On the Strength of Uniqueness Quantification in Primitive Positive Formulas

Victor Lagerkvist*[1] and Gustav Nordh†[2]

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

Abstract

Uniqueness quantification ($\exists!$) is a quantifier in first-order logic where one requires that exactly one element exists satisfying a given property. In this paper we investigate the strength of uniqueness quantification when it is used in place of existential quantification in conjunctive formulas over a given set of relations Γ, so-called *primitive positive definitions* (pp-definitions). We fully classify the Boolean sets of relations where uniqueness quantification has the same strength as existential quantification in pp-definitions and give several results valid for arbitrary finite domains. We also consider applications of $\exists!$-quantified pp-definitions in computer science, which can be used to study the computational complexity of problems where the number of solutions is important. Using our classification we give a new and simplified proof of the trichotomy theorem for the unique satisfiability problem, and prove a general result for the unique constraint satisfaction problem. Studying these problems in a more rigorous framework also turns out to be advantageous in the context of lower bounds, and we relate the complexity of these problems to the *exponential-time hypothesis*.

1 Introduction

A *primitive positive definition* (pp-definition) over a relational structure $\mathcal{A} = (A; R_1, \ldots, R_k)$ is a first-order formula $\exists y_1, \ldots, y_m \colon \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ with free variables $x_1, \ldots, x_n$ where $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ is a conjunctive formula. Primitive positive definitions have been extremely influential in the last decades due to their one-to-one correspondence with term algebras in universal algebra, making them a cornerstone in the *algebraic approach* for studying computational complexity [1, 10]. In short, pp-definitions can be used to obtain classical "gadget reductions" between problems by replacing constraints by their pp-definitions, which in the process might introduce fresh variables viewed as being existentially quantified. This approach has successfully been used to study the complexity of e.g. the *constraint satisfaction problem* (CSP) which recently led to a dichotomy between tractable and NP-complete CSPs [6, 31]. However, these reductions are typically not sufficient for optimisation problems and other variants of satisfiability, where one needs reductions preserving the number of models, so-called *parsimonious* reductions. Despite the tremendous advances in the algebraic approach there is currently a lack of methods for studying problems requiring parsimonious reductions, and in this paper we take the first step in developing such a framework. The requirement of parsimonious reductions

*victor.lagerkvist@liu.se
†gustav.nordh@gmail.com

can be realised by restricting existential quantification to *unique quantification* ($\exists!$), where we explicitly require that the variable in question can be expressed as a unique combination of other variables. That is, $\mathcal{A} \models \exists! x_i \colon \varphi(x_1, \ldots, x_i, \ldots, x_n)$ if and only if there exists a function f such that $f(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) = a_i$, for all $a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n \in A$ where $\mathcal{A} \models \varphi(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n)$. This notion of unique quantification is not the only one possible and we discuss an alternative viewpoint in Section 5. As a first step in understanding the applicability of uniqueness quantification in complexity classifications we are interested in studying the expressive power of unique existential quantification when used in place of existential quantification in pp-definitions, which we call *upp-definitions*. Any variables introduced by the resulting gadget reductions are then uniquely determined and do not affect the number of models.

Our main question is then: for which relational structures $\mathcal{A}$ is it the case that for every pp-formula $\varphi(x_1, \ldots, x_n)$ there exists a upp-formula $\vartheta(x_1, \ldots, x_n)$ such that $\mathcal{A} \models \varphi(a_1, \ldots, a_n) \Leftrightarrow \mathcal{A} \models \vartheta(a_1, \ldots, a_n)$ for all $a_1, \ldots, a_n \in A$? If this holds over $\mathcal{A}$ then uniqueness quantification has the same expressive power as existential quantification. The practical motivation for studying this is that if upp-definitions are as powerful as pp-definitions in $\mathcal{A}$, then, intuitively, any gadget reduction between two problems can be replaced with a parsimonious reduction. Given the generality of this question a complete answer for arbitrary relational structures is well out of reach, and we begin by introducing simplifying concepts. First, pp-definitions can be viewed as a closure operator over relations, and the resulting closed sets of relations are known as *relational clones*, or *co-clones* [23]. For each universe A the set of co-clones over A then forms a lattice when ordered by set inclusion, and given a set of relations Γ we write $\langle \Gamma \rangle$ for the smallest co-clone over A containing Γ. Similarly, closure under upp-definitions can also be viewed as a closure operator, and we write $\langle \Gamma \rangle_{\exists!}$ for the smallest set of relations over A containing Γ and which is closed under upp-definitions. Using these notions the question of the expressive strength of upp-definitions can be stated as: for which sets of relations Γ is it the case that $\langle \Gamma \rangle = \langle \Gamma \rangle_{\exists!}$? The main advantage behind this viewpoint is that a co-clone $\langle \Gamma \rangle$ can be described as the set of relations *invariant* under a set of operations F, $\mathrm{Inv}(F)$, such that the operations in F describe all permissible combinations of tuples in relations from Γ. An operation $f \in F$ is also said to be a *polymorphism* of Γ and if we let $\mathrm{Pol}(\Gamma)$ be the set of polymorphisms of Γ then $\mathrm{Pol}(\Gamma)$ is called a *clone*. This relationship allows us to characterise the cases that need to be considered by using known properties of $\mathrm{Pol}(\Gamma)$, which is sometimes simpler than working only on the relational side. This strategy will prove to be particularly useful for Boolean sets of relations since all Boolean clones and co-clones have been determined [26].

