A Dichotomy Theorem for the Inverse Satisfiability Problem

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

1 **Institut für Algebra, TU Dresden, Dresden, Germany**
victor.lagerqvist@tu-dresden.de
2 **Department of Computer and Information Science, Linköping University, Linköping, Sweden**
biman.roy@liu.se

Abstract

The *inverse satisfiability problem* over a set of Boolean relations Γ (Inv-SAT(Γ)) is the computational decision problem of, given a relation R, deciding whether there exists a SAT(Γ) instance with R as its set of models. This problem is co-NP-complete in general and a dichotomy theorem for finite Γ containing the constant Boolean relations was obtained by Kavvadias and Sideri. In this paper we remove the latter condition and prove that Inv-SAT(Γ) is always either tractable or co-NP-complete for all finite sets of relations Γ, thus solving a problem open since 1998. Very few of the techniques used by Kavvadias and Sideri are applicable and we have to turn to more recently developed algebraic approaches based on *partial polymorphisms*. We also consider the case when Γ is infinite, where the situation differs markedly from the case of SAT. More precisely, we show that there exists infinite Γ such that Inv-SAT(Γ) is tractable even though there exists finite $\Delta \subset \Gamma$ such that Inv-SAT(Δ) is co-NP-complete.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, G.2.0 Discrete Mathematics General

Keywords and phrases Clone Theory, Universal Algebra, Satisfiability Problems

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2017.39

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 either always tractable or is NP-complete [21]; a property that should not be taken for granted in light of the NP-intermediate problems constructed by Ladner [15]. In this paper 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 it sets of models. In fact, for every problem in NP there exists a corresponding inverse problem, and we refer the reader to Chen [7] for a survey on this topic. Contrary to SAT(Γ), Inv-SAT(Γ)

* The author has received funding from the DFG-funded project "Homogene Strukturen, Bedingungserfüllungsprobleme, und topologische Klone" (Project number 622397).
† The author is partially supported by the *National Graduate School in Computer Science* (CUGS), Sweden.

is in general co-NP-complete, and its computational complexity was studied by Kavvadias and Sideri [14]. 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. Such a property does not hold a priori for Inv-SAT(Γ), which increases the number of cases we need to consider. 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. [8] and to Barto [1]. 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, 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 [19]. Hence, instead of directly reasoning by properties of constraint languages, we can instead 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*. Unfortunately, partial polymorphisms are not nearly as well-studied as total polymorphisms, which makes such classifications more problematic. To tackle this issue we use the algebraic techniques developed by Schnoor and Schnoor [23] and Lagerkvist [16] and are able 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 [10].

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). Somewhat 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. Hence, for infinite languages, the complexity of Inv-SAT(Γ) is markedly different from the complexity of enumeration, even though the complexity coincides for finite languages. Moreover, we provide an algebraic criterion for this phenomena based on the expressive power of the partial polymorphisms of Γ, and conjecture that this property is both necessary and sufficient.

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 $\Pr_{i_1,\ldots,i_{n'}}(t) = (t[i_1],\ldots,t[i_{n'}])$ 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 $\Pr_{i_1,\ldots,i_{n'}}(R) = \{\Pr_{i_1,\ldots,i_{n'}}(t) \mid t \in R\}$. 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) $\Gamma \subseteq BR$ 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 $f : V \to \{0,1\}$ such that $(f(x_1),\ldots,x_{\mathrm{ar}(R)}) \in R$ for every $R(x_1,\ldots,x_{\mathrm{ar}(R)}) \in C$?

▶ **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.

We will sometimes view a SAT(Γ) instance as a conjunctive formula φ and write $Sols(\varphi)$ to denote its set of models. 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 formally, we define INV-SAT(Γ) 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. In general the INV-SAT(Γ) problem is co-NP-complete and a dichotomy theorem is known for finite and ultraidempotent constraint languages Γ [14].

▶ **Theorem 2.** *Let Γ be a finite and ultraidempotent constraint language. Then* INV-SAT(Γ) *is either co-NP-complete or tractable.*

▶ **Example 3.** 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 2 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 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 $R(x_1, \ldots, x_n) \equiv \exists y_1, \ldots, y_{n'}.R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$, 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 now 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 $\langle\Gamma\rangle_{\not\exists}$ denote the smallest set of relations containing Γ and which is closed under taking qfpp-definitions. 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 $\langle\Gamma\rangle_{\not\exists}$. 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., $R \in \langle\Gamma\rangle_{\not\exists}$. Whenever convenient we will therefore assume that Inv-SAT(Γ) is the problem of checking whether $R \in \langle\Gamma\rangle_{\not\exists}$. 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 [24].

