE-k-SAT are co-NP-complete; two natural problems that are missing from the aforementioned complexity result by Kavvadias and Sideri.

After having proven the dichotomy theorem for INV-SAT(Γ) for finite Γ we investigate the case when Γ is infinite in Section 4. For SAT(Γ), Schaefer's dichotomy theorem remain valid also for infinite languages, and given the similarity between SAT(Γ) and INV-SAT(Γ), one might conjecture that the same is also true for INV-SAT(Γ). Surprisingly, this turns out to be false: we show that there exists an infinite constraint language Γ such that (1) INV-SAT(Γ) is tractable, (2) SAT(Γ) is NP-hard, and (3) there exists finite $\Delta \subset \Gamma$ such that INV-SAT(Δ) is co-NP-complete. The existence of such a Γ is perhaps not so clear from the relational side, but becomes more evident when viewing INV-SAT(Γ) through the lens of pPol(Γ). We provide an algebraic criterion for this phenomena based on the expressive power of pPol(Γ), and conjecture that this property is both necessary and sufficient, in the sense that INV-SAT(Γ) is tractable if it holds and is co-NP-complete otherwise. Furthermore, we are not aware of any other problems parameterized by constraint languages where finiteness implies hardness and infiniteness implies tractability, in the above sense. In fact, problems where the complexity between finite and infinite languages does not coincide are extremely rare, and the closest example is a variant of the propositional abduction problem where it is known that there exists an infinite constraint language resulting in NP-hardness even though every finite subset results in tractability [13].

Last, in Section 5 we turn to the multi-valued generalization of SAT known as the *constraint satisfaction problem* over Γ (CSP(Γ)). This problem also admits a natural inverse problem INV-CSP(Γ) which has not been systematically studied before. Generalizing the constructions from Kavvadias and Sideri [15] we are able to prove that INV-CSP(Γ) is co-NP-complete if Pol(Γ) only contains trivial polymorphisms of the form $\pi(x_1, \ldots, x_i, \ldots, x_n) = x_i$. Extending this to other cases where CSP(Γ) is NP-complete does not appear to be straightforward, but we manage to use our general result to show that INV-CSP($\{\neq_D\}$), where $\neq_D$ is the binary inequality relation over a finite set D, is co-NP-complete. Furthermore, INV-CSP($\{\neq_D\}$) for $|D| = k$ is an alternative formulation of INVERSE k-COLOURING; a problem whose complexity status was left as an open question in Chen [8].

2 Preliminaries

A *Boolean relation* is a subset of $\{0,1\}^n$ for some $n \geq 1$, and if R is a relation we write $\mathrm{ar}(R)$ to denote its arity. For a tuple $t = (x_1, \ldots, x_n)$ we write $t[i]$ to denote the ith element x_i, and $\mathrm{Pr}_{i_1,\ldots,i_{n'}}(t) = (t[i_1], \ldots, t[i_{n'}])$ to denote the *projection* on the coordinates $i_1, \ldots, i_{n'} \in \{1, \ldots, n\}$. Similarly, for an n-ary relation R we let $\mathrm{Pr}_{i_1,\ldots,i_{n'}}(R) = \{\mathrm{Pr}_{i_1,\ldots,i_{n'}}(t) \mid t \in R\}$. We will typically use first-order logical formulas to define relations, and write

$$R(x_1, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n)$$

to define the relation

$$R = \{(f(x_1), \ldots, f(x_n)) \mid f \text{ is a model of } \varphi(x_1, \ldots, x_n)\}.$$

Let BR denote the set of all Boolean relations and $\Pi_{\mathbb{B}}$ the set of all Boolean projections, i.e., operations of the form $\pi_i^n(x_1, \ldots, x_i, \ldots, x_n) = x_i$.

A (not necessarily finite) $\Gamma \subseteq \mathsf{BR}$ is called a *Boolean constraint language*, or, if there is no risk for confusion, simply a constraint language. If $\{(0)\}, \{(1)\} \in \Gamma$ then we say that Γ is *ultraidempotent*. We prefer the term ultraidempotent over idempotent since the latter typically only requires that the constant relations are primitively positively definable (see Section 2.2 for a definition of this concept).

2.1 The Inverse Satisfiability Problem

The *parameterized satisfiability problem* over a constraint language Γ (SAT(Γ)) is the computational decision problem defined as follows.

> INSTANCE: A tuple (V, C) where V is a set of variables and C a set of constraint applications of the form $R(x_1, \ldots, x_{\mathrm{ar}(R)})$ where $R \in \Gamma$ and $x_1, \ldots, x_{\mathrm{ar}(R)} \in V$.
> QUESTION: Does there exist a function $f\colon V \to \{0,1\}$ such that $(f(x_1), \ldots, f(x_{\mathrm{ar}(R)})) \in R$ for every $R(x_1, \ldots, x_{\mathrm{ar}(R)}) \in C$?

If $\Gamma = \{R\}$ is singleton then we write SAT(R) instead of SAT(Γ).

Example 1. *Let $R_{1/3}$ be the ternary relation $\{(0,0,1),(0,1,0),(1,0,0)\}$. Then* SAT($R_{1/3}$) *can be viewed as a formulation of the* 1-IN-3-SAT *problem without negation, and is well-known to be NP-complete. More generally, if we for each $k \geq 3$ define $R_{1/k} = \{(x_1, \ldots, x_k) \in \{0,1\}^k \mid x_1 + \ldots + x_k = 1\}$ then* SAT($R_{1/k}$) *is an alternative formulation of* 1-IN-k-SAT *without negation.*

Depending on the constraint language in question many additional satisfiability problems may be viewed as SAT(Γ) *problems. For example, if we for $k \geq 1$ let Γ^k_{SAT} be the constraint language where each $R \in \Gamma^k_{\mathrm{SAT}}$ is the set of models of a k-clause, then* SAT(Γ^k_{SAT}) *is a natural formulation of the k-SAT problem.*

We will sometimes view a SAT(Γ) instance $(\{x_1, \ldots, x_n\}, C)$ as a conjunctive formula φ and write $Sols(\varphi)$ to denote its set of models. Note that $Sols(\varphi)$ is not formally a relation, but can easily be treated as a relation of arity n by ordering the variables $x_1, \ldots, x_n$ and treating each model as an n-ary tuple. The *inverse satisfiability problem* over a constraint language Γ (INV-SAT(Γ)) can then be viewed as the problem of, given a relation R, determining whether there exists a SAT(Γ) instance with precisely R as it set of models. More precisely, the problem is defined as follows.

> INSTANCE: A Boolean relation R.
> QUESTION: Does there exist a SAT(Γ) instance φ such that $Sols(\varphi) = R$?

If this question can be answered in polynomial time with respect to the number of bits required to represent R then we say that INV-SAT(Γ) is tractable. For simplicity, we will represent the input relation R as a list of tuples. In general the INV-SAT(Γ) problem is co-NP-complete and a dichotomy theorem is known for finite and ultraidempotent constraint languages Γ [15].

Theorem 1. *Let Γ be a finite and ultraidempotent Boolean constraint language. Then* INV-SAT(Γ) *is either co-NP-complete or tractable.*

Example 2. *Consider the relation $R_{1/3}$ from Example 1. Then* Inv-SAT($R_{1/3}$) *is the problem of, given a relation R, deciding if there exists a 1-in-3-SAT instance without negation with exactly R as its set of models. Since $\{R_{1/3}\}$ is not ultraidempotent we cannot however use Theorem 1 to conclude that* Inv-SAT($R_{1/3}$) *is co-NP-complete. We will return to this problem in Section 3 where we prove our dichotomy theorem for* Inv-SAT(Γ)*.*

2.2 Closure Operators on Relations

In this section we introduce two closure operators on sets of relations that will be important when explaining the algebraic approach in the forthcoming section. First, if R is an n-ary Boolean relation and Γ a constraint language we say that R has a *primitive positive definition* over Γ if

$$R(x_1, \ldots, x_n) \equiv \exists y_1, \ldots, y_{n'} \colon R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m),$$

where each $R_i \in \Gamma \cup \{(0,0),(1,1)\}$ and each $\mathbf{x}_i$ is a tuple of variables over $x_1, \ldots, x_n$, $y_1, \ldots, y_{n'}$ of length $\mathrm{ar}(R_i)$. In other words R is definable over Γ by a (possibly) existentially quantified, conjunctive formula of constraints over Γ and the equality relation $\{(0,0),(1,1)\}$. Given a constraint language Γ we then write $\langle \Gamma \rangle$ to denote the smallest set of relations containing Γ and which is closed under taking pp-definition. Sets of the form $\langle \Gamma \rangle$ are called *relational clones* or *co-clones*.

Similarly, say that an n-ary Boolean relation has a *quantifier-free primitive positive definition* (qfpp-definition) over a constraint language Γ if

$$R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m),$$

where each $R_i \in \Gamma \cup \{(0,0),(1,1)\}$ and each $\mathbf{x}_i$ is a tuple of variables over $x_1, \ldots, x_n$ of length $\mathrm{ar}(R_i)$. Let $\langle \Gamma \rangle_{\nexists}$ denote the smallest set of relations containing Γ and which is closed under taking qfpp-definitions. If $\Gamma = \{R\}$ is singleton then we for simplicity write $\langle R \rangle_{\nexists}$ instead of $\langle \Gamma \rangle_{\nexists}$. These sets are usually called *weak systems* or *weak partial co-clones*. We remark that there is a very strong connection between Inv-SAT(Γ) and the set $\langle \Gamma \rangle_{\nexists}$. To see this, note that an instance of Inv-SAT(Γ) is simply a relation R, and the question of whether there exists an instance φ of SAT(Γ) with $Sols(\varphi) = R$, can be rephrased as whether R admits a qfpp-definition over Γ, i.e., $R \in \langle \Gamma \rangle_{\nexists}$. Whenever convenient we will therefore assume that Inv-SAT(Γ) is the problem of checking whether $R \in \langle \Gamma \rangle_{\nexists}$. We remark that the related problem of checking whether R admits a pp-definition over Γ is tractable for Boolean Γ [9] but co-NEXPTIME-hard for sufficiently large, but finite, domains [26].

2.3 Closure Operators on Operations

Let $f : \{0,1\}^k \to \{0,1\}$ be a k-ary Boolean operation and R an n-ary Boolean relation. We say that f *preserves* R, that f is a *polymorphism* of R, or that R is *invariant* under f, if $f(t_1, \ldots, t_k) \in R$ for every sequence of tuples $t_1, \ldots, t_k \in R$, where

$$f(t_1, \ldots, t_k) = (f(t_1[1], \ldots, t_k[1]), \ldots, (f(t_1[n], \ldots, t_k[n]))).$$

We write $\mathrm{Pol}(R)$ for the set of polymorphisms of the relation R and if Γ is a constraint language we let $\mathrm{Pol}(\Gamma) = \bigcap_{R \in \Gamma} \mathrm{Pol}(R)$. Sets of the form $\mathrm{Pol}(\Gamma)$ are usually called *clones* and are known to be sets of operations containing all projections (i.e., $\Pi_{\mathbb{B}} \subseteq \mathrm{Pol}(\Gamma)$) and closed under composition (i.e., if $f, g_1, \ldots, g_m \in \mathrm{Pol}(\Gamma)$ where f has arity m and each g_i arity n then the n-ary operation $f \circ g_1, \ldots, g_m(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ is included in $\mathrm{Pol}(\Gamma)$). We let $[F]$ be the smallest clone containing the set F, i.e., the intersection of all clones containing F.