Our Results Our main research question is to identify Γ such that $\langle \Delta \rangle_{\exists!} = \langle \Gamma \rangle$ for each Δ such that $\langle \Delta \rangle = \langle \Gamma \rangle$. If this holds we say that $\langle \Gamma \rangle$ is $\exists!$-*covered*. The main difficulty for proving this is that it might not be possible to directly transform a pp-definition into an equivalent upp-definition. To mitigate this we analyse relations in co-clones using *partial polymorphisms*, which allows us to analyse their expressibility in a very nuanced way. In Section 3.1 we show how partial polymorphisms can be leveraged to prove that a given co-clone is $\exists!$-covered. Most notably, we prove that $\langle \Gamma \rangle$ is $\exists!$-covered if $\mathrm{Pol}(\Gamma)$ consists only of projections of the form $\pi(x_1, \ldots, x_i, \ldots, x_n) = x_i$, or of projections and constant operations. As a consequence, Γ pp-defines all relations over A if and only if Γ upp-defines all relations over A. One way of interpreting this result is that if Γ is "sufficiently expressive" then pp-definitions can always be turned into upp-definitions. However, there also exists $\exists!$-covered co-clones where the reason is rather that Γ is "sufficiently weak". For example, if Γ is invariant under the affine operation $x - y + z \,(\mathrm{mod}\, |A|)$, then existential quantification does not add any expressive power over unique existential quantification, since any existentially quantified variable occurring in a pp-definition can be expressed via a linear equation, and is therefore uniquely determined by other arguments. In Section 3.2 we then turn to the Boolean domain, and obtain a full classification of the $\exists!$-covered co-clones. Based on the results in Section 3.1 it is reasonable to expect that the covering property holds for sufficiently expressive languages and sufficiently weak languages, but that there may exist cases in between where unique quantification differs from existential quantification. This is indeed true, and we prove that the Boolean co-clones corresponding to non-positive Horn clauses, implicative and positive clauses, and their dual cases, are not $\exists!$-covered. Last, in Section 4 we demonstrate how the results from Section 3 can be used for obtaining complexity classifications of computational problems. One example of a problem requiring parsimonious reductions is the *unique satisfiability problem* over a Boolean set of relations Γ (U-SAT(Γ)) and its multi-valued generalization the *unique constraint satisfaction problem* (U-CSP(Γ)), where the goal is to determine if there exists a unique model of a given conjunctive Γ-formula. The complexity of U-SAT(Γ) was settled by Juban [18] for finite sets of relations Γ, essentially using a large case analysis. Using the results from Section 3.2 this complexity classification can instead be proved in a succinct manner, and we are also able to extend the classification to infinite Γ and large classes of non-Boolean Γ. This systematic approach is also advantageous for proving lower bounds, and we relate the complexity of U-SAT(Γ) to the highly influential *exponential-time hypothesis* (ETH) [13], by showing that none of the intractable cases of U-SAT(Γ) admit subexponential algorithms without violating the ETH.

Related Work Primitive positive definitions with uniqueness quantification appeared in Creignou & Hermann [7] in the context of "quasi-equivalent" logical formulas, and in the textbook by Creignou et al. [8] under the name of *faithful implementations*. Similarly, upp-definitions were utilised by Kavvadias & Sideri [19] to study the complexity of the *inverse satisfiability problem*. A related topic is *frozen quantification*, which can be viewed as uniqueness quantification restricted to variables that are constant in any model [24].

2 Preliminaries

2.1 Operations and Relations

In the sequel, let $D \subseteq \mathbb{N}$ be a finite domain of values. A k-ary function $f \colon D^k \to D$ is sometimes referred to as an *operation* over D and we write $\mathrm{ar}(f)$ to denote the arity k. Similarly, a *partial operation* over D is a map $f \colon \mathrm{dom}(f) \to D$ where $\mathrm{dom}(f) \subseteq D^k$ is called the *domain* of f, and we let $\mathrm{ar}(f) = k$ be the arity of f. If f and g are k-ary partial operations such that $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ and $f(t) = g(t)$ for each $t \in \mathrm{dom}(f)$ then f is said to be a *suboperation* of g. For $k \geq 1$ and $1 \leq i \leq k$ we let π^k_i be the ith *projection*, i.e., $\pi^k_i(x_1, \ldots, x_i, \ldots, x_k) = x_i$ for all $x_1, \ldots, x_i, \ldots, x_k \in D$. We write OP_D for the set of all operations over D and pOP_D for the set of all partial operations over D. As a notational shorthand we for $k \geq 1$ write $[k]$ for the set $\{1, \ldots, k\}$. For $d \in D$ we by $\mathbf{d}^n$ denote the constant n-ary tuple $(d, \ldots, d)$. Say that a k-ary $f \in \mathsf{OP}_D$ is *essentially unary* if there exists unary $g \in \mathsf{OP}_D$ and i such that $f(x_1, \ldots, x_i, \ldots, x_n) = g(x_i)$ for all $x_1, \ldots, x_i, \ldots, x_n \in D$.

Given an n-ary relation $R \subseteq D^n$ we write $\mathrm{ar}(R)$ to denote its arity n. If $t = (x_1, \ldots, x_n)$ is an n-ary tuple we write $t[i]$ to denote the ith element x_i, and $\mathrm{Proj}_{i_1, \ldots, i_{n'}}(t) = (t[i_1], \ldots, t[i_{n'}])$ to denote the *projection* on the coordinates $i_1, \ldots, i_{n'} \in \{1, \ldots, n\}$. Similarly, if R is an n-ary relation we let $\mathrm{Proj}_{i_1, \ldots, i_{n'}}(R) = \{\mathrm{Proj}_{i_1, \ldots, i_{n'}}(t) \mid t \in R\}$. The ith argument of a relation R is said to be *redundant* if there exists $j \neq i$ such that $t[i] = t[j]$ for each $t \in R$, and is said to be *fictitious* if for all $t \in R$ and $d \in D$ have $t' \in R$ where $t'[i] = d$ and $\mathrm{Proj}_{1, \ldots, i-1, i+1, \ldots, n}(t) = \mathrm{Proj}_{1, \ldots, i-1, i+1, \ldots, n}(t')$.