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 Pol(R) for the set of polymorphisms of the relation R and if Γ is a constraint language we let $\mathrm{Pol}(\Gamma) = \bigcap_{\mathrm{R}\in\Gamma} \mathrm{Pol(R)}$. Sets of the form 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 $f \star g_1, \ldots, g_m(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ is included in Pol(Γ)). We let $[F]$ be the smallest clone containing the set F. There is a powerful connection between clones and co-clones which we will now describe. 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, $\mathrm{Pol}(\mathrm{Inv(F)}) = \mathrm{[F]}$. We now have the following *Galois connection* between Inv($\cdot$) and Pol($\cdot$).

▶ **Theorem 4** ([3, 4, 11]). *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 Galois 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 $\mathrm{domain}(f)$. We now say

that a k-ary partial operation f is 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 $(t_1[i], \ldots, t_k[i]) \notin \text{domain}(f)$, for every sequence $t_1, \ldots, t_k \in R$. We write pPol(R) for the set of all partial polymorphisms of R and pPol(Γ) for the set $\bigcap_{R \in \Gamma} \text{pPol}(\Gamma)$. 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 $f \in \text{pPol}(\Gamma)$ then $g \in \text{pPol}(\Gamma)$ for every g such that $\text{domain}(g) \subseteq \text{domain}(f)$ and g matches the values of f for these arguments. 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 $[F]_s$. The reason why we define these technical concepts will be explained in Section 4 where we study the complexity of Inv-SAT(Γ) when pPol(Γ) 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, $\langle\Gamma\rangle_{\not\exists} = \text{Inv}(\text{pPol}(\Gamma))$ and $[F]_s = \text{pPol}(\text{Inv}(\text{F}))$. We then have the following Galois connection between Inv(·) and pPol(·), due to Geiger [11] and Romov [20].

▶ **Theorem 5** ([11, 20]). *Let* Γ *and* Γ′ *be two constraint languages. Then* $\Gamma \subseteq \langle\Gamma'\rangle_{\not\exists}$ *if and only if* $\text{pPol}(\Gamma') \subseteq \text{pPol}(\Gamma)$.

Using the results in this section we can now present the dichotomy theorem from Kavvadias and Sideri [14] more precisely as follows.

▶ **Theorem 6.** *Let* Γ *be a finite and ultraidempotent constraint language. Then* Inv-SAT(Γ) *is co-NP-complete if* $\text{Pol}(\Gamma) = \Pi_{\mathbb{B}}$ *and is tractable otherwise.*

We remark that the tractable cases in Theorem 6 stem from the observation that if one can enumerate the solutions of SAT(Γ) with polynomial delay, then Inv-SAT(Γ) must be tractable. To see this, let R be an instance of Inv-SAT(Γ), and begin by computing a qfpp-definition φ over Γ with the property that $Sols(\varphi) \supseteq R$, according to the strategy in Kavvadias and Sideri [14]. 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 6 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 6 remain valid even if Γ is not ultraidempotent since the enumeration algorithms works equivalently well in these cases. To better describe the remaining cases we will need to define the following Boolean operations.

▶ **Definition 7.** We define the following Boolean operations.