We will now describe a powerful connection between clones and co-clones. First, if we let Inv(F) be the set of all relations invariant under the set of operations F, it is known that Inv(F) is in fact closed under pp-definitions, i.e., is a co-clone. Second, for any constraint language Γ it is known that $\mathrm{Inv}(\mathrm{Pol}(\Gamma)) = \langle \Gamma \rangle$, and that for any set of operations F, $\mathrm{Pol}(\mathrm{Inv(F)}) = [\mathrm{F}]$. This results in the following inverse relationship between $\mathrm{Inv}(\cdot)$ and $\mathrm{Pol}(\cdot)$.

Theorem 2 ([4, 5, 12])**.** *Let Γ and Γ' be two constraint languages. Then $\Gamma \subseteq \langle \Gamma' \rangle$ if and only if $\mathrm{Pol}(\Gamma') \subseteq \mathrm{Pol}(\Gamma)$.*

There is a similar connection between weak systems and sets of *partial operations*. Formally, we view a (Boolean) partial operation f of arity k as a mapping $X \to \{0,1\}$ where $X \subseteq \{0,1\}^k$ is called the *domain* of f and denoted by $\mathrm{domain}(f)$. Then a k-ary partial operation f is said to be a *partial polymorphism* of an n-ary relation R if either $f(t_1, \ldots, t_k) \in R$ or there exists $1 \leq i \leq n$ such that $(t_1[i], \ldots, t_k[i]) \notin \mathrm{domain}(f)$, for every sequence $t_1, \ldots, t_k \in R$. We write pPol(R) for the set of all partial polymorphisms of R and $\mathrm{pPol}(\Gamma)$ for the set $\bigcap_{R \in \Gamma} \mathrm{pPol}(\Gamma)$. These sets are usually referred to as *strong partial clones* and are known to be sets of partial operations containing all projections, closed under composition, and closed under taking subfunctions. More precisely, composition of partial operations is defined in exactly the same way as composition of total operations, but the resulting partial operation is only defined for a sequence of arguments if every partial operation in the composition is defined; and by closed under taking subfunctions we mean that if $f \in \mathrm{pPol}(\Gamma)$ then $g \in \mathrm{pPol}(\Gamma)$ for every g such that $\mathrm{domain}(g) \subseteq \mathrm{domain}(f)$ and such that g agrees with f for the tuples in $\mathrm{domain}(g)$. We write $[F]_s$ for the smallest strong partial clone containing F, and say that $[F]_s$ is *finitely generated* if there exists finite $G \subseteq [F]_s$ such that $[F]_s = [G]_s$, is infinitely generated otherwise, and in both cases we say that G is a *base* of $[F]_s$. For additional background concerning these concepts we refer the reader to Lau [20]. The reason why we define these technical concepts will be explained in Section 4 where we study the complexity of $\textsc{Inv-SAT}(\Gamma)$ when $\mathrm{pPol}(\Gamma)$ is finitely generated.

Similar to the total case, if we let Inv(F) be the set of relations invariant under the set of partial operations F, then it is known that Inv(F) is closed under qfpp-definitions, and is therefore a weak system. Moreover, $\langle \Gamma \rangle_{\nexists} = \mathrm{Inv}(\mathrm{pPol}(\Gamma))$ and $[F]_s = \mathrm{pPol}(\mathrm{Inv(F)})$. We then have the following connection between $\mathrm{Inv}(\cdot)$ and $\mathrm{pPol}(\cdot)$, due to Geiger [12] and Romov [22].

Theorem 3 ([12, 22])**.** *Let Γ and Γ' be two constraint languages. Then $\Gamma \subseteq \langle \Gamma' \rangle_{\nexists}$ if and only if $\mathrm{pPol}(\Gamma') \subseteq \mathrm{pPol}(\Gamma)$.*

Using the concepts in this section we can now present the dichotomy theorem from Kavvadias and Sideri [15] more precisely as follows.

Theorem 4. *Let Γ be a finite and ultraidempotent constraint language. Then $\textsc{Inv-SAT}(\Gamma)$ is co-NP-complete if $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$ and is tractable otherwise.*

We remark that the tractable cases in Theorem 4 stem from the observation that if one can enumerate the solutions of $\mathrm{SAT}(\Gamma)$ with polynomial delay, then $\textsc{Inv-SAT}(\Gamma)$ must be tractable. To see this, let R be an instance of Inv-SAT(Γ). We begin by computing a qfpp-definition φ over Γ with the property that $Sols(\varphi) \supseteq R$. This can be accomplished by letting φ be the conjunction of *all* possible Γ-constraints over $\text{ar}(R) = n$ variables not contradicting R, and since we have at most $O(n^r)$ such constraints, where $r = \max_{S \in \Gamma} \text{ar}(S)$, φ can be constructed in polynomial time. Then it is sufficient to enumerate at most $|R| + 1$ solutions to φ (viewed as an instance of SAT(Γ)) and stop if any of these solutions do not match the tuples in R.

3 A Dichotomy Theorem for Inv-SAT(Γ)

In this section we will extend Theorem 4 to finite constraint languages that are not necessarily ultraidempotent, in order to obtain a complete dichotomy theorem for Inv-SAT(Γ). First observe that the tractable cases of Theorem 4 remain valid even if Γ is not ultraidempotent since solutions to SAT(Γ) can be enumerated with polynomial delay in those cases [11]. To better describe the remaining cases we will need to define the following Boolean operations.

Definition 5. *We define the following Boolean operations.*

1. $f_0(x) = 0$,
2. $f_1(x) = 1$,
3. $\overline{x} = 1 - x$

Then, using the terminology from Böhler et al. [6, 7], $[\{f_0, f_1, \overline{x}\}] = \mathsf{N}$, $[\{f_0, f_1\}] = \mathsf{I}$, $[\{f_0\}] = \mathsf{I}_0$, $[\{f_1\}] = \mathsf{I}_1$, $[\{\overline{x}\}] = \mathsf{N}_2$, and $[\{\pi^1_1\}] = \mathsf{I}_2 = \Pi_{\mathbb{B}}$. Our aim is now to prove the following theorem, which is visualized in Figure 1.

Theorem 6. *Let* Γ *be a finite constraint language. Then* Inv-SAT(Γ) *is co-NP-complete if* $\text{Pol}(\Gamma) \subseteq [F]$ *for* $F \subseteq \{f_0, f_1, \overline{x}\}$ *and is tractable otherwise.*

The intuition behind the theorem is that one cannot enumerate the solutions of SAT(Γ) with polynomial delay if $\text{Pol}(\Gamma) \subseteq [F]$ for $F \subseteq \{f_0, f_1, \overline{x}\}$, unless P = NP [11]. However, before we turn to these proofs, it might be helpful to review how dichotomy theorems for problems parameterized by Boolean constraint languages are usually obtained. Hence, assume that $X(\Gamma)$ is a computational decision problem for which it is true that $X(\Gamma)$ admits a polynomial-time reduction to $X(\Delta)$ whenever $\text{Pol}(\Delta) \subseteq \text{Pol}(\Gamma)$. Then, what one needs to do is simply to take every clone Pol(Γ) in Post's lattice [21] and determine the complexity of $X(\Gamma)$, since the results then automatically carry over to every $X(\Delta)$ such that $\text{Pol}(\Delta) = \text{Pol}(\Gamma)$. This is e.g. the case for SAT and many Boolean optimization and logical reasoning problems [10]. For the Inv-SAT(Γ) problem we do not have such a result, implying that the proof strategy for Theorem 6 intrinsically will be more complex. We will show that this difficulty can be mitigated by using properties of weak systems in Section 3.1, and in Section 3.2 we use these results in order to prove Theorem 6.

3.1 Algebraic Simplifications

Even though we do not have a general reduction result for Inv-SAT(Γ) it is still possible to provide a characterization of the qfpp-definable relations over a constraint language Γ.

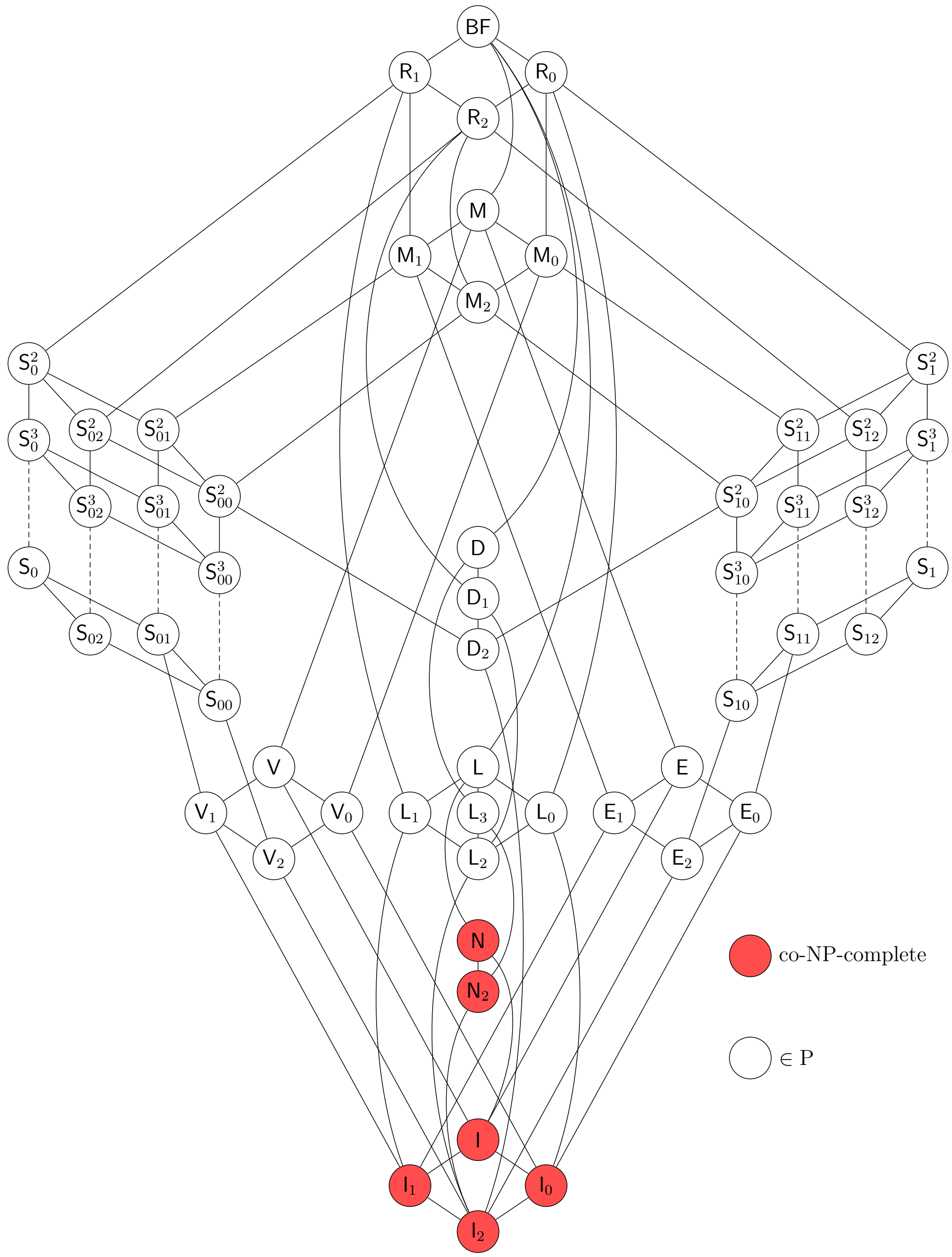

Figure 1: The complexity of INV-SAT(Γ) for finite Γ.