We write Eq_D for the equality relation $\{(x, x) \mid x \in D\}$ over D. We will often represent relations by their defining first-order formulas, and if $\varphi(x_1, \ldots, x_n)$ is a first-order formula with n free variables we write $R(x_1, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n)$ to define the relation $R = \{(f(x_1), \ldots, f(x_n)) \mid f \text{ is a model of } \varphi(x_1, \ldots, x_n)\}$. We let REL^n_D be the set of all n-ary relations over D, $\mathsf{REL}^{\leq n}_D = \bigcup_{i=1}^{n} \mathsf{REL}^n_D$, and $\mathsf{REL}_D = \bigcup_{i=1}^{\infty} \mathsf{REL}^i_D$. A set $\Gamma \subseteq \mathsf{REL}_D$ will sometimes be called a *constraint language*. Each k-ary operation $f \in \mathsf{OP}_D$ can be associated with a

$(k+1)$-ary relation $f^\bullet = \{(x_1, \ldots, x_k, f(x_1, \ldots, x_k)) \mid x_1, \ldots, x_k \in D\}$, called the *graph* of f.

2.2 Primitive Positive Definitions and Determined Variables

We say that an n-ary relation R has a *primitive positive definition* (pp-definition) over a set of relations Γ over a domain D 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 $\mathbf{x}_i$ is a tuple of variables over $x_1, \ldots, x_n, y_1, \ldots, y_{n'}$ of length $\mathrm{ar}(R_i)$ and each $R_i \in \Gamma \cup \{\mathrm{Eq}_D\}$. Hence, R can be defined as a (potentially) existentially quantified conjunctive formula over $\Gamma \cup \{\mathrm{Eq}_D\}$. We will occasionally be interested in pp-definitions not making use of existential quantification, and call pp-definitions of this restricted type *quantifier-free primitive positive definitions* (qfpp-definitions).

Definition 1. *Let R be an n-ary relation over a domain D. We say that $1 \leq i \leq n$ is* uniquely determined*, or just* determined*, if there exists $i_1, \ldots, i_k \in [n]$ and a function $h : D^k \to D$ such that $h(t[i_1], \ldots, \ldots, t[i_k]) = t[i]$ for each $t \in R$.*

When defining relations in terms of logical formulas we will occasionally also say that the ith variable is uniquely determined, rather than the ith index.

Definition 2. *An n-ary relation R has a* unique primitive positive definition *(upp-definition) over a set of relations Γ if there exists a pp-definition*

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

of R over Γ where each y_i is uniquely determined by $x_1, \ldots, x_n$.

We typically write $\exists! y_1, \ldots, y_{n'}$ for the existentially quantified variables in a upp-definition. Following Nordh & Zanuttini [24] we refer to unique existential quantification over constant arguments as *frozen existential quantification* ($i \in [\mathrm{ar}(R)]$ is constant if there exists $d \in D$ such that $t[i] = d$ for each $t \in R$). If R is upp-definable over Γ via a upp-definition only making use of frozen existential quantification then we say that R is *freezingly pp-definable* (fpp-definable) over Γ. Let us define the following closure operators over relations.

Definition 3. *Let Γ be a set of relations. Then we define (1) $\langle \Gamma \rangle = \{R \mid R$ has a pp-definition over $\Gamma\}$, (2), $\langle \Gamma \rangle_{\exists!} = \{R \mid R$ has a upp-definition over $\Gamma\}$, (3), $\langle \Gamma \rangle_{\mathrm{fr}} = \{R \mid R$ has an fpp-definition over $\Gamma\}$, and (4), $\langle \Gamma \rangle_{\nexists} = \{R \mid R$ has a qfpp-definition over $\Gamma\}$.*

In all cases Γ is called a *base*. If $\Gamma = \{R\}$ is singleton then we write $\langle R \rangle$ instead of $\langle \Gamma \rangle$, and similarly for the other operators. Sets of relations of the form $\langle \Gamma \rangle$ are usually called *relational clones*, or *co-clones*, sets of the form $\langle \Gamma \rangle_{\nexists}$ *weak systems*, or *weak partial co-clones*, and sets of the form $\langle \Gamma \rangle_{\mathrm{fr}}$ are known as *frozen partial co-clones*. Note that $\langle \Gamma \rangle \supseteq \langle \Gamma \rangle_{\exists!} \supseteq \langle \Gamma \rangle_{\mathrm{fr}} \supseteq \langle \Gamma \rangle_{\nexists}$ for any $\Gamma \subseteq \mathsf{REL}_D$.

Co-clones and weak systems can be described via algebraic invariants known as *polymorphisms* and *partial polymorphism*. More precisely, if $R \in \mathsf{REL}_D^n$ and $f \in \mathsf{OP}_D$ is a k-ary operation, then for $t_1, \ldots, t_k \in R$ we let $f(t_1, \ldots, t_k) = (f(t_1[1], \ldots, t_k[1]), \ldots, f(t_1[n], \ldots, t_k[n]))$. We then say that a k-ary partial operation f *preserves* an n-ary relation R if $f(t_1, \ldots, t_k) \in R$ or there exists $i \in [n]$ such that $(t_1[i], \ldots, t_k[i]) \notin \mathrm{dom}(f)$, for each sequence of tuples $t_1, \ldots, t_k \in R$. If f preserves R then R is also said to be *invariant* under f. Note that if f is total then the condition is simply that $f(t_1, \ldots, t_k) \in R$ for each sequence $t_1, \ldots, t_k \in R$. We then let $\mathrm{pPol}(R) = \{f \in \mathsf{pOP}_D \mid f \text{ preserves } R\}$, $\mathrm{Pol}(R) = \mathrm{pPol}(R) \cap \mathsf{OP}_D$, $\mathrm{pPol}(\Gamma) = \bigcap_{R \in \Gamma} \mathrm{pPol}(R)$, and $\mathrm{Pol}(\Gamma) = \bigcap_{R \in \Gamma} \mathrm{Pol}(R)$. Similarly, if F is a set of (partial) operations we let $\mathrm{Inv}(F)$ be the set of relations invariant under F, and write $\mathrm{Inv}(f)$ if $F = \{f\}$ is singleton. It is then known that $\mathrm{Inv}(F)$ is a co-clone if $F \subseteq \mathsf{OP}_D$ and that $\mathrm{Inv}(F)$ is a weak system if $F \subseteq \mathsf{pOP}_D$. More generally, $\langle \Gamma \rangle = \mathrm{Inv}(\mathrm{Pol}(\Gamma))$ and $\langle \Gamma \rangle_{\nexists} = \mathrm{Inv}(\mathrm{pPol}(\Gamma))$, resulting in the following *Galois connections*.