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

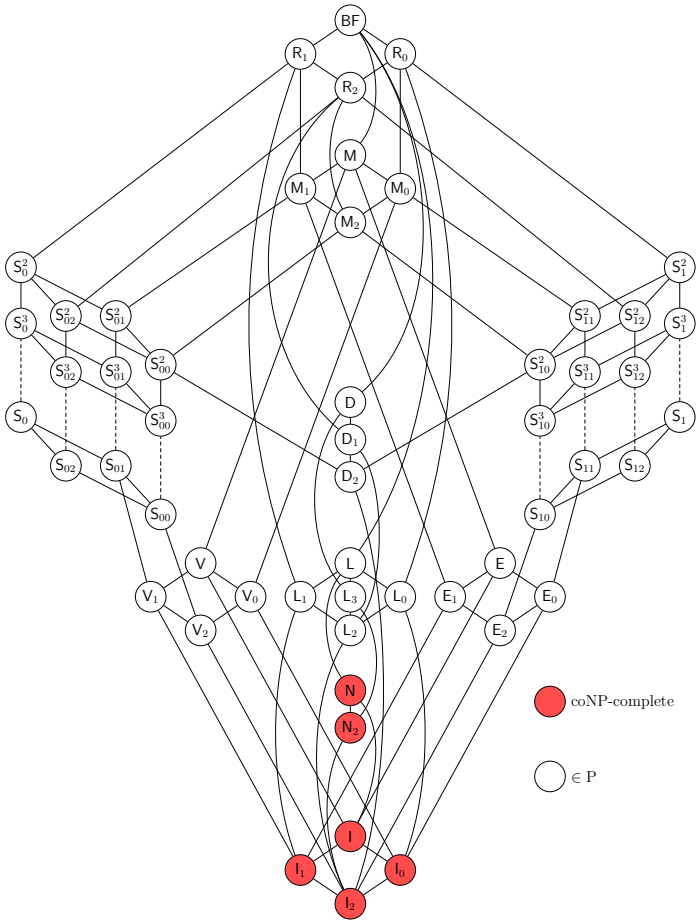

Figure 1 The complexity of Inv-SAT(Γ) for finite Γ.

Then, using the terminology from Böhler et al. [5, 6], $[\{f_0, f_1, \overline{x}\}] = \mathsf{N}$, $[\{f_0, f_1\}] = \mathsf{I}$, $[\{f_0\}] = \mathsf{I}_0$, $[\{f_1\}] = \mathsf{I}_1$, $[\{\overline{x}\}] = \mathsf{N}_2$, 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. The intuition behind the theorem is that one cannot enumerate the solutions of SAT(Γ) with polynomial delay if Pol(Γ) ⊆ [F] for $F \subseteq \{f_0, f_1, \overline{x}\}$, unless P = NP [10].

▶ **Theorem 8.** *Let* Γ *be a finite constraint language. Then* Inv-SAT(Γ) *is co-NP-complete if* Pol(Γ) ⊆ [F] *for* $F \subseteq \{f_0, f_1, \overline{x}\}$ *and is tractable otherwise.*

The theorem will be proved in Lemma 11, Lemma 12, Lemma 14, and Lemma 16. At this stage 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 Pol(Δ) ⊆ Pol(Γ). Then, what one needs to do is simply to take every clone Pol(Γ) in Post's lattice and determine the complexity of $X(\Gamma)$, since the results then automatically carry over to every $X(\Delta)$ such that Pol(Δ) = Pol(Γ). This is e.g. the case for SAT and many Boolean optimization and logical reasoning problems [8]. For the Inv-SAT(Γ) problem we do not have such a result, implying that the proof strategy is more complex. However, we will see that it is possible to overcome this using properties of weak systems. For this we will need the following lemma.

▶ **Lemma 9.** *Let* $\mathrm{Pol}(\Gamma) \subseteq [\mathrm{F}]$ *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* $\mathrm{Pol}(\Gamma) = [\{\bar{\mathrm{x}}\}]$,
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* $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0, \mathrm{f}_1, \bar{\mathrm{x}}\}]$,
4. $\tau_{\to} = \{(0,0),(1,0),(1,1)\} \in \langle\Gamma\rangle_{\nexists}$ *if* $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0, \mathrm{f}_1\}]$,
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* $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0\}]$, *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* $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_1\}]$.

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 [23] and Lagerkvist [16]. We first remark that for $\mathrm{Pol}(\Gamma) \in \{[\{\mathrm{f}_0\}], [\{\mathrm{f}_1\}], [\{\mathrm{f}_0, \mathrm{f}_1, \bar{\mathrm{x}}\}]\}$ the relations follow immediately from Theorem 11 in Lagerkvist [16]. Hence, the remaining cases are when $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$, $\mathrm{Pol}(\Gamma) = [\bar{\mathrm{x}}]$, and $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0, \mathrm{f}_1\}]$. First assume that $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$. From Lagerkvist [16] 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 $\mathrm{Pol}(\Gamma) = [\{\bar{\mathrm{x}}\}]$. 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 [13, 16]. 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 $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0, \mathrm{f}_1\}]$, the relation $R = \{(0,0,0,0),(0,0,1,1),(0,1,0,1),(1,1,1,1)\} \in \langle\Gamma\rangle_{\nexists}$ [16], and this relation can qfpp-define $\tau_{\to}$ by $\tau_{\to}(x_1,x_2) \equiv R(x_1,x_1,x_2,x_2)$. ◀