The following lemma is helpful in the sense that it for each distinct case precisely describes the relations that can be freely employed when constructing qfpp-definitions.

Lemma 7. *Let* $\mathrm{Pol}(\Gamma) \subseteq [F]$ *for* $F \subseteq \{f_0, f_1, \bar{x}\}$. *Then*

1. $\tau^{01} = \{(0,1)\}, \tau^{01}_{\neq} = \{(0,1,0,1),(1,0,0,1)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$,
2. $\tau_{\neq} = \{(0,1),(1,0)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{\bar{x}\}]$,
3. $\tau_{f_0,f_1,\bar{x}} = \{(0,0,0,0),(1,1,1,1),(0,1,0,1),(1,0,0,1),(1,0,1,0),(0,1,1,0)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{f_0, f_1, \bar{x}\}]$,
4. $\tau_{\rightarrow} = \{(0,0),(1,0),(1,1)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{f_0, f_1\}]$,
5. $\tau^{01}_{\neq} \cup \{(0,0,0,0)\} = \{(0,0,0,0),(0,1,0,1),(1,0,0,1)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{f_0\}]$, *and*
6. $\tau^{01}_{\neq} \cup \{(1,1,1,1)\} = \{(0,1,0,1),(1,0,0,1),(1,1,1,1)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{f_1\}]$.

Proof. We consider each case in turn. The various cases follow a similar structure and make use of the algebraic machinery developed by Schnoor and Schnoor [25] and Lagerkvist [17]. We first remark that for $\mathrm{Pol}(\Gamma) \in \{[\{f_0\}], [\{f_1\}], [\{f_0, f_1, \bar{x}\}]\}$ the relations follow immediately from Theorem 11 in Lagerkvist [17]. Hence, the remaining cases are when $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$, $\mathrm{Pol}(\Gamma) = [\bar{x}]$, and $\mathrm{Pol}(\Gamma) = [\{f_0, f_1\}]$. First assume that $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$. From Lagerkvist [17] we know that $R^{\neq\neq\neq 01}_{1/3} \in \langle\Gamma\rangle_{\nexists}$ where

$$R^{\neq\neq\neq 01}_{1/3} = \{(0,0,1,1,1,0,0,1),(0,1,0,1,0,1,0,1),(1,0,0,0,1,1,0,1)\},$$

and using this relation we can qfpp-define $\tau^{01}_{\neq}$ as

$$\tau^{01}_{\neq}(x_1,x_2,x_3,x_4) \equiv R^{\neq\neq\neq 01}_{1/3}(x_1,x_2,x_3,x_2,x_1,x_4,x_3,x_4)$$

and τ^{01} as $\tau^{01}(x_1,x_2) \equiv \tau^{01}_{\neq}(x_1,x_2,x_1,x_2)$. Now assume that $\mathrm{Pol}(\Gamma) = [\{\bar{x}\}]$. In this case it is known that the relation $R^{\neq\neq\neq\neq}_{2/4} = R^{\neq\neq\neq 01}_{1/3} \cup \{\bar{t} \mid t \in R^{\neq\neq\neq 01}_{1/3}\}$ is qfpp-definable by Γ [14, 17]. Using this relation one can verify that $\tau_{\neq}(x_1,x_2) \equiv R^{\neq\neq\neq\neq}_{2/4}(x_1,x_1,x_2,x_2,x_2,x_1,x_1,x_2)$. Last, for $\mathrm{Pol}(\Gamma) = [\{f_0, f_1\}]$, the relation $R = \{(0,0,0,0),(0,0,1,1),(0,1,0,1),(1,1,1,1)\} \in \langle\Gamma\rangle_{\nexists}$ [17], and this relation can qfpp-define $\tau_{\rightarrow}$ by $\tau_{\rightarrow}(x_1,x_2) \equiv R(x_1,x_1,x_2,x_2)$. $\square$

For example, if $\mathrm{Pol}(\Gamma) = [\{\bar{x}\}]$ then we know that Γ is expressive enough to qfpp-define the binary inequality relation $\tau_{\neq}$. Before we begin to prove Theorem 6 in the forthcoming section we present a lemma that simplifies some of the arguments. If R is an n-ary relation then the ith argument is *redundant* if there exists $j \neq i$ such that $t[i] = t[j]$ for every $t \in R$, and R is said to be *irredundant* if it does not have any redundant arguments. It is not difficult to see that there for any R exists an irredundant relation R^{irr} with the property that $\langle R\rangle_{\nexists} = \langle R^{\mathrm{irr}}\rangle_{\nexists}$, and we obtain the following lemma.

Lemma 8. *Let* Γ *be a constraint language and* R *an* n*-ary relation. Then* $R \in \langle\Gamma\rangle_{\nexists}$ *if and only if* $R^{\mathrm{irr}} \in \langle\Gamma\rangle_{\nexists}$.

3.2 The Hard Cases

In this section we handle the various hardness proofs necessary for proving Theorem 6. The various cases will be be proved in Lemma 9, Lemma 10, Lemma 13, and Lemma 15. In these reductions we need the ability to output an arbitrary yes- or no-instance of INV-SAT(Γ). Clearly, a yes-instance can easily be produced by simply outputting $R \in \Gamma$, but to find $R \notin \langle \Gamma \rangle_{\nexists}$ requires a bit more work. We will provide a proof sketch for how such a relation can be constructed. Begin by enumerating all partial polymorphisms of Γ up to arity $k+1$, where k is the maximum arity of any relation in Γ. It is well-known that any finite Boolean constraint language containing only relations of arity k contains a partial polymorphism of arity at most $k+1$ which is not a partial projection [19]. Hence, let f denote such a partial polymorphism of arity $n \leq k+1$, and let $\mathrm{domain}(f) = \{t_1, \ldots, t_m\}$. Now consider the relation R obtained by for each $1 \leq i \leq n$ adding the tuple $(t_1[i], \ldots, t_m[i])$. By construction, f does not preserve R since it is not a subfunction of a projection, which by Theorem 3 implies that $R \notin \langle \Gamma \rangle_{\nexists}$.

With this observation we are now ready to prove our first hardness result, and begin with the case when $\mathrm{Pol}(\Gamma)$ consists only of projections (which due to Theorem 2 implies that Γ can pp-define every Boolean relation).

Lemma 9. *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$*. Then* INV-SAT(Γ) *is co-NP-complete.*

Proof. First consider the constraint language

$$\Delta = \{\tau \times \{(0,1)\} \mid \tau \in \Gamma\} \cup \{\{(0)\}, \{(1)\},$$

i.e., each relation in Γ is adjoined with two constant arguments, and in addition Δ contains both $\{(0)\}$ and $\{(1)\}$. Since Δ is ultraidempotent and $\mathrm{Pol}(\Delta) = \mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$ it follows from Theorem 4 that INV-SAT(Δ) is co-NP-complete, and we will therefore prove NP-hardness of INV-SAT(Γ) by reducing from INV-SAT(Δ).

Hence, let R be an n-ary relation, i.e., an instance of INV-SAT(Δ). For simplicity we assume that R does not contain any redundant arguments, which we without loss of generality may assume by Lemma 8. The basic idea behind the reduction is now to perform a case analysis on the number of constant arguments in the relation R, which influences the existence of a qfpp-definition of R over Δ or Γ. For example, if there exists $1 \leq i, j \leq n$ such that $\mathrm{Pr}_{i,j}(R) = \{(0,1)\}$, then any qfpp-definition of R over Δ can be assumed to contain the two constraints $\{(0)\}(x_i)$ and $\{(1)\}(x_j)$.

Let us now proceed as follows. If there exists $1 \leq i \leq n$ such that $\mathrm{Pr}_i(R) = \{(0)\}$ but no $1 \leq j \leq n$ such that $\mathrm{Pr}_j(R) = \{(1)\}$, then $R \in \langle \Delta \rangle_{\nexists}$ if and only if $R \in \langle \{(0)\} \rangle_{\nexists}$. Similarly, if there exists $1 \leq i \leq n$ such that $\mathrm{Pr}_i(R) = \{(1)\}$ but no $1 \leq j \leq n$ such that $\mathrm{Pr}_j(R) = \{(0)\}$, then $R \in \langle \Delta \rangle_{\nexists}$ if and only if $R \in \langle \{(1)\} \rangle_{\nexists}$. In both these cases we can compute the answer in polynomial time and output an arbitrary yes- or no-instance to INV-SAT(Γ). Similarly, if R does not contain any constant argument, i.e., there does not exist $1 \leq i \leq n$ where $|\mathrm{Pr}_i(R)| = 1$, then $R \notin \langle \Delta \rangle_{\nexists}$ and we output an arbitrary no-instance.

Hence, assume that there exist both i and j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$. We then claim that $R \in \langle \Delta \rangle_{\nexists}$ if and only if $R \in \langle \Gamma \rangle_{\nexists}$. Hence, first assume that $R \in \langle \Delta \rangle_{\nexists}$, and let

$$R(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n) \wedge \{(0)\}(x_i) \wedge \{(1)\}(x_j)$$

be a qfpp-definition witnessing this, where we without loss of generality assume that every constraint in $\varphi(x_1, \ldots, x_n)$ is of the form $\tau_k \times \{(0,1)\}(\mathbf{x}_k, x_i, x_j)$ for $\tau_k \times \{(0,1)\} \in \Delta$, where $\mathbf{x}_k$ is a tuple of variables of length $\mathrm{ar}(\tau_k)$ not containing x_i or x_j. Then we may obtain a qfpp-definition of R over Γ by first replacing $\{(0)\}(x_i) \wedge \{(1)\}(x_j)$ by the single constraint $\{(0,1)\}(x_i, x_j)$, and then replacing every constraint $\tau_k \times \{(0,1)\}(\mathbf{x}_k, x_i, x_j)$ in $\varphi(x_1, \ldots, x_n)$ by $\tau_k(\mathbf{x}_k)$. This is clearly a valid qfpp-definition of R over Γ since $\{(0,1)\} \in \langle\Gamma\rangle_{\nexists}$ by Lemma 7. For the other direction, assume that $R \in \langle\Gamma\rangle_{\nexists}$ and let

$$R(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n)$$

be a qfpp-definition of R over Γ, where every constraint in $\varphi(x_1, \ldots, x_n)$ is of the form $\tau(\mathbf{x})$ for $\tau \in \Gamma$. Then we can construct a qfpp-definition of R over Δ by first introducing the constraints $\{(0)\}(x_i)$ and $\{(1)\}(x_j)$, and then replacing every $\tau_k(\mathbf{x}_k)$ by $\tau_k \times \{(0,1)\}(\mathbf{x}_k, x_i, x_j)$. □

Note that $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$ for every remaining case. Hence, equipped with Lemma 9 we now have a canonical problem to reduce from.