Table 1: Relations.

Relation	Definition
F	$\{(0)\}$
T	$\{(1)\}$
Ne	$\{(0,1),(1,0)\}$
n-EVEN	$\{(x_1,\ldots,x_n) \in \{0,1\}^n \mid x_1+\ldots+x_n \text{ is even}\}$
n-EVEN$^{n\neq}$	$n\text{-EVEN}(x_1,\ldots,x_n) \wedge \mathrm{Ne}(x_1,x_{n+1}) \wedge \ldots \wedge \mathrm{Ne}(x_n,x_{2n})$
n-ODD	$\{(x_1,\ldots,x_n) \in \{0,1\}^n \mid x_1+\ldots+x_n \text{ is odd}\}$
n-ODD$^{n\neq}$	$n\text{-ODD}(x_1,\ldots,x_n) \wedge \mathrm{Ne}(x_1,x_{n+1}) \wedge \ldots \wedge \mathrm{Ne}(x_n,x_{2n})$
NA^n	$\{0,1\}^n \setminus \{(1,\ldots,1)\}$

Theorem 4 ([3, 4, 12, 28])**.** *Let Γ and Δ be two sets of relations. Then $\Gamma \subseteq \langle\Delta\rangle$ if and only if $\mathrm{Pol}(\Delta) \subseteq \mathrm{Pol}(\Gamma)$ and $\Gamma \subseteq \langle\Delta\rangle_{\nexists}$ if and only if $\mathrm{pPol}(\Delta) \subseteq \mathrm{pPol}(\Gamma)$.*

Last, we remark that sets of the form $\mathrm{Pol}(\Gamma)$ and $\mathrm{pPol}(\Gamma)$ are usually called *clones*, and *strong partial clones*, respectively, and form lattices when ordered by set inclusion. Boolean clones are particularly well understood and the induced lattice is known as *Post's lattice* [26]. If $F \subseteq \mathsf{OP}_D$ then we write $[F]$ for the intersection of all clones over D containing F. Hence, $[F]$ is the smallest clone over D containing F.

2.3 Weak and Plain Bases of Co-Clones

In this section we introduce two special types of bases of a co-clone, that are useful for understanding the expressibility of upp-definitions.

Definition 5 (Schnoor & Schnoor [30])**.** *Let $\langle\Gamma\rangle$ be a co-clone. A base Γ_w of $\langle\Gamma\rangle$ with the property that $\langle\Gamma_w\rangle_{\nexists} \subseteq \langle\Delta\rangle_{\nexists}$ for every base Δ of $\langle\Gamma\rangle$ is called a* weak base *of $\langle\Gamma\rangle$.*

Although not immediate from Definition 5, Schnoor & Schnoor [30] proved that a weak base exists whenever $\langle\Gamma\rangle$ admits a finite base, by the following relational construction.

Definition 6. *For $s \geq 1$ we let $U^s_D = \{t_1,\ldots,t_s\}$ where $t_1,\ldots,t_s$ is the sequence of $|D|^s$-ary tuples such that $(t_1[1],\ldots,t_s[1]),\ldots,(t_1[|D|^s],\ldots,t_s[|D|^s])$ is a lexicographic enumeration of D^s.*

Given a relation R and a set of operations F over a domain D, we let

$$F(R) = \bigcap_{R' \in \mathrm{Inv}(F),\, R \subseteq R' \in \mathsf{REL}_D} R'.$$

We typically write U^s instead of U^s_D if the domain D is clear from the context, and say that a co-clone $\mathrm{Inv}(\mathsf{C})$ has *core-size* s if there exist relations R, R' such that $\mathrm{Pol}(R) = \mathsf{C}$, $R = \mathsf{C}(R')$, and $s = |R'|$. Weak bases can then be described via core-sizes as follows (a clone C is finitely related if there exists a finite base of $\mathrm{Inv}(\mathsf{C})$).

Theorem 7 (Schnoor & Schnoor [30])**.** *Let C be a finitely related clone where $\mathrm{Inv}(\mathsf{C})$ has core-size s. Then $\mathsf{C}(U^t)$ is a weak base of $\mathrm{Inv}(\mathsf{C})$ for every $t \geq s$.*

See Table 2 for a list of weak bases for the Boolean co-clones of interest in this paper [20, 21]. Here, and in the sequel, we use the co-clone terminology developed by Reith & Wagner [27] and Böhler et al. [5], where a Boolean co-clone $\mathrm{Inv}(\mathsf{C})$ is typically written as IC. Many relations in Table 2 are provided by their defining logical formulas; for example, $x_1 \rightarrow x_2$ is the binary relation $\{(0,0),(0,1),(1,1)\}$. See Table 1 for definitions of the remaining relations. As a convention we use c_0 to indicate a variable which is constant 0 in any model, and c_1 for a variable which is constant 1. On the functional side we use the bases by Böhler et al. [5] and let $\mathsf{I}_2 = [\pi^1_1], \mathsf{I}_0 = [0], \mathsf{I}_1 = [1], \mathsf{I} = [\{0,1\}]$, $\mathsf{N}_2 = [\overline{x}]$, $\mathsf{N} = [\{\overline{x},0,1\}]$, $\mathsf{E}_2 = [\wedge]$, $\mathsf{E}_0 = [\{\wedge,0\}]$, $\mathsf{E}_1 = [\{\wedge,1\}]$, $\mathsf{E} = [\{\wedge,0,1\}]$, $\mathsf{L}_2 = [x \oplus y \oplus z]$, and $\mathsf{S}_{11} = [\{x \wedge (y \vee z), 0\}]$, where $\overline{x} = 1 - x$ and

Table 2: Weak and plain bases of selected Boolean co-clones.