The usefulness of this lemma is that we now have a better understanding of the expressiveness of the languages under consideration. For example, if $\mathrm{Pol}(\Gamma) = [\{\bar{\mathrm{x}}\}]$ then we know that Γ is expressive enough to qfpp-define the binary inequality relation $\tau_{\neq}$. Before we begin to prove Theorem 8 we present a lemma that will simplify some of the forthcoming reductions. 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 10.** *Let* Γ *be a constraint language and* R *an* n*-ary relation. Then* $R \in \langle\Gamma\rangle_{\nexists}$ *if and only if* $R^{\mathrm{irr}} \in \langle\Gamma\rangle_{\nexists}$.

In some of the forthcoming reductions we will 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 which is not a partial projection [18]. 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 5 implies that $R \notin \langle\Gamma\rangle_{\nexists}$. We are now ready to prove our first result, and begin with the case when $\mathrm{Pol}(\Gamma)$ consists only of projections (which due to Theorem 4 implies that Γ can pp-define every Boolean relation).

▶ **Lemma 11.** *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = \Pi_{\mathbb{B}}$. *Then* Inv-SAT($\Gamma$) *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 6 that Inv-SAT(Δ) is co-NP-complete. Hence, let R be an n-ary relation, i.e., an instance of Inv-SAT(Δ). We now observe that 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 $R \in \langle\Delta\rangle_{\not\exists}$ if and only if $R \in \langle\{(0)\}\rangle_{\not\exists}$. 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 $R \in \langle\Delta\rangle_{\not\exists}$ if and only if $R \in \langle\{(1)\}\rangle_{\not\exists}$. In both these cases we can compute the answer in polynomial time and output an arbitrary yes- or no-instance to Inv-SAT(Γ).

Hence, assume that there exist both i and j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$, and for simplicity assume that R does not contain any redundant arguments, which is possible by Lemma 10. We then claim that $R \in \langle\Delta\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma\rangle_{\not\exists}$. Hence, first assume that $R \in \langle\Delta\rangle_{\not\exists}$, 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 $\tau \times \{(0,1)\} \in \Delta$, 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 $\{(0,1)\} \in \langle\Gamma\rangle_{\not\exists}$ by Lemma 9. For the other direction, assume that $R \in \langle\Gamma\rangle_{\not\exists}$ 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 $\tau_k \times \{(0,1)\}(\mathbf{x}_k, x_i, x_j)$. ◀

▶ **Lemma 12.** *Let* Γ *be a finite constraint language such that* $\mathrm{Pol}(\Gamma) = [\{\bar{\mathrm{x}}\}]$. *Then* Inv-SAT($\Gamma$) *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 11, 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 $R \in \langle\Gamma\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$, 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 $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ or $R \in \langle\Gamma\rangle_{\not\exists}$, 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 10 we may without loss of generality assume that R does not contain any other constant arguments.

In this case we claim that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ if and only if $\neg(R) \in \langle\Gamma\rangle_{\not\exists}$, where $\neg(R) = R \cup \{\bar{t} \mid t \in R\}$, i.e., R closed under complement. Assume first that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ 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 $\neg(R)$ 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 9. 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 $R' = \neg(R)$. It is easy to see that $\neg(R) \subseteq R'$. Hence, let $t \in R'$,

assume that $t \notin \neg(R)$, and observe that this also implies that $\bar{t} \notin \neg(R)$, since $\neg(R)$ is closed under complement. Due to the construction of R' this is clearly not possible. For the other direction, assume that $\neg(R) \in \langle\Gamma\rangle_{\not\exists}$ and let $\neg(R)(x_1, \ldots, x_n) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$ 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 $t \in \neg(R)$ satisfying $t[i] = 0$ and $t[j] = 1$. ◀

For the case when Γ is preserved by a constant operation we will first need to show co-NP-completeness of the following auxiliary problem. Say that an n-ary Boolean relation R is *complementary saturated* if there for every $1 \leq i \leq n$ exists $1 \leq j \leq n$ such that $t[i] \neq t[j]$ for every $t \in R$. In other words for each argument of the relation there exists an argument which is its complement. For a finite constraint language Γ by Inv-SAT$^{\neq}(\Gamma)$ now we denote the structurally restricted Inv-SAT(Γ) problem defined as follows.