Lemma 10. *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = [\{\bar{x}\}]$. *Then* INV-SAT(Γ) *is co-NP-complete.*

Proof. We will give a polynomial-time reduction from INV-SAT($\Gamma \cup \{(0,1)\}$), which is co-NP-complete by Lemma 9, since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$. Hence, let R be an n-ary relation, i.e, an instance of INV-SAT($\Gamma \cup \{(0,1)\}$). If there exist neither i nor j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$ then $R \in \langle\Gamma\rangle_{\nexists}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\nexists}$, and the output of the reduction is simply R. Furthermore, if there exists $1 \leq i \leq n$ such that $\mathrm{Pr}_i(R) = \{(0)\}$ but no $1 \leq j \leq n$ such that $\mathrm{Pr}_j(R) = \{(1)\}$, or vice versa, then it cannot be the case that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\nexists}$ or $R \in \langle\Gamma\rangle_{\nexists}$, which again implies that we may simply output R. This implies that the only remaining case is when there exist both i and j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$. By Lemma 8 we may without loss of generality assume that R does not contain any other constant arguments.

In this case we claim that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\nexists}$ if and only if $(R \cup \neg(R)) \in \langle\Gamma\rangle_{\nexists}$, where $\neg(R) = R \cup \{\bar{t} \mid t \in R\}$. In other words the relation $R \cup \neg(R)$ consists of R and the complement of each tuple in R. Assume first that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\nexists}$ and let

$$R(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i, x_j)$$

denote a qfpp-definition over $\Gamma \cup \{(0,1)\}$, where $R_1, \ldots, R_m \in \Gamma$. We will construct a qfpp-definition of $(R \cup \neg(R))$ over Γ as follows. First, we replace $\{(0,1)\}(x_i, x_j)$ by the constraint $\tau_{\neq}(x_i, x_j)$, which is qfpp-definable over Γ by Lemma 7. Then every other constraint is kept unchanged and we obtain the qfpp-definition

$$R'(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \tau_{\neq}(x_i, x_j)$$

over Γ. We claim that $R' = (R \cup \neg(R))$. It is easy to see that $(R \cup \neg(R)) \subseteq R'$. Hence, let $t \in R'$, and note that $t[i] \neq t[j]$ holds due to the constraint $\tau_{\neq}(x_i, x_j)$ in the above qfpp-definition. If $t[i] = 0$ and $t[j] = 1$ then $t \in R \subseteq R \cup \neg(R)$. Otherwise, $t[i] = 1$ and $t[i] = 0$, and it must then instead hold that $t \in \neg(R) \subseteq R \cup \neg(R)$. For the other direction, assume that $(R \cup \neg(R)) \in \langle\Gamma\rangle_{\nexists}$ and let $(R \cup \neg(R))(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$ be a qfpp-definition over Γ where $R_1, \ldots, R_m \in \Gamma$. We can then qfpp-define R using Γ and $\{(0,1)\}$ as $R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i, x_j)$, since this only keeps $t \in (R \cup \neg(R))$ satisfying $t[i] = 0$ and $t[j] = 1$. $\square$

For the case when Γ is preserved by a constant operation it is not easy to directly reduce from INV-SAT($\Gamma \cup \{(0,1)\}$), and we will first need to show co-NP-completeness of an auxiliary problem, where input relations satisfy the following additional property.

Definition 11. *An n-ary Boolean relation R is* complementary saturated *if (1) there exists $1 \leq i, j \leq n$ such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$, and (2) for every $1 \leq i \leq n$ there exists $1 \leq j \leq n$ such that $t[i] \neq t[j]$ for every $t \in R$.*

In other words the relation contains the two constant Boolean arguments, and for each argument of the relation there exists an argument which is its complement. For a finite constraint language Γ we then by INV-SAT$^{\neq}(\Gamma)$ denote the following structurally restricted INV-SAT(Γ) problem.

INSTANCE: A complementary saturated Boolean relation R.
QUESTION: $R \in \langle\Gamma\rangle_{\nexists}$?

We will now prove that INV-SAT$^{\neq}(\Gamma)$ remains co-NP-complete when $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$.

Lemma 12. *Let Γ be a finite constraint language such that $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$. Then INV-SAT$^{\neq}(\Gamma)$ is co-NP-complete.*

Proof. We will first construct the language $\Gamma^{\neq}$ for every $R \in \Gamma$ by letting $R^{\neq} \in \Gamma^{\neq}$ where $R^{\neq}$ is obtained by adding the minimum number of arguments to R such that $R^{\neq}$ is complementary saturated. Without loss of generality we assume that the arguments to each relation $R^{\neq} \in \Gamma^{\neq}$ are ordered such that $\mathrm{Pr}_{1,\ldots,\mathrm{ar}(R)}(R^{\neq}) = R$. Observe that $\mathrm{Pol}(\Gamma^{\neq}) = \Pi_{\mathbb{B}}$, which by Lemma 9 implies that INV-SAT($\Gamma^{\neq}$) is co-NP-complete, and that we can prove the claim of the lemma by reducing from INV-SAT($\Gamma^{\neq}$).

Hence, let R be an n-ary relation, which we for simplicity assume is irredundant by Lemma 8. Assume that R is not complementary saturated and is not a valid instance of INV-SAT$^{\neq}(\Gamma)$. Then $R \notin \langle\Gamma^{\neq}\rangle_{\nexists}$ since every qfpp-definition over $\Gamma^{\neq}$ defining an irredundant relation can only use constraints over complementary saturated relations from $\Gamma^{\neq}$. Hence, we may simply output an arbitrary no-instance.

Otherwise, R is already a valid instance of INV-SAT$^{\neq}(\Gamma)$, and in this case we claim that $R \in \langle\Gamma^{\neq}\rangle_{\nexists}$ if and only if $R \in \langle\Gamma\rangle_{\nexists}$. First assume that $R \in \langle\Gamma^{\neq}\rangle_{\nexists}$. Via Lemma 7 we know that $\tau_{\neq}^{01} \in \langle\Gamma\rangle_{\nexists}$, and from this property it follows that $\Gamma^{\neq} \subseteq \langle\Gamma\rangle_{\nexists}$, implying that $R \in \langle\Gamma\rangle_{\nexists}$. For the other direction, assume that $R \in \langle\Gamma\rangle_{\nexists}$. Let

$$R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$$

be a qfpp-definition over Γ, and for each tuple of variables $\mathbf{x}_i$ let $\mathbf{y}_i$ denote the corresponding tuple of complementary variables. It then follows that

$$R(x_1, \ldots, x_n) \equiv R_1^{\neq}(\mathbf{x}_1, \mathbf{y}_1) \wedge \ldots \wedge R_m^{\neq}(\mathbf{x}_m, \mathbf{y}_m)$$

is a valid qfpp-definition of R over $\Gamma^{\neq}$. $\square$

We can now handle the remaining cases by reducing from $\textsc{Inv-SAT}^{\neq}(\Gamma \cup \{(0,1)\})$ instead of $\textsc{Inv-SAT}(\Gamma \cup \{(0,1)\})$, which is significantly easier.

Lemma 13. *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = [\{f_0\}]$ *or* $\mathrm{Pol}(\Gamma) = [\{f_1\}]$. *Then* $\textsc{Inv-SAT}(\Gamma)$ *is co-NP-complete.*

Proof. We present the proof for the case when $\mathrm{Pol}(\Gamma) = [\{f_0\}]$ since the other case is entirely analogous. In order to prove this we will give a polynomial-time reduction from $\textsc{Inv-SAT}^{\neq}(\Gamma \cup \{(0,1)\})$ to $\textsc{Inv-SAT}(\Gamma)$. The problem $\textsc{Inv-SAT}^{\neq}(\Gamma \cup \{(0,1)\})$ is co-NP-complete by Lemma 12 since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$.

Let R be an instance of $\textsc{Inv-SAT}^{\neq}(\Gamma \cup \{(0,1)\})$ of arity n. From Lemma 8 we may in addition assume that R is irredundant. If there does not exist $i, j \in \{1, \ldots, n\}$ such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$ then it is already the case that $R \in \langle \Gamma \rangle_{\nexists}$ if and only if $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$; therefore we assume that such i and j exist. First construct the relation $R' = R \cup \{(0, \ldots, 0)\}$, i.e., the relation R adjoined with the constant 0 tuple. We will now prove that $R' \in \langle \Gamma \rangle_{\nexists}$ if and only if $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$. Hence, first assume that $R' \in \langle \Gamma \rangle_{\nexists}$ and let

$$R'(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$$

be a qfpp-definition over Γ. Then consider the qfpp-definition

$$R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i, x_j).$$

This qfpp-definition is correct since the additional constraint $\{(0,1)\}(x_i, x_j)$ will ensure that the constant 0 tuple, included in R' but not in R, cannot be a model. For the other direction assume that $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$ and let

$$R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i, x_j)$$

denote a qfpp-definition, where we without loss of generality assume that $R_1, \ldots, R_m$ belong to Γ. Now recall the relation

$$\tau_{\neq}^{01} \cup \{(0,0,0,0)\} = \{(0,1,0,1), (1,0,0,1), (0,0,0,0)\}$$

from Lemma 7, and observe that this relation is nothing else than the binary inequality relation with two constant arguments, adjoined with the constant 0 tuple. We will use this relation as a gadget in order to enforce that the correct inequalities hold between the complementary variables. Hence, assume that the arity of R is $2k+2$, that the variables occurring in positions $k+1, \ldots, 2k$ are the complement of the k first, and that the last two arguments are constant 0 and constant 1, respectively. These assumptions can be made without loss of generality since R is irredundant. Then consider the qfpp-definition

$$R''(x_1, \ldots, x_{2k}, x_{2k+1}, x_{2k+2}) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \bigwedge_{i=1}^{k} \tau_{\neq}^{01} \cup \{(0,0,0,0)\}(x_i, x_{i+k}, x_{2k+1}, x_{2k+2}).$$

We claim that $R'' = R'$. The direction $R' \subseteq R''$ is clear, and we therefore concentrate on proving that $R'' \subseteq R'$. Consider a tuple $t \in R''$. If $t[i] \neq t[i]$ for each $1 \leq i \leq k$

then $t[2k+1] = 0$ and $t[2k+2] = 1$, and $t \in R \subseteq R'$. Otherwise there exists i where $t[i] = t[i+k]$. However, $t[i] = t[i+k] = 1$ cannot happen since such a tuple is inconsistent with the constraint $\tau_{\neq}^{01} \cup \{(0,0,0,0)\}(x_i, x_{i+k}, x_{2k+1}, x_{2k+2})$ in the above definition. Hence, $t[i] = t[i+k] = 0$. But then $t[2k+2] = 0$, which in turn implies that $t[j] = t[j+k] = 0$ for each $1 \leq j \leq k$, and that t is the constant 0 tuple $(0, \ldots, 0) \in R'$. □

Lemma 14. *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = [\{f_0, f_1\}]$. *Then* INV-SAT($\Gamma$) *is co-NP-complete.*

Proof. In order to prove the result we will give a polynomial-time reduction from INV-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$, which is co-NP-complete since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$. Hence, let R be an n-ary relation. Since R is complementary saturated there exists $1 \leq i, j \leq n$ such that $\mathrm{Pr}_i = \{(0)\}$ and $\mathrm{Pr}_j = \{(1)\}$. Construct the relation $R' = R \cup \{(0, \ldots, 0), (1, \ldots, 1)\}$. We claim that $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$ if and only if $R' \in \langle \Gamma \rangle_{\nexists}$. For the first direction, assume that $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$ and let

$$R(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i, x_j)$$

denote a qfpp-definition such that $R_1, \ldots, R_m \in \Gamma$. Recall that the relation $\tau_{\rightarrow} = \{(0,0), (0,1), (1,1)\}$ from Lemma 7 is qfpp-definable by Γ. Now construct the qfpp-definition

$$R'(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \bigwedge_{k=1}^{n} (\tau_{\rightarrow}(x_i, x_k) \wedge \tau_{\rightarrow}(x_k, x_j).$$