C	Weak base of $\mathrm{Inv}(\mathsf{C})$	Plain base of $\mathrm{Inv}(\mathsf{C})$
S_1^n	$\{\mathrm{NA}^n(x_1,\ldots,x_n) \wedge F(c_0)\}$	$\{\mathrm{NA}^n\}$
S_1	$\{\mathrm{NA}^n(x_1,\ldots,x_n) \wedge F(c_0) \mid n \geq 2\}$	$\{\mathrm{NA}^n \mid n \geq 1\}$
S_{12}^n	$\{\mathrm{NA}^n(x_1,\ldots,x_n) \wedge F(c_0) \wedge T(c_1)\}$	$\{\mathrm{NA}^n, T(c_1)\}$
S_{12}	$\{\mathrm{NA}^n(x_1,\ldots,x_n) \wedge F(c_0) \wedge T(c_1) \mid n \geq 2\}$	$\{\mathrm{NA}^n \mid n \geq 1\} \cup \{T(c_1)\}$
S_{11}^n	$\{\mathrm{NA}^n(x_1,\ldots,x_n) \wedge (\neg x \rightarrow \neg x_1 \cdots \neg x_n) \wedge F(c_0)\}$	$\{\mathrm{NA}^n, (x_1 \rightarrow x_2)\}$
S_{11}	$\{R_{\mathsf{S}_{11}^n} \mid n \geq 2\}$	$\{\mathrm{NA}^n \mid n \geq 1\} \cup \{(x_1 \rightarrow x_2)\}$
S_{10}^n	$\{R_{\mathsf{S}_{11}^n}(x_1,\ldots,x_n,c_0) \wedge T(c_1)\}$	$\{\mathrm{NA}^n, (x_1 \rightarrow x_2), T(c_1)\}$
S_{10}	$\{R_{\mathsf{S}_{10}^n} \mid n \geq 2\}$	$\{\mathrm{NA}^n \mid n \geq 1\} \cup \{(x_1 \rightarrow x_2), T(c_1)\}$
D	$\{(x_1 \oplus x_2 = 1)\}$	$\{(x_1 \oplus x_2 = 1)\}$
D_1	$\{(x_1 \oplus x_2 = 1) \wedge F(c_0)\} \wedge T(c_1)$	$\{(x_1 \oplus x_2 = 1)\} \cup \{F(c_0), T(c_1)\}$
D_2	$\{(x_1 \vee x_2) \wedge \mathrm{Ne}(x_1,x_3) \wedge \mathrm{Ne}(x_2,x_4) \wedge F(c_0) \wedge T(c_1)\}$	$\{(x_1 \vee x_2), (\neg x_1 \vee x_2), (\neg x_1 \vee \neg x_2)\}$
E	$\{(x_1 \leftrightarrow x_2x_3) \wedge (x_2 \vee x_3 \rightarrow x_4)\}$	$\{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \geq 1\}$
E_0	$\{(x_1 \leftrightarrow x_2x_3) \wedge (x_2 \vee x_3 \rightarrow x_4) \wedge F(c_0)\}$	$\{\mathrm{NA}^n \mid n \in \mathbb{N}\} \cup \{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \geq 1\}$
E_1	$\{(x_1 \leftrightarrow x_2x_3) \wedge T(c_1)\}$	$\{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \in \mathbb{N}\}$
E_2	$\{(x_1 \leftrightarrow x_2x_3) \wedge F(c_0) \wedge T(c_1)\}$	$\{\mathrm{NA}^n \mid n \in \mathbb{N}\} \cup \{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \in \mathbb{N}\}$

where $0, 1$ are shorthands for the two constant Boolean operations. We conclude this section by defining the dual notion of a weak base.

Definition 8 (Creignou et al. [9])**.** *Let $\langle\Gamma\rangle$ be a co-clone. A base Γ_p of $\langle\Gamma\rangle$ with the property that $\langle\Delta\rangle_{\nexists} \subseteq \langle\Gamma_p\rangle_{\nexists}$ for every base Δ of $\langle\Gamma\rangle$ is called a* plain base *of $\langle\Gamma\rangle$.*

Clearly, every co-clone is a trivial plain base of itself, but the question remains for which co-clones more succinct plain bases can be found. For arbitrary finite domains little is known but in the Boolean domain succinct plain bases have been described [9] (see Table 2).

2.4 Duality

Many questions concerning Boolean co-clones can be simplified by only considering parts of Post's lattice. If $f \in \mathsf{OP}_{\{0,1\}}$ is k-ary then the *dual* of f, $\mathrm{dual}(f)$, is the operation $\mathrm{dual}(f)(x_1,\ldots,x_k) = \overline{f(\overline{x_1},\ldots,\overline{x_k})}$, and we let $\mathrm{dual}(F) = \{\mathrm{dual}(f) \mid f \in F\}$ for a set $F \subseteq \mathsf{OP}_{\{0,1\}}$. Each Boolean clone C can then be associated with a dual clone $\mathrm{dual}(\mathsf{C})$. Similarly, for $R \in \mathsf{REL}_{\{0,1\}}$ we let $\mathrm{dual}(R) = \{\bar{t} \mid t \in R\}$ and $\mathrm{dual}(\Gamma) = \{\mathrm{dual}(R) \mid R \in \Gamma\}$ for $\Gamma \subseteq \mathsf{REL}_{\{0,1\}}$. It is then known that $\mathrm{Inv}(\mathrm{dual}(\mathsf{C})) = \mathrm{dual}(\mathrm{Inv}(\mathsf{C}))$.