Instance: A complementary saturated Boolean relation R. Question: $R \in \langle\Gamma\rangle_{\not\exists}$?

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

▶ **Lemma 13.** *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 $R^{\neq} \in \Gamma^{\neq}$ is ordered in such a way that $\mathrm{Pr}_{1,\ldots,\mathrm{ar}(R)}(R^{\neq}) = R$, and that the remaining arguments are the complement of the arguments of R. Observe that $\mathrm{Pol}(\Gamma^{\neq}) = \Pi_{\mathbb{B}}$, which by Lemma 11 implies that Inv-SAT$(\Gamma^{\neq})$ is co-NP-complete.

Hence, let R be an n-ary relation. Assume that R is not complementary saturated, i.e., not a valid instance of Inv-SAT$^{\neq}(\Gamma)$. Then either $R \in \langle\{(0,0),(1,1)\}\rangle_{\not\exists} \subseteq \langle\Gamma^{\neq}\rangle_{\not\exists}$, in which case we output an arbitrary yes-instance, or $R \notin \langle\Gamma^{\neq}\rangle_{\not\exists}$, in which case we output an arbitrary no-instance. Otherwise R is already a valid instance of Inv-SAT$^{\neq}(\Gamma)$, and in this case we claim that $R \in \langle\Gamma^{\neq}\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma\rangle_{\not\exists}$. First assume that $R \in \langle\Gamma^{\neq}\rangle_{\not\exists}$. Via Lemma 9 we know that $\tau^{01}_{\neq} \in \langle\Gamma\rangle_{\not\exists}$, and from this property it follows that $\Gamma^{\neq} \subseteq \langle\Gamma\rangle_{\not\exists}$, implying that $R \in \langle\Gamma\rangle_{\not\exists}$. For the other direction, assume that $R \in \langle\Gamma\rangle_{\not\exists}$. 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 $R(x_1, \ldots, x_n) \equiv R^{\neq}_1(\mathbf{x}_1, \mathbf{y}_1) \wedge \ldots \wedge R^{\neq}_m(\mathbf{x}_m, \mathbf{y}_m)$ is a valid qfpp-definition of R over $\Gamma^{\neq}$. ◀

▶ **Lemma 14.** *Let Γ be a finite constraint language such that $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0\}]$ or $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_1\}]$. Then* Inv-SAT$(\Gamma)$ *is co-NP-complete.*

Proof. We present the proof for the case when $\mathrm{Pol}(\Gamma = [\{\mathrm{f}_1\}]$ since the other case is entirely analogous. In order to prove this we will give a polynomial-time reduction from Inv-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$ to Inv-SAT(Γ). The problem Inv-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$ is co-NP-complete by Lemma 13 since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$.

Let R be an instance Inv-SAT$^{\neq}(\Gamma \cup \{(0,1)\})$ of arity n. 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 $R \in \langle\Gamma\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$; therefore we assume that such i and j exist. First

construct the relation $R' = R \cup \{(0,\dots,0)\}$, i.e., the relation R adjoined with the constant 0 tuple. We will now prove that $R' \in \langle\Gamma\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$. Hence, first assume that $R' \in \langle\Gamma\rangle_{\not\exists}$ and let $R'(x_1,\dots,x_n) \equiv R_1(\mathbf{x}_1) \wedge \dots \wedge R_m(\mathbf{x}_m)$ be a qfpp-definition over Γ. Then consider the qfpp-definition $R(x_1,\dots,x_n) \equiv R_1(\mathbf{x}_1) \wedge \dots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i,x_j)$. The intuition behind this qfpp-definition is that the additional constraint $\{(0,1)\}(x_i,x_j)$ will ensure that the constant 0 tuple included in R' but not in R. For the other direction assume that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ and let $R(x_1,\dots,x_n) \equiv R_1(\mathbf{x}_1) \wedge \dots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i,x_j)$ denote a qfpp-definition, where we without loss of generality assume that $R_1,\dots,R_m \in \Gamma$. 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 9, 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 occuring in positions $k+1,\dots,2k$ are the complement of the k first, and that the last two arguments are constant 0 and constant 1, respectively. We can thus qfpp-define R' as