To see that this definition is correct, observe that the additional constraints of the form $(\tau_{\rightarrow}(x_i, x_k) \wedge \tau_{\rightarrow}(x_k, x_j)$ ensure that either x_i and x_j are assigned 0 and 1, respectively, or every variable is assigned 0 or 1, resulting in the two constant tuples $(0, \ldots, 0)$ and $(1, \ldots, 1)$. The other direction ($R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$ if $R' \in \langle \Gamma \rangle_{\nexists}$) can be proven using similar arguments as in the proof of Lemma 13. □

Lemma 15. *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = [\{f_0, f_1, \bar{x}\}]$. *Then* INV-SAT($\Gamma$) *is co-NP-complete.*

Proof. As $\mathrm{Pol}(\Gamma) = [\{f_0, f_1, \bar{x}\}]$ it follows that $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$. We will give a polynomial-time reduction from INV-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$ to INV-SAT(Γ) (INV-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$ is co-NP-complete since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$). Let R be an instance of INV-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$. Since R is complementary saturated there exist i and j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$. For simplicity we will also assume that R is irredundant, $n = 2k+2$, $i = 2k+1$, $j = 2k+2$, and that the arguments in positions $k+1, \ldots, 2k$ are the complement of the k first. Construct the relation

$$R' = R \cup \{\bar{t} | t \in R\} \cup \{(0, \ldots, 0), (1, \ldots, 1)\}.$$

We will prove that $R' \in \langle \Gamma \rangle_{\nexists}$ if and only if $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$. Therefore, first assume that $R \in \langle \Gamma \cup \{(0,1)\} \rangle_{\nexists}$ and let

$$R(x_1, \ldots, x_{2k+2}) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_{2k+1}, x_{2k+2})$$

be a qfpp-definition over Γ where $R_1, \ldots, R_m \in \Gamma$. Now consider the qfpp-definition

$$R''(x_1, \ldots x_{2k+2}) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \bigwedge_{l \in \{1,\ldots,n\}} \tau_{f_0,f_1,\overline{x}}(x_l, x_{l+k}, x_{2k+1}, x_{2k+2}),$$

where $\tau_{f_0,f_1,\overline{x}} = \{(0,0,0,0),(1,1,1,1),(0,1,0,1),(1,0,0,1),(1,0,1,0),(0,1,1,0)\} \in \langle\Gamma\rangle_{\nexists}$ is the relation from Lemma 7. We claim that $R' = R''$, i.e., then the above qfpp-definition defines R'. It is clearly the case that $R \subseteq R''$, and this also implies that $R' \subseteq R''$ since R'' is closed under f_0, f_1, and $\overline{x}$. For the other direction, assume there exists $t \in R'' \setminus R'$. It must then be the case that t is not constant 0 or constant 1, and furthermore also that $\bar{t} \notin R'$. Assume first that there exists $1 \leq l \leq k$ such that $t[l] = t[l+k]$. Then, due to the constraints

$$\bigwedge_{l \in \{1,\ldots,n\}} \tau_{f_0,f_1,\overline{x}}(x_l, x_{l+k}, x_{2k+1}, x_{2k+2}),$$

it is easy to verify that this will force $t[2k+1] = t[2k+2] = t[l]$, which in turn implies that $t[l] = t[l']$ for every $1 \leq l' \leq 2k+2$, and also that $t \in R$. This contradicts the assumption, and we conclude (1) that $t[l] \neq t[l+k]$ for every $1 \leq l \leq k$ and (2) that $t[2k+1] = 0$ and $t[2k+2] = 1$ or $t[2k+1] = 1$ and $t[2k+2] = 0$. In the first case it directly follows that $t \in R \subseteq R'$, and in the second case that $\bar{t} \in R$, and hence that $t \in R'$.

To prove the reverse direction we assume that

$$R'(x_1, \ldots x_{2k+2}) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$$

where $R_1, \ldots, R_m \in \Gamma$. We can then qfpp-define R by

$$R(x_1, \ldots x_{2k+2}) \equiv R'(x_1, \ldots x_{2k+2}) \wedge \{(0,1)\}(x_{2k+1}, x_{2k+2}).$$

Hence, $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\nexists}$ if and only if $R' \in \langle\Gamma\rangle_{\nexists}$. This concludes the reduction. $\square$

By combining Lemma 9– 10 and Lemma 13– 15 we have thus finally proven Theorem 6.

Example 3. *We can now answer the question regarding the complexity of* INV-SAT($R_{1/3}$) *from Example 2. Any existing reduction from* 3*-SAT to* 1-IN-3-SAT *can be converted to a pp-definition of* Γ^3_{SAT} *over* $R_{1/3}$ *implying that* $R_{1/3}$ *is preserved only by the projections. Hence,* $\mathrm{Pol}(R_{1/3}) = \Pi_{\mathbb{B}}$*, and Theorem 6 then reveals that* INV-SAT($R_{1/3}$) *is indeed co-NP-complete. More generally Theorem 6 also implies that* INV-SAT($R_{1/k}$) *is co-NP-complete for every* $k \geq 3$*. Another, perhaps more illuminating example, is the inverse problem for* NOT-ALL-EQUAL-k-SAT*. This problem can be formulated as* INV-SAT(Γ^k_{NAE}) *where* $\Gamma^k_{\text{NAE}} = \{0,1\}^k \setminus \{(0,\ldots,0),(1,\ldots,1)\}$*.It is known that* Γ^k_{NAE} *for* $k \geq 3$ *is only closed under complement [9], so using Theorem 6 we conclude that* INV-SAT(Γ^k_{NAE}) *is co-NP-complete for every* $k \geq 3$*. Note that this case is entirely absent from Theorem 4 since* Γ^k_{NAE} *cannot even pp-define the constant relations.*

By Theorem 6 we know that INV-SAT(Γ) is polynomial-time reducible to INV-SAT(Δ) if $\Gamma \subseteq \langle\Delta\rangle_{\nexists}$ (indeed, even $\Gamma \subseteq \langle\Delta\rangle$) for finite languages Γ and Δ. When such a relationship holds it is sometimes said to hold *a posteriori* for the problem in question. However, proving that INV-SAT(Γ) is polynomial-time reducible to INV-SAT(Δ) if $\Gamma \subseteq \langle\Delta\rangle_{\nexists}$ unconditionally of Theorem 6 appears to be highly challenging. We may state this conjecture as follows.

Conjecture 1. *Let Γ and Δ be two finite, Boolean constraint languages. If $\Gamma \subseteq \langle \Delta \rangle_{\nexists}$ then* Inv-SAT(Γ) *is polynomial-time many-one reducible to* Inv-SAT(Δ) *(independent of Theorem 6).*

Thus far we have been unable to make significant progress. However, it is possible to give important special cases where such a result is true.

Theorem 16. *Let Γ and Δ be two finite, Boolean constraint languages. If there exist partial operations $g_1, \ldots, g_m$ such that $\mathrm{pPol}(\Gamma) = [\mathrm{pPol}(\Delta) \cup \{\mathrm{g_1, \ldots, g_m}\}]_\mathrm{s}$ then* Inv-SAT(Γ) *is polynomial-time many-one reducible to* Inv-SAT(Δ).

Proof. Let R be an n-ary Boolean relation. We first check if $g_i \in \mathrm{pPol(R)}$ for every $1 \leq i \leq n$. This can be done in polynomial time since m is a fixed constant. If no, then $R \notin \langle \Gamma \rangle_{\nexists}$ (Theorem 3) and the result of the reduction is an arbitrary no-instance of Inv-SAT(Δ). Hence, assume that every g_i preserves R. In this case we claim that $R \in \langle \Gamma \rangle_{\nexists}$ if and only if $R \in \langle \Delta \rangle_{\nexists}$. First, assume that $R \in \langle \Gamma \rangle_{\nexists}$. Then $R \in \langle \Delta \rangle_{\nexists}$ since $\Gamma \subseteq \langle \Delta \rangle_{\nexists}$ from the assumption that $\mathrm{pPol}(\Gamma) = [\mathrm{pPol}(\Delta) \cup \{\mathrm{g_1, \ldots, g_m}\}]_\mathrm{s}$ and Theorem 3. Second, assume that $R \in \langle \Delta \rangle_{\nexists}$, and, thus, that $\mathrm{pPol}(\Delta) \subseteq \mathrm{pPol(R)}$. But then $\mathrm{pPol}(\Gamma) \subseteq \mathrm{pPol(R)}$, too, since (1) $g_1, \ldots, g_m \in \mathrm{pPol(R)}$ and (2) $\mathrm{pPol}(\Gamma) = [\mathrm{pPol}(\Delta) \cup \{\mathrm{g_1, \ldots, g_m}\}]_\mathrm{s}$. We conclude that $R \in \langle \Gamma \rangle_{\nexists}$ if and only if $R \in \langle \Delta \rangle_{\nexists}$, and the output of the reduction is simply R itself. □

This, for example, holds in the important special case when $\langle \Delta \rangle_{\nexists}$ *covers* $\langle \Gamma \rangle_{\nexists}$, i.e., when there does not exist Γ' such that $\langle \Gamma \rangle_{\nexists} \subset \langle \Gamma' \rangle_{\nexists} \subset \langle \Delta \rangle_{\nexists}$. Generalizing Theorem 16 to arbitrary Γ and Δ such that $\langle \Gamma \rangle_{\nexists} \subseteq \langle \Delta \rangle_{\nexists}$ appears difficult. The fundamental problem is that, even if we were given an infinite set $G = \{g_1, g_2, \ldots\}$ of partial operations such that $\mathrm{pPol}(\Gamma) = [\mathrm{pPol}(\Delta) \cup \mathrm{G}]_\mathrm{s}$, the condition $G \subseteq \mathrm{pPol(R)}$ cannot necessarily be checked in polynomial time.

4 The Inv-SAT(Γ) Problem over Infinite Constraint Languages

Since we have proven that Inv-SAT(Γ) is always either tractable or co-NP-complete for finite constraint languages Γ, it is tempting to investigate the case when Γ is infinite. First, it is important to note that Schaefer's dichotomy theorem for SAT(Γ) is also valid for infinite constraint languages, and in fact that many natural satisfiability problems such as CNF-SAT, Horn-SAT, and linear equations modulo 2, can only be represented as SAT(Γ) problems over infinite Γ. It thus makes sense to ask whether it is possible to extend Theorem 6 to infinite constraint languages.

First, note that if SAT(Γ) is NP-complete then SAT(Δ) is NP-complete whenever $\Delta \subseteq \Gamma$. This straightforward property does *not* hold for Inv-SAT(Γ), since, for example, Inv-SAT($\{R_{1/3}\}$) is co-NP-complete but Inv-SAT(BR) is trivially solvable in polynomial time by always answering "yes". We will now describe a more general class of tractable Inv-SAT(Γ) problems based on properties of the partial polymorphisms of Γ. Here, we need the property that $\mathrm{pPol}(\Gamma)$ is finitely generated, i.e., that there exists a finite set of partial operations F such that $[F]_s = \mathrm{pPol}(\Gamma)$.