3 The Expressive Power of Unique Existential Quantification

The main goal of this paper is to understand when the expressive power of unique existential quantification coincides with existential quantification in primitive positive formulas. Let us first consider an example where a pp-definition can be rewritten into a upp-definition.

Example 9. *Consider the canonical reduction from k-SAT to $(k-1)$-SAT via pp-definitions of the form $(x_1 \vee \ldots \vee x_k) \equiv \exists y\colon (x_1 \vee \ldots \vee x_{k-2} \vee y) \wedge (x_{k-1} \vee x_k \vee \neg y)$. In this pp-definition the auxiliary variable y is* not *uniquely determined since, for example, $y = 0$ and $y = 1$ are both consistent with $x_1 = 1, \ldots x_{k-2} = 1, x_{k-1} = 1, x_k = 1$. On the other hand, if we instead take the pp-definition $(x_1 \vee \ldots \vee x_k) \equiv \exists y\colon (x_1 \vee \ldots \vee x_{k-2} \vee y) \wedge (y \leftrightarrow (x_{k-1} \vee x_k))$, which can be expressed by $(k-1)$-SAT, it is easily verified that y is determined by x_{k-1} and x_k.*

Using the algebraic terminology from Section 2 this property can be phrased as follows.

Definition 10. *A co-clone $\langle\Gamma\rangle$ is* $\exists!$-covered *if $\langle\Gamma\rangle = \langle\Delta\rangle_{\exists!}$ for every base Δ of $\langle\Gamma\rangle$.*

Thus, we are interested in determining the $\exists!$-covered co-clones, and since every constraint language Γ belongs to a co-clone, namely $\langle\Gamma\rangle$, Definition 10 precisely captures the aforementioned

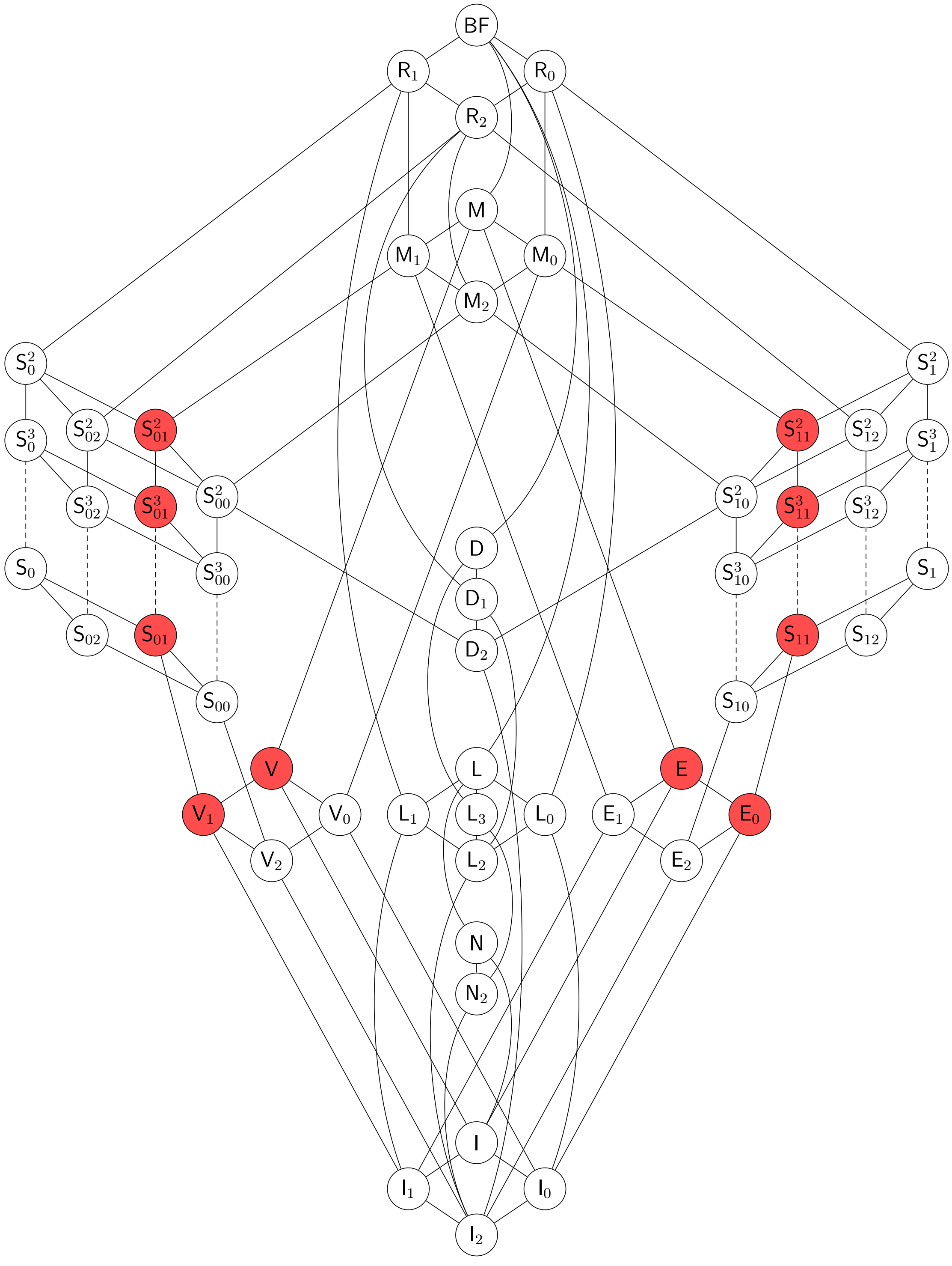

Figure 1: The lattice of Boolean clones. $\mathrm{Inv}(\mathsf{C})$ is coloured in red if and only if $\mathrm{Inv}(\mathsf{C})$ is not $\exists!$-covered.

question concerning the expressive strength of uniqueness quantification in primitive positive formulas. The remainder of this section will be dedicated to proving covering results of this form, with a particular focus on proving a full classification for the Boolean domain. See Figure 1 for a visualisation of this dichotomy. We begin in Section 3.1 by outlining some of the main ideas required to prove that a co-clone is $\exists!$-covered, and consider covering results applicable for arbitrary finite domains. In Section 3.2 we turn to the Boolean domain where we prove the classification in Figure 1.