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

To see that this definition is indeed correct, note that if x_i and x_{i+k} are both assigned the value 0, then this also forces the variable x_{2k+2} the value 0. But this implies that every other variable must be assigned 0 as well, yielding the constant 0 tuple which is included in R'. ◀

▶ **Lemma 15.** *Let Γ be a finite constraint language such that $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0,\mathrm{f}_1\}]$. 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($\Gamma \cup \{(0,1)\}$), which is co-NP-complete since $\mathrm{Pol}(\Gamma \cup \{(0,1)\}) = \Pi_{\mathbb{B}}$. Hence, let R be an n-ary relation. If there does not exist $i,j \in \{1,\dots,n\}$ such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$ then it is already the case that $R \in \langle\Gamma\rangle_{\not\exists}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$. Therefore, assume that such i and j exist, and construct the relation $R' = R \cup \{(0,\dots,0),(1,\dots,1)\}$. We claim that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ if and only if $R' \in \langle\Gamma\rangle_{\not\exists}$. For the first direction, assume that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ and let $R(x_1,\dots,x_n) \equiv R_1(\mathbf{x}_1) \wedge \dots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_i,x_j)$ denote a qfpp-definition such that $R_1,\dots,R_m \in \Gamma$. Recall that $\tau_{\to} = \{(0,0),(0,1),(1,1)\}$ from Lemma 9 is qfpp-definable by Γ. Now construct the qfpp-definition

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

To see that this definition is correct, observe that the additional constraints of the form $(\tau_{\to}(x_i,x_k) \wedge \tau_{\to}(x_k,x_j)$ 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 $(0,\dots,0)$ and $(1,\dots,1)$. The other direction ($R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\exists}$ if $R' \in \langle\Gamma\rangle_{\not\exists}$) can be proven using similar arguments as in the proof of Lemma 14. ◀

▶ **Lemma 16.** *Let Γ be a finite constraint language such that $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0,\mathrm{f}_1,\bar{\mathrm{x}}\}]$. Then* Inv-SAT($\Gamma$) *is co-NP-complete.*

Proof. As $\mathrm{Pol}(\Gamma) = [\{\mathrm{f}_0, \mathrm{f}_1, \bar{\mathrm{x}}\}]$ 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$^{\neq}(\Gamma \cup \{(0,1)\})$. First we check whether there exists i and j such that $\mathrm{Pr}_i(R) = \{(0)\}$ and $\mathrm{Pr}_j(R) = \{(1)\}$. If this is not the case then $R \in \langle\Gamma\rangle_{\not\equiv}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\equiv}$, and we are done. Therefore assume that such i and j exists. 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 $R' \in \langle\Gamma\rangle_{\not\equiv}$ if and only if $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\equiv}$. Therefore, first assume that $R \in \langle\Gamma \cup \{(0,1)\}\rangle_{\not\equiv}$ and let $R(x_1,\ldots,x_{2k+2}) \equiv R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m) \wedge \{(0,1)\}(x_{2k-1}, x_{2k})$ be a qfpp-definition over Γ where $R_1, \ldots, R_m \in \Gamma$. Now consider the qfpp-definition

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

where $\tau_{f_0,f_1,\bar{x}} = \{(0,0,0,0),(1,1,1,1),(0,1,0,1),(1,0,0,1),(1,0,1,0),(0,1,1,0)\} \in \langle\Gamma\rangle_{\not\equiv}$ is the relation from Lemma 9. 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 $\bar{x}$. 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 $\bigwedge_{l \in \{1,\ldots,n\}} \tau_{f_0,f_1,\bar{x}}(x_l, x_{l+k}, x_{2k+1}, x_{2k+2})$, 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 $R(x_1,\ldots x_{2k+2}) \equiv R'(x_1,\ldots x_{2k+2}) \wedge \{(0,1)\}(x_{2k+1}, x_{2k+2})$. This concludes the reduction. ◀

By combining Lemma 11– 12 and Lemma 14– 16 we have thus finally proven Theorem 8.

▶ **Example 17.** We can now answer the question regarding the complexity of Inv-SAT$(\{R_{1/3}\})$ from Example 3. It is not hard to verify that $R_{1/3}$ is only preserved by the projections, from which it follows that $\mathrm{Pol}(\mathrm{R}_{1/3}) = \Pi_{\mathbb{B}}$. An application of Theorem 8 then reveals that Inv-SAT$(\{R_{1/3}\})$ is indeed co-NP-complete.