Theorem 17. *Let Γ be a constraint language such that* $\mathrm{pPol}(\Gamma)$ *is finitely generated with respect to a set F of partial operations. Then* Inv-SAT(Γ) *is solvable in polynomial time.*

Proof. Let R be an instance of Inv-SAT(Γ) of arity n. Due to Theorem 3 the question $R \in \langle\Gamma\rangle_{\nexists}$ is equivalent to checking whether $F \subseteq \mathrm{pPol(R)}$, or, put otherwise, whether R is preserved by every partial operation in F. Now consider the following algorithm.

1. Let k be the maximum arity among the partial operations in F.
2. For each $1 \leq i \leq k$ enumerate all sequences $t_1, \ldots, t_i \in R$.
3. For each $f \in F$ of arity i compute $f(t_1, \ldots, t_i) = t$. If $t \notin R$ then answer "no".
4. Answer "yes".

As remarked, this algorithm is sound and complete since $R \in \langle\Gamma\rangle_{\nexists}$ if and only if every $f \in F$ preserves R, and an i-ary partial operation f preserves R if and only if there does not exist $t_1, \ldots, t_i \in R$ such that $f(t_1, \ldots, t_i) \notin R$. Regarding the time complexity, we in the ith iteration enumerate all sequences of tuples from R of length i, which takes $O(|R|^i)$ time, and for each $f \in F$ check whether f applied to this sequence results in a tuple included in R, which takes $O(i \cdot n \cdot |R|)$ time. Put together this gives a running time of $O(k \cdot |F| \cdot |R|^k \cdot k \cdot n \cdot |R|) = O(k^2 \cdot |F| \cdot |R|^{k+1} \cdot n)$ which is bounded by a polynomial since k is a fixed constant. $\square$

It is worth remarking that Γ is *always* infinite when $\mathrm{pPol}(\Gamma)$ is finitely generated and $\mathrm{Pol}(\Gamma) \supseteq [\{f_0, f_1, \overline{x}\}]$ [18] — hence there is no possible overlap between Theorem 6 and Theorem 17. This result may be seen as surprising since computational problems parameterized by Boolean constraint languages tend to be rather well-behaved. To the best of our knowledge only a variant of the propositional abduction problem, exhibits a similar difference in complexity between finite and infinite constraint languages [13], but with the crucial distinction that the finite language results in tractability while the infinite language results in NP-hardness.

At this stage it is fair to ask if Inv-SAT(Γ) is always tractable when Γ is infinite. This is however not the case. First take any finite constraint language Γ such that Inv-SAT(Γ) is co-NP-complete by Theorem 6. Then consider the infinite constraint language $\langle\Gamma\rangle_{\nexists}$ obtained by closing Γ under qfpp-definitions. Clearly, Inv-SAT(Γ) and Inv-SAT($\langle\Gamma\rangle_{\nexists}$) are the same computational problem, and in particular Inv-SAT($\langle\Gamma\rangle_{\nexists}$) is co-NP-complete even though $\langle\Gamma\rangle_{\nexists}$ is infinite. Based on these observations and Theorem 17, it is natural to conjecture that the question of whether Inv-SAT(Γ) is co-NP-complete or tractable does not depend on whether Γ is finite or infinite, but rather whether $\mathrm{pPol}(\Gamma)$ is sufficiently simple. We thus make the following conjecture.

Conjecture 2. *Let Γ be a Boolean constraint language such that* $\mathrm{Pol}(\Gamma) \supseteq [\{f_0, f_1, \overline{x}\}]$. *Then* Inv-SAT($\Gamma$) *is tractable if* $\mathrm{pPol}(\Gamma)$ *is finitely generated and is co-NP-hard otherwise.*

In general terms the situation is therefore as follows. There exists infinite F such that Inv-SAT(Inv(F)) is co-NP-complete (simply by letting $F = \mathrm{pPol}(\Gamma)$ for any finite Γ where $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$) even though Inv-SAT(Inv(G)) is tractable for every finite subset $G \subset F$. The reader familiar with Ladner's technique for constructing NP-intermediate problems [16] will at this stage likely see some resemblances. Ladner's technique is based on "blowing holes" in the set of instances of a computational problem in such a way that the resulting problem is too dense to be tractable but too sparse too be intractable. The technique has also been generalized to blow holes in constraint languages in order to construct NP-intermediate (infinite-domain) CSPs and NP-intermediate abduction problems [13]. Thus, it is tempting, although not straightforward, to apply this technique to Inv(F) and attempt to construct $F' \subset F$ which is not finite but such that Inv-SAT(Inv(F)) is not polynomial-time reducible to Inv-SAT(Inv(F′)).

5 The Inverse Constraint Satisfaction Problem

In this section we investigate the complexity of a multi-valued generalization of Inv-SAT(Γ) based on the *constraint satisfaction problem* (CSP). This problem can be defined analogously to SAT(Γ) with the exception that the constraint language Γ is allowed to contain non-Boolean relation. Hence, if D is a domain and Γ is a constraint language over D, then an instance of the CSP problem over Γ (CSP(Γ)) is a tuple (V, C) consisting of a set of variables V and a set of constraints C over V and Γ, and the objective is to determine if there exists $f \colon V \to D$ satisfying all constraints. Similarly to the Boolean case it is possible to associate an instance (V, C) of CSP(Γ) with a logical formula φ, and we write $Sols(\varphi)$ for the set of all models of φ. We then define the *inverse constraint satisfaction problem* over a constraint language Γ (Inv-CSP(Γ)) over a domain D as follows.

Instance: A relation R over D.
Question: Does there exist a CSP(Γ) instance φ such that $Sols(\varphi) = R$?

All of the algebraic techniques introduced in Section 2 are valid for arbitrary finite domains in the context of CSPs, and we refrain from defining them again. Thus, for example, if Γ is a constraint language over D then we write Pol(Γ) for the set of polymorphisms over D, and Pol(Γ) is again called a clone. In addition, if $\Gamma = \{R\}$ is singleton then we write CSP(R) instead of CSP($\{R\}$).

Example 4. *The k-colouring problem is a well-known problem of determining if a graph (V, E) can be coloured using at most k colours. This problem can be easily formulated as a CSP as follows. Let $D = \{0, 1, \ldots, k-1\}$ and define the inequality relation over D as $\neq_D = \{(x, y) \in D^2 \mid x \neq y\}$. Then we create an instance of* CSP($\neq_D$) *by viewing V as a set of variables, and for each $\{x, y\} \in E$ we introduce a constraint $\neq_D (x, y)$. It is then straightforward to verify that the resulting* CSP($\neq_D$) *instance admits a solution if and only if (V, E) admits a k-colouring.*

The k-colouring problem also has a natural inverse problem inverse k-colouring: given set of k-colourings over n vertices, determine if there exists a graph over n vertices which is colourable precisely according to this set. By the preceding example it is clear that the inverse k-colouring problem can be formulated as Inv-CSP($\neq_D$)*. This problem was mentioned as an open problem in Chen [8], and as we will see later in this section, inverse k-colouring is in P for $k = 2$ but is co-NP-complete for every $k \geq 3$.*

Let us now proceed by studying the complexity of Inv-CSP(Γ). Inclusion in co-NP can easily be proved using similar arguments as in the Boolean domain [15], i.e., a no-instance

$R \notin \langle\Gamma\rangle_{\nexists}$ can be witnessed by a formula φ and a tuple t where (1) φ is the conjunction of all Γ-constraints over $\mathrm{ar}(R)$ variables consistent with each model in R, and (2) $t \in Sols(\varphi) \setminus R$.

Lemma 18. *Let Γ be a finite constraint language over a finite domain D. Then* Inv-CSP(Γ) *is included in co-NP.*

Before we can state our hardness results we need a few technical definitions. Let R be a k-ary relation. Say that the argument $i \in \{1,\ldots,k\}$ is *determined* in R if $|\mathrm{Pr}_{1,\ldots,i-1,i+1,\ldots,k}(R)| = |R|$. Similarly, if

$$\exists y_1,\ldots,y_i,\ldots,y_l\colon \varphi(x_1,\ldots,x_k,y_1,\ldots,y_i,\ldots,y_l)$$

is a pp-definition with free variables $x_1,\ldots,x_k$ we say that y_i is determined by $x_1,\ldots,x_k$ if the $(k+1)$th argument is determined in the relation R defined by

$$R(x_1,\ldots,x_k,y_i) \equiv \exists y_1,\ldots,y_{i-1},y_{i+1},\ldots,y_l\colon \varphi(x_1,\ldots,x_k,y_1,\ldots,y_l).$$

Definition 19. *Let Γ be a constraint language over a finite domain. We say that Γ* faithfully defines *a k-ary relation R with a if there exists a pp-definition $R(x_1,\ldots,x_k) \equiv \exists y_1,\ldots,y_l\colon \varphi(x_1,\ldots,x_k,y_1,\ldots,y_l)$ over Γ where each y_i is determined by $x_1,\ldots,x_n$.*

Hence, existential quantification is only allowed if the variable is determined by the free variables in the pp-definition. Using algebra we can prove that any constraint language Γ over a finite domain whose only polymorphisms are the projections, can faithfully define every Boolean relation. Kavvadias and Sideri [15] proved this for the Boolean domain using a complicated case analysis over all possible Boolean formulas. Thus, the following result is a significant strengthening.

Lemma 20. *Let Γ be a constraint language over a finite domain D preserved only by the projections over D. Then Γ can faithfully define all Boolean relations.*

Proof. The idea behind the proof is that first we create a relation R such that $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(R)$ and which is qfpp-definable by Γ. Then we prove that this particular relation R can faithfully define all the Boolean relations. Let $D = \{0,1,\ldots,k-1\}$ for $k > 2$. We begin by constructing the relation $R = \{t_1,\ldots,t_k\}$ of arity k^k consisting of k tuples $t_1,\ldots,t_k$ such that there for each $t \in \{0,1,\ldots,k-1\}^k$ exists a unique $i \in \{1,\ldots,k^k\}$ such that $(t_1[i],\ldots,t_k[i]) = t$. Next, we claim that $\mathrm{Pol}(R)$ consists only of projections. Otherwise, by Rosenberg's classification of minimal clones [23], it follows that $[\{g\}] \subseteq [\{f\}]$ where g is either a unary operation which is not a projection, an idempotent binary operation, a ternary majority operation, a ternary minority operation, or an m-ary semiprojection, for $3 \leq m \leq k$. For each such operation g of arity $1 \leq n \leq k$ there then exists $t_1,\ldots,t_n \in R$ such that there for every $t \in D^n$ exists i such that $(t_1[i],\ldots,t_n[i]) = t$. Furthermore, since g is not a projection, $g(t_1,\ldots,t_n) \notin \{t_1,\ldots,t_n\}$, and it is straightforward to also show that $g(t_1,\ldots,t_n) \notin R$. Hence, $\mathrm{Pol}(R) = \mathrm{Pol}(\Gamma)$. Theorem 4.11 in Schnoor and Schnoor [25] then implies that, for this particular relation R, $R \in \langle\Gamma\rangle_{\nexists}$, which also implies that Γ faithfully defines R since a qfpp-definition is a restricted case of a faithful definition. Next, it is clear that there exist (distinct) $i_1,\ldots,i_k \in \{1,\ldots,k^k\}$ such that $\mathrm{Pr}_{i_1,\ldots,i_k}(R) = R_{1/k}$. Moreover, since $|R_{1/k}| = k$ it follows that every other argument in R is determined by $i_1,\ldots,i_k$, and R can therefore faithfully define $R_{1/k}$. Then it follows that Γ faithfully defines R, which faithfully defines $R_{1/k}$ which in turn faithfully defines every Boolean relation by Theorem 3 in Kavvadias and Sideri [15]. $\square$

In order to prove co-NP-hardness of Inv-CSP(Γ) we will modify the proof from Kavvadias and Sideri [15], and will need the following two definitions.