3.1 General Constructions

Given an arbitrary constraint language Γ it can be difficult to directly reason about the strength of upp-definitions over Γ. Fortunately, there are methods to mitigate this difficulty. Recall from Definition 5 that a weak base of a co-clone $\langle\Gamma\rangle$ is a base which is qfpp-definable by any other base of $\langle\Gamma\rangle$, and that a plain base is a base with the property that it can qfpp-define every relation in the co-clone. We then have the following useful lemma.

Lemma 11. *Let $\langle\Gamma\rangle$ be a co-clone with a weak base Γ_w and a plain base Γ_p. If $\Gamma_p \subseteq \langle\Gamma_w\rangle_{\exists!}$ then $\langle\Gamma\rangle$ is $\exists!$-covered.*

Proof. Let Δ be a base of $\langle\Gamma\rangle$ and take an arbitrary n-ary relation $R \in \langle\Gamma\rangle$. First, take a qfpp-definition $R(x_1, \ldots, x_n) \equiv \varphi(x_1, \ldots, x_n)$ over Γ_p. By assumption, Γ_w can upp-define every relation in Γ_p, and it follows that

$$R(x_1, \ldots, x_n) \equiv \exists! y_1, \ldots, y_m \colon \varphi'(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

for a Γ_w-formula $\varphi'(x_1, \ldots, x_n, y_1, \ldots, y_m)$ since each constraint in $\varphi(x_1, \ldots, x_n)$ can be replaced by its upp-definition over Γ_w. Last, since Δ can qfpp-define Γ_w, we can obtain a upp-definition of R by replacing each constraint in $\varphi'(x_1, \ldots, x_n, y_1, \ldots, y_m)$ by its qfpp-definition over Δ. $\square$

Although not difficult to prove, Lemma 11 offers the advantage that it is sufficient to prove that $\Gamma_p \subseteq \langle\Gamma_w\rangle_{\exists!}$ for two constraint languages Γ_w and Γ_p. Let us now illustrate some additional techniques for proving that $\langle\Gamma\rangle$ is $\exists!$-covered. Theorem 7 in Section 2.3 shows that the relation $\mathsf{C}(U^s)$ is a weak base of $\mathrm{Inv}(\mathsf{C})$ for s larger than or equal to the core-size of $\mathrm{Inv}(\mathsf{C})$. For s smaller than the core-size we have the following description of $\mathsf{C}(U^s)$.

Theorem 12. *Let C be a finitely related clone over a finite domain D. Then, for every $s \geq 1$, $\mathsf{C}(U^s) \in \langle\Gamma\rangle_{\not\exists}$ for every base Γ of $\mathrm{Inv}(\mathsf{C})$.*

Proof. The intuitive meaning behind the relation $\mathsf{C}(U^s)$ is that it may be viewed as a relational representation of the set of all s-ary operations of a clone C, in the sense that there for each s-ary $f \in \mathsf{C}$ exists $t_f \in \mathsf{C}(U^s)$ such that $f(t_1, \ldots, t_s) = t_f$, where $\{t_1, \ldots, t_s\} = U^s$. Moreover, the operation f preserves $R \in \Gamma$ if and only if $(f(t_1[1], \ldots, t_s[1]), \ldots, f(t_1[n], \ldots, t_s[n])) \in R$. In a qfpp-definition of $\mathsf{C}(U^s)$ we then associate each variable x with an element of D^s, and then for each $R \in \Gamma$ and $t_1, \ldots, t_s \in R$ add the constraint $R((t_1[1], \ldots, t_s[1]), \ldots, (t_1[n], \ldots, t_s[n]))$. For further details, see Theorem 2 in Bodnarchuk et al. [4], or Theorem 15 in Dalmau [11]. $\square$

The applications of Theorem 12 in the context of upp-definitions might not be immediate. However, observe that each argument $i \in [|D|^s]$ of U^s is determined by at most s other arguments, and if C is sufficiently simple, this property can be proved to hold also for $\mathsf{C}(U^s)$. This intuition can then be formalised into the following general theorem.

Theorem 13. *Let $\mathrm{Pol}(\Gamma)$ be a clone over a finite domain D such that each $f \in \mathrm{Pol}(\Gamma)$ is a constant operation or a projection. Then $\langle\Gamma\rangle$ is $\exists!$-covered.*

Proof. Let F be a set of operations such that $[F] = \mathrm{Pol}(\Gamma)$. We may without loss of generality assume that $F = \{f_1, \ldots, f_k\}$ for unary operations f_l such that $f_l(x) = d_l$ for some $d_l \in D$. Take an arbitrary n-ary relation $R \in \langle \Gamma \rangle$. Let $s = |R|$ and consider the relation $F(U^s)$ from Definition 6. Our aim is to prove that $F(U^s)$ can upp-define R, which is sufficient since $F(U^s) \in \langle \Gamma \rangle_{\not\exists}$ via Theorem 12. Let $i_1, \ldots, i_n \in [|D|^s]$ denote the indices satisfying $\mathrm{Proj}_{i_1,\ldots,i_n}(F(U^s)) = R$.