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 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 8 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.

▶ **Theorem 18.** *Let* Γ *be a constraint language such that* pPol(Γ) *admits a finite base* F*. Then* Inv-SAT(Γ) *is solvable in polynomial time.*

Proof. Let R be an instance of Inv-SAT(Γ) of arity n. Due to the Galois connection in Theorem 5 the question $R \in \langle\Gamma\rangle_{\nexists}$ is equivalent to checking whether $F \subseteq \mathrm{pPol}(\{\mathrm{R}\})$, 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. ◀

It is worth remarking that Γ is *always* infinite when pPol(Γ) is finitely generated and $\mathrm{Pol}(\Gamma) \supseteq [\{\mathrm{f}_0, \mathrm{f}_1, \overline{\mathrm{x}}]$ [17] — hence there is no possible overlap between Theorem 8 and Theorem 18. This result may be seen as surprising since computational problems parameterized by Boolean constraint languages tend to be rather well-behaved, and 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 [12]. 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 8. 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 18, 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 pPol(Γ) is sufficiently simple. We thus make the following conjecture.

▶ **Conjecture 19.** *Let* Γ *be a Boolean constraint language such that* $\mathrm{Pol}(\Gamma) \supseteq [\{\mathrm{f}_0, \mathrm{f}_1, \overline{\mathrm{x}}\}]$*. Then* Inv-SAT(Γ) *is tractable if* pPol(Γ) *is finitely generated and is co-NP-hard otherwise.*

5 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 demonstrated that Inv-SAT(Γ) is also a relevant problem for infinite constraint languages, even though the situation differs drastically from the finite case. These results raise a few different directions for future research.

A Dichotomy Theorem for Infinite Constraint Languages

A good starting point for proving Conjecture 19 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 $\Gamma_{\text{XSAT}} = \{R_{1/k} \mid k \geq 3\}$, $R_{1/k} = \{(b_1, \ldots, b_k) \in \{0,1\}^k \mid b_1 + \ldots + b_k = 1\}$, where both these properties can be proven to hold. Is Inv-SAT(Γ_{XSAT}) tractable or co-NP-complete?

The Inverse Constraint Satisfaction Problem

The *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)) is a multi-valued generalization of SAT where Γ may contain non-Boolean relations. One may then define Inv-CSP(Γ) analogously to Inv-SAT and ask if a dichotomy theorem can be obtained for finite Γ. This is likely a good deal harder than the Boolean case and a starting point would be to compare the complexity of Inv-CSP(Γ) to the complexity of enumerating solutions of Inv-CSP(Γ) with polynomial delay [22]. In particular it would be interesting to find examples of Γ such that Inv-CSP(Γ) is tractable even though the enumeration problem is not tractable.

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 [2], so there is little hope in obtaining a complete dichotomy.

Acknowledgements. We thank the anonymous reviewers for several helpful comments.

References

1. L. Barto. Constraint satisfaction problem and universal algebra. *ACM SIGLOG News*, 1(2):14–24, 2014.
2. M. Bodirsky, M. Pinsker, and T. Tsankov. Decidability of definability. *Journal of Symbolic Logic*, 78(4):1036–1054, 2013.
3. V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. I. *Cybernetics*, 5:243–252, 1969.
4. V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. II. *Cybernetics*, 5:531–539, 1969.
5. 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.
6. 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.
7. H. Chen. Inverse NP problems. *Computational Complexity*, 17(1):94–118, 2008.
8. 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.
9. Nadia Creignou, Phokion G. Kolaitis, and Bruno Zanuttini. Structure identification of boolean relations and plain bases for co-clones. *J. Comput. Syst. Sci.*, 74(7):1103–1115, 2008. doi:10.1016/j.jcss.2008.02.005.

10 Creignou, N. and Hebrard, J.-J. On generating all solutions of generalized satisfiability problems. *RAIRO-Theor. Inf. Appl.*, 31(6):499–511, 1997.

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

12 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.

13 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.

14 D. Kavvadias and M. Sideri. The inverse satisfiability problem. *SIAM Journal on Computing*, 28:152–163, 1998.

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

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

17 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.

18 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.

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

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

21 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.

22 H. Schnoor and I. Schnoor. Enumerating all solutions for constraint satisfaction problems. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS-2007)*, volume 4393, pages 694–705. Springer, 2007.

23 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.

24 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.