Definition 21. *Let $I = (V, C)$ be an instance of* CSP(Γ) *over D, and let $f\colon V' \to D$, $V' \subseteq V$, be a partial function. Say that f is* consistent *with I if there for every $R_i(x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R_i)}}) \in C$ exists an extension of f satisfying $R_i(x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R_i)}})$.*

Thus, a consistent (partial) assignment of variables in a CSP instance does not directly contradict any single constraint in the instance.

Definition 22. *Let Γ be a finite constraint language over a finite domain which faithfully defines all Boolean relations. Let $I = (V, C)$ be an instance of* $\mathrm{SAT}(\Gamma^3_{\mathrm{SAT}})$ *where $|V| = n$. Let $x_1, \ldots, x_n$ be an enumeration of V. For $r \geq 1$ we define the relation γ^r_I as follows.*

1. *Let $(W_1, f_1), \ldots, (W_k, f_k)$ be an enumeration such that $W_i \subseteq V$, $|W_i| = 3r$, and $f_i\colon W_i \to \{0, 1\}$ a partial function which is consistent with I.*
2. *Given (W_i, f_i) and a variable $x \in V$ we let*

$$\tau_{W_i,f_i}(x) = \begin{cases} (0, 1, \overbrace{0, \ldots, 0}^{k}) & \text{if } x \in W_i, f_i(x) = 1 \\ (1, 0, \overbrace{0, \ldots, 0}^{k}) & \text{if } x \in W_i, f_i(x) = 0 \\ (0, 0, \overbrace{1, \ldots, 1}^{i-1}, \underbrace{0, \ldots, 0}_{k-i+1}) & \text{if } x \notin W_i. \end{cases} \tag{1}$$

3. *β_{W_i,f_i} is the concatenation of $\tau_{W_i,f_i}(x_1), \ldots, \tau_{W_i,f_i}(x_n)$.*
4. *Let (V', C') be the instance of* $\mathrm{SAT}(\Gamma^3_{\mathrm{SAT}})$ *consisting of all possible constraints (i.e., 3-clauses) not contradicted by any β_{W_i,f_i} (viewed as a model). In particular, $Sols((V', C')) \supseteq \{\beta_{W_1,f_1}, \ldots, \beta_{W_k,f_k}\}$.*
5. *Let*

$$Sols((V', C'))(x_1, \ldots, x_{n(k+2)}) \equiv \exists y_1, \ldots, y_r \colon \varphi(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r)$$

denote a faithful definition over Γ.

6. *Let γ^r_I be the relation*

$$\gamma^r_I(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r) \equiv \varphi(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r).$$

Although not obvious from the definition, $|\gamma^r_I| = k$, since each β_{W_i,f_i} can be uniquely matched with the freshly introduced arguments, and γ^r_I can thus for each fixed $r \geq 1$ be constructed in polynomial time with respect to I. In order to lift the result from Kavvadias and Sideri [15] we will also need the following lemma, relating the expressive power of finite constraint languages to the expressive power of Γ^k_{SAT}.

Lemma 23. *Let Γ be a constraint language over a finite domain D where $\max_{S \in \Gamma} \mathrm{ar}(S) = r - 1$, and let R be an r-ary Boolean relation. If $R \notin \langle \Gamma^{r-1}_{\mathrm{SAT}} \rangle_{\nexists}$ then $R \notin \langle \Gamma \rangle_{\nexists}$.*

Proof. For a relation R over D let $R_{|\mathbb{B}} = \{t \mid t \in R, t \in \{0,1\}^{\mathrm{ar}(R)}\} \subseteq R$ be the largest subset of R which is Boolean, and let $\Gamma_{|\mathbb{B}} = \{R_{|\mathbb{B}} \mid R \in \Gamma\}$. Now, let R be an r-ary Boolean relation such that $R \notin \langle \Gamma^{r-1}_{\mathrm{SAT}} \rangle_{\nexists}$, and assume that R is qfpp-definable over Γ with a qfpp-definition $R(x_1, \ldots, x_r) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$ for $R_1, \ldots, R_m \in \Gamma \cup \{\mathrm{Eq_D}\}$ ($\mathrm{Eq_D}$ is the equality relation $\{(x,x) \mid x \in D\}$). Let $R'(x_1, \ldots, x_r) \equiv R_{1_{|\mathbb{B}}}(\mathbf{x}_1) \wedge \ldots \wedge R_{m_{|\mathbb{B}}}(\mathbf{x}_m)$. We claim that $R = R'$. Since each $R_{i_{|\mathbb{B}}} \subseteq R_i$ it directly follows that $R' \subseteq R$, and we only need to show that $t \in R \setminus R'$ cannot exist. If $t \in R \setminus R'$ exists then there must exist $R_{i_{|\mathbb{B}}}(\mathbf{x}_i)$, $\mathbf{x}_i = (x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R_i)}})$, such that $\mathrm{Pr}_{i_1, \ldots, i_{\mathrm{ar}(R_i)}}(t) \notin R_{i_{|\mathbb{B}}}$, which is clearly impossible since t is Boolean and since $R_{i_{|\mathbb{B}}}$ is the largest Boolean subset of R_i. It is then known that $\langle \Delta \rangle_{\nexists} \subseteq \langle \Gamma^{r-1}_{\mathrm{SAT}} \rangle_{\nexists}$ for any Boolean Δ such that $\max_{R \in \Delta} \mathrm{ar}(R) = r - 1$ [9], which in particular implies that $\Gamma_{|\mathbb{B}} \subseteq \langle \Gamma^{r-1}_{\mathrm{SAT}} \rangle_{\nexists}$ and that $R \in \langle \Gamma^{r-1}_{\mathrm{SAT}} \rangle_{\nexists}$. □

We can then prove co-NP-hardness with a reduction from the 3-UNSATISFIABILITY problem, i.e., the co-NP-complete problem of checking if an instance of $\mathrm{SAT}(\Gamma^3_{\mathrm{SAT}})$ is unsatisfiable.

Theorem 24. *Let Γ be a finite constraint language over a finite domain $D = \{0, 1, \ldots, k - 1\}$. If Γ is preserved only by the projections over D then* INV-CSP$(\Gamma \cup \{\{(0)\}, \{(1)\}, \ldots, \{(k - 1)\}\})$ *is co-NP-complete.*

Proof. Membership in co-NP follows from Lemma 18. Let $r = \max_{R \in \Gamma} \mathrm{ar}(R)$. To prove co-NP-hardness we will give a reduction from 3-UNSATISFIABILITY. Hence, let $I = (V, C)$ be an instance of 3-UNSATISFIABILITY. We then construct the relation γ^r_I from Definition 22, which is well-defined since Γ can faithfully define all Boolean relations by Lemma 20. Let k be the value from Definition 22. We claim that $\gamma^r_I \in \langle \Gamma \rangle_{\nexists}$ if and only if I is unsatisfiable.

First assume that I is satisfiable. Then $\gamma^r_I \notin \langle \Gamma^r_{\mathrm{SAT}} \rangle_{\nexists}$ by Theorem 4 in Kavvadias and Sideri [15], and Lemma 23 then implies that $\gamma^r_I \notin \langle \Gamma \cup \{\{(0)\}, \{(1)\}, \ldots, \{(k-1)\}\} \rangle_{\nexists}$.

Second, assume that I is unsatisfiable. Let $\gamma = \mathrm{Pr}_{1, \ldots, n(k+2)}(\gamma^r_I)$. Then γ can be qfpp-defined by Γ^3_{SAT} by Kavvadias and Sideri [15][Theorem 4], implying that $\mathit{Sols}((V', C')) = \gamma = \{\beta_{W_1, f_1}, \ldots, \beta_{W_k, f_k}\}$ for the $\mathrm{SAT}(\Gamma^3_{\mathrm{SAT}})$ instance (V', C') from step (4) in Definition 22. By recapitulating step (5) and step (6) in Definition 22 we then see that

$$\gamma^r_I(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r) \equiv \varphi(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r)$$

where $\varphi(x_1, \ldots, x_{n(k+2)}, y_1, \ldots, y_r)$ is a qfpp-definition of $\mathit{Sols}((V', C')) = \gamma$ over Γ. Hence, $\gamma^r_I \in \langle \Gamma \cup \{\{(0)\}, \{(1)\}, \ldots, \{(k-1)\}\} \rangle_{\nexists}$. □

It is straightforward to prove that Theorem 24 can be leveraged to show co-NP-completeness for INV-CSP(Γ) when $\mathrm{Pol}(\Gamma)$ contains only projections.

Theorem 25. *Let Γ be a finite constraint language over a finite domain D, preserved only by the projections over D. Then* INV-CSP(Γ) *is co-NP-complete.*

Proof. Since the proof is similar to Lemma 9, handling the Boolean case, we show the construction in detail but only provide a sketch of the correctness proof. Let $D = \{0, 1, \ldots, k - 1\}$ and construct the constraint language

$$\Delta = \{R \times \{(0, 1, \ldots, k-1)\} \mid R \in \Gamma\} \cup \{\{(0)\}, \{(1)\}, \ldots, \{(k-1)\}\}).$$

The problem INV-CSP(Δ) is co-NP-complete by Theorem 24, and it is therefore sufficient to give a reduction from INV-CSP(Δ) to INV-CSP(Γ).

Hence, let R be an n-ary, irredundant instance of INV-CSP(Δ). Similar to the Boolean case we can first check if $R \in \langle \chi \rangle_{\not\exists}$ for each $\chi \subseteq \{\{(0)\}, \{(1)\}, \ldots, \{(k-1)\}\}$, and if this is the case for some χ we then check if $\chi \subseteq \langle \Gamma \rangle_{\not\exists}$ and either output R itself or a no-instance of INV-CSP(Γ). In other words we for each subset χ of the constant relations over D check whether the relation R can be qfpp-defined with χ, and if χ is not qfpp-definable by Γ then we need to output a no-instance. These checks can be performed in polynomial time since $|D| = k$ is fixed and since Γ is finite.

If, on the other hand, $R \notin \langle \chi \rangle_{\not\exists}$ for every $\chi \subseteq \{\{(0)\}, \{(1)\}, \ldots, \{(k-1)\}\}$, then we check whether there exists $i_1, \ldots, i_k \in \{1, \ldots, n\}$ such that $\mathrm{Pr}_{i_1,\ldots,i_k}(R) = \{(0, 1, \ldots, k-1)\}$. If yes, then the output of the reduction is R itself, and if not then we output an arbitrary no-instance. The reason for outputting a no-instance is simply that any relation which does not contain all the constant arguments over D cannot be defined by a qfpp-definition using a relation from $\{R \times \{(0, 1, \ldots, k-1)\} \mid R \in \Gamma\} \subseteq \Delta$. Correctness then follows from recapitulating the proof of Lemma 9, and the observation that $\{(0, 1, \ldots, k-1)\} \in \langle \Gamma \rangle_{\not\exists}$ [1, Theorem 3.6]. $\square$

To exemplify this result we will prove that INVERSE k-COLOURING problem mentioned in Example 4 is co-NP-complete for every $k \geq 3$ (for $k = 2$ or $k = 1$ tractability follows from Theorem 6).

Theorem 26. INVERSE k-COLOURING *is co-NP-complete for* $k \geq 3$.

Proof. Recall that INVERSE k-COLOURING can be realized as INV-CSP($\neq_D$) for $D = \{0, \ldots k-1\}$ and $\neq_D = \{(x, y) \in D^2 \mid x \neq y\}$. Furthermore it is known that $\mathrm{Pol}(\{\neq_D, \{(0)\}, \ldots, \{(k-1)\})$ consists only of projections [1]. Theorem 24 then implies that INV-CSP($\{\neq_D, \{(0)\}, \ldots, \{(k-1)\}$) is co-NP-complete, and Theorem 25 furthermore implies that INV-CSP($\{\neq_D, \{(0, 1, \ldots, k-1)\}$) is co-NP-complete, too. Hence, we will show co-NP-hardness of INVERSE k-COLOURING by a reduction from INV-CSP($\{\neq_D, \{(0, 1, \ldots, k-1)\}\}$).

Let R be an n-ary relation over D. First, if there does not exist $i \in \{1, \ldots, n\}$ such that $|\mathrm{Pr}_i(R)| = 1$ then it is already the case that $R \in \langle \neq_D \rangle_{\not\exists}$ if and only if $R \in \langle \{\neq_D, \{(0, 1, \ldots, k-1)\}\} \rangle_{\not\exists}$. Second, if there exists $i \in \{1, \ldots, n\}$ such that $|\mathrm{Pr}_i(R)| = 1$ then $R \in \langle \{\neq_D, \{(0, 1, \ldots, k-1)\}\} \rangle_{\not\exists}$ only if there for every $d \in \{0, \ldots, k-1\}$ exists $j \in \{1, \ldots, n\}$ such that $\mathrm{Pr}_j(R) = \{(d)\}$. The reason why this holds is that the language $\{\neq_D, \{(0, 1, \ldots, k-1)\}\}$ can only qfpp-define a relation with constant arguments using the k-ary relation $\{(0, 1, \ldots, k-1)\}$, which enforces the remaining constant values.

Therefore, we may without loss of generality assume that

$$\mathrm{Pr}_{n-k+1,n-k+2,\ldots,n}(R) = \{(0, 1, \ldots, k-1)\}$$

and that no other argument in R is constant. Let $m = n - k + 1$. We then construct a relation R' by, for each permutation $\rho \colon D \to D$, adding the set $\{(\rho(x_1), \ldots, \rho(x_n)) \mid (x_1, \ldots, x_n) \in R\}$. This may be viewed as a finite-domain generalisation of the relation constructed in Lemma 10, obtained from closing the input relation under the permutation $\overline{x}$. Here, it is important to note that every unary $f \in \mathrm{Pol}(\neq_D)$ is in fact a permutation of this form. Assume that $R \in \langle \{\neq_D, \{(0, 1, \ldots, k-1)\}\} \rangle_{\not\exists}$ and let

$$\varphi(x_1, \ldots, x_n) \wedge \{(0, 1, \ldots, k-1)\}(x_{m,\ldots,x_n})$$

be a qfpp-definition where $\varphi(x_1, \ldots, x_n)$ consists only of $\neq_D$-constraints. Let

$$R_1(x_1, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n) \wedge \bigwedge_{i,j \in \{m,\ldots,n\}, i \neq j} (x_i \neq_D x_j).$$

We claim that $R_1 = R'$. Indeed, we trivially have that $R \subseteq R_1$, and if

$$(\tau(x_1), \ldots, \tau(x_n)) \in R'$$

for a permutation τ and tuple $(x_1, \ldots, x_n) \in R$, it also holds that

$$(\tau(x_1), \ldots, \tau(x_n)) \in R_1$$

since $R_1 \in \langle \neq_D \rangle_{\not\exists} \subseteq \langle \neq_D \rangle$ implies that R_1 is closed under all permutations. For the other direction, let $t \in R_1$, and let ρ be the permutation $\rho(0) = t[m], \ldots, \rho(k-1) = t[n]$. This is indeed a well-defined permutation due to the constraints

$$\bigwedge_{i,j \in \{m,\ldots,n\}, i \neq j} (x_i \neq_D x_j)$$

in the qfpp-definition of R_1. But then $\rho^{-1}(t) = s$ (applied componentwise) is included in R, implying that $\rho(s) = t \in R'$.

For the other direction, assume that $R' \in \langle \neq_D \rangle_{\not\exists}$ and let

$$R'(x_1, \ldots, x_n) \equiv \phi(x_1, \ldots, x_n)$$

be a qfpp-definition. Then we may qfpp-define R as

$$R(x_1, \ldots, x_n) \equiv \phi(x_1, \ldots, x_n) \wedge \{(0, 1, \ldots, k-1)\}(x_m, \ldots, x_n).$$

Hence, $R \in \langle \{\neq_D, \{(0, 1, \ldots, k-1)\}\} \rangle_{\not\exists}$ if and only if $R' \in \langle \neq_D \rangle_{\not\exists}$, implying that Inv-CSP($\neq_D$) is co-NP-complete. □

6 Concluding Remarks

We have studied the complexity of Inv-SAT(Γ) and obtained a complete dichotomy theorem for finite Γ. To prove this we first limited the number of cases we needed to consider with polymorphisms, and for each such case then used expressibility results based on partial polymorphisms, in order to proceed with the required reductions. We also showed that Inv-SAT(Γ) is a relevant problem for infinite constraint languages, even though the complexity landscape differs drastically from the finite case. Last, we demonstrated that the inverse CSP problem is an interesting topic in its own right, and used our general results to resolve the complexity status of inverse k-colouring, mentioned as an open problem in Chen [8]. Let us now consider a few different directions for future research.

A Dichotomy Theorem for Infinite Constraint Languages A good starting point for proving Conjecture 2 is to find examples of infinite Γ such that (1) there does not exist any finite $\Delta \subset \Gamma$ such that $\langle \Gamma \rangle_{\not\exists} = \langle \Delta \rangle_{\not\exists}$ and (2) pPol(Γ) is infinitely generated. One candidate for such a language is $\Gamma_{\text{XSAT}} = \{R_{1/k} \mid k \geq 3\}$, where both these properties can be proven to hold. Is Inv-SAT(Γ_{XSAT}) tractable or co-NP-complete?

Inverse Logical Reasoning Problems A wide range of problems parameterized by Boolean constraint languages have been considered in the literature (cf. the survey by Creignou et al. [9]). Natural inverse problems can be defined for many of these problems and seem especially interesting for logical reasoning problems such as abduction and circumscription. While dichotomy results akin to our dichotomy for Inv-SAT(Γ) might not be easy, the approach used in this article at least suggests the possibility. It would be particularly interesting to find examples where the inverse problem is tractable while the original problem is intractable. Theorem 17 already gives such examples for Inv-SAT(Γ), but it is not obvious that this argument is valid for different problems.

The Inverse Constraint Satisfaction Problem over Infinite Domains Another tempting problem is to study Inv-CSP(Γ) over infinite domains. In this case some extra care is needed since the instance R cannot always be represented explicitly as a list of tuples. However, there exist well-studied, so called *ω-categorical*, constraint languages where the Inv-CSP problem could be interesting, since there exist better methods to represent relations than listing its tuples. However, even the problem of checking if $R \in \langle \Gamma \rangle$ for Γ over infinite domains is in general undecidable [3], so there is little hope in obtaining a complete dichotomy.

Acknowledgements

We thank the anonymous reviewers for several helpful comments. The first author has received funding from the DFG-funded project "Homogene Strukturen, Bedingungserfüllungsprobleme, und topologische Klone" (Project number 622397), and from the Swedish Research Council (VR) under grant 2019-03690. The second author is partially supported by the *National Graduate School in Computer Science* (CUGS), Sweden.

References

[1] L. Barto. The constraint satisfaction problem and universal algebra. *Bulletin of Symbolic Logic*, 21(3):319–337, 2015.

[2] L. Barto, A. Krokhin, and R. Willard. Polymorphisms, and How to Use Them. In A. Krokhin and S. Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017.

[3] M. Bodirsky, M. Pinsker, and T. Tsankov. Decidability of definability. *Journal of Symbolic Logic*, 78(4):1036–1054, 2013.

[4] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. I. *Cybernetics*, 5:243–252, 1969.

[5] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. II. *Cybernetics*, 5:531–539, 1969.

[6] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM SIGACT-Newsletter*, 34(4):38–52, 2003.

[7] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part II: Constraint satisfaction problems. *ACM SIGACT-Newsletter*, 35(1):22–35, 2004.

[8] H. Chen. Inverse NP problems. *Computational Complexity*, 17(1):94–118, 2008.

[9] N. Creignou, P. Kolaitis, and B. Zanuttini. Structure identification of Boolean relations and plain bases for co-clones. *Journal of Computer and System Sciences*, 74(7):1103–1115, November 2008.

[10] N. Creignou and H. Vollmer. Boolean constraint satisfaction problems: When does Post's lattice help? In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 3–37. Springer Berlin Heidelberg, 2008.

[11] Creignou, N. and Hebrard, J.-J. On generating all solutions of generalized satisfiability problems. *RAIRO - Theoretical Informatics and Applications*, 31(6):499–511, 1997.

[12] D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27(1):95–100, 1968.

[13] P. Jonsson, V. Lagerkvist, and G. Nordh. Constructing NP-intermediate problems by blowing holes with parameters of various properties. *Theoretical Computer Science*, 581:67–82, 2015.

[14] P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Strong partial clones and the time complexity of SAT problems. *Journal of Computer and System Sciences*, 84:52 – 78, 2017.

[15] D. Kavvadias and M. Sideri. The inverse satisfiability problem. *SIAM Journal on Computing*, 28(1):152–163, 1998.

[16] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[17] V. Lagerkvist. Weak bases of Boolean co-clones. *Information Processing Letters*, 114(9):462–468, 2014.

[18] V. Lagerkvist and M. Wahlström. The power of primitive positive definitions with polynomially many variables. *Journal of Logic and Computation*, 27(5):1465–1488, 2017.

[19] V. Lagerkvist, M. Wahlström, and B. Zanuttini. Bounded bases of strong partial clones. In *Proceedings of the 45th International Symposium on Multiple-Valued Logic (ISMVL-2015)*, pages 189–194, 2015.

[20] D. Lau. *Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory (Springer Monographs in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[21] E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

[22] B.A. Romov. The algebras of partial functions and their invariants. *Cybernetics*, 17(2):157–167, 1981.

[23] I.G. Rosenberg. *Minimal clones I: the five types.* Colloquia mathematica Societatis János Bolyai. 1986.

[24] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory Of Computing (STOC-78)*, pages 216–226. ACM Press, 1978.

[25] H. Schnoor and I. Schnoor. Partial polymorphisms and constraint satisfaction problems. In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 229–254. Springer Berlin Heidelberg, 2008.

[26] R. Willard. Testing expressibility is hard. In *Proceedings of the 16th International Conference Principles and Practice of Constraint Programming (CP-2010)*, volume 6308 of *Lecture Notes in Computer Science*, pages 9–23. Springer, 2010.