If $k = 0$, and $\mathrm{Pol}(\Gamma)$ consists only of projections, then $F(U^s) = U^s$, and each argument in $[|D|^s] \backslash \{i_1, \ldots, i_n\}$ is already determined by $i_1, \ldots, i_n$, and by the preceding remark $R \in \langle F(U^s) \rangle_{\exists!}$. Therefore, assume that $k \geq 1$. For each $f_l \in F$ then observe that $(d_l, \ldots, d_l) \in F(U^s)$ and that $(d_l, \ldots, d_l) \in \mathrm{Proj}_{i_1,\ldots,i_n}(U^s)$. Choose $j_1, j_2 \in [|D|^s]$ such that $t[j_1] \neq t[j_2]$ for $t \in U^s$ if and only if $\mathrm{Proj}_{i_1,\ldots,i_n}(t) = (d_l, \ldots, d_l)$, for a d_l such that $f_l(x) = d_l$. Thus, we choose a pair of indices differing in U^s if and only if the projection on $i_1, \ldots, i_n$ is constant. Such a choice is always possible since the arguments of U^s enumerate all s-ary tuples over D. Then construct the relation $R'(x_1, \ldots, x_{|D|^s}) \equiv F(U^s)(x_1, \ldots, x_{|D|^s}) \wedge \mathrm{Eq}(x_{j_1}, x_{j_2})$. It follows that $\mathrm{Proj}_{i_1,\ldots,i_n}(R') = R$, and that every argument $l \in [|D|^s] \setminus \{i_1, \ldots, i_n\}$ is determined by $i_1, \ldots, i_n$. Hence, $R \in \langle F(U^s) \rangle_{\exists!}$. ☐

Theorem 13 implies that $\langle \Gamma \rangle$ is $\exists!$-covered if Γ is sufficiently powerful, and in particular implies that REL_D is $\exists!$-covered for every finite D. Hence, Γ pp-defines every relation if and only if Γ upp-defines every relation. However, as we will now illustrate, this is not the only possible case when a co-clone is $\exists!$-covered.

Lemma 14. *Let F be a set of operations over a finite domain D. If each argument $i \in [\mathrm{ar}(R)]$ is either fictitious or determined for every $R \in \mathrm{Inv}(F)$, then $\mathrm{Inv}(F)$ is $\exists!$-covered.*

Proof. Let Γ be a set of relations such that $\langle \Gamma \rangle = \mathrm{Inv}(F)$, and let $R \in \mathrm{Inv}(F)$ be an n-ary relation. Let $R(x_1, \ldots, x_n) \equiv \exists y_1, \ldots, y_m \colon \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ denote a pp-definition of R over Γ. First consider the relation $R^m(x_1, \ldots, x_n, y_1, \ldots, y_m) \equiv \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ which is upp-definable (indeed, even qfpp-definable) over Γ. Hence, R^m is preserved by F, implying that the $(n+m)$th argument is either fictitious or determined. In the first case we construct the relation

$$R^{m-1}(x_1, \ldots, x_n, y_1, \ldots, y_{m-1}) \equiv \exists! y_m \colon \mathrm{Eq}_D(x_1, y_m) \wedge \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m).$$

In the second case, we can directly upp-define the $(n+m-1)$-ary relation R^{m-1} as

$$R^{m-1}(x_1, \ldots, x_n, y_1, \ldots, y_{m-1}) \equiv \exists! y_m \colon \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m).$$

Since $R^{m-1} \in \langle \Gamma \rangle_{\exists!} \subseteq \mathrm{Inv}(F)$, it is clear that this procedure can be repeated until the relation R is upp-defined. ☐

Theorem 15. *Let D be a finite domain such that $|D|$ is prime, and let $f(x, y, z) = x - y + z \,(\mathrm{mod}\, |D|)$. Then, for any constraint language Γ over D such that $\langle \Gamma \rangle \subseteq \mathrm{Inv}(f)$, $\langle \Gamma \rangle$ is $\exists!$-covered.*

Proof. We will prove that the preconditions of Lemma 14 are satisfied for $\mathrm{Inv}(f)$, which is sufficient to prove the claim. Let R be invariant under f. Then it is known that R is the solution space of a system of linear equations modulo $|D|$ [15], from which it follows that each argument is either determined, since it can be written as a unique combination of other arguments, or is fictitious. ☐

3.2 Boolean Constraint Languages

In this section we use the techniques developed so far to prove that the classification in Figure 1 is correct. Note first that $\mathrm{Inv}(\mathsf{C})$ is $\exists!$-covered if and only if $\mathrm{Inv}(\mathrm{dual}(\mathsf{C}))$ is $\exists!$-covered, since a upp-definition $\exists! y_1, \ldots, y_{n'} \colon R_1(\mathbf{x}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m)$ of n-ary $R \in \mathrm{Inv}(\mathsf{C})$ immediately yields a upp-definition $\exists! y_1, \ldots, y_{n'} \colon \mathrm{dual}(R_1)(\mathbf{x}_1) \wedge \ldots \wedge \mathrm{dual}(R_m)(\mathbf{x}_m)$ of $\mathrm{dual}(R) \in \mathrm{Inv}(\mathrm{dual}(\mathsf{C}))$. Thus, to simplify the presentation we omit the case when $\mathsf{C} \supseteq \mathsf{V}_2$ in Figure 1. Let us begin with the cases following directly from Section 3.1 or from existing results (recall that IC is a shorthand for $\mathrm{Inv}(\mathsf{C})$).

Lemma 16. *Let* IC *be a Boolean co-clone. Then* IC *is* $\exists!$*-covered if* $\mathsf{IC} \subseteq \mathsf{IM}_2$, $\mathsf{IC} \subseteq \mathsf{IL}_2$, $\mathsf{IC} \subseteq \mathsf{IS}_{12}$, $\mathsf{IC} = \mathsf{IS}_{10}$, $\mathsf{IC} = \mathsf{IS}_{10}^n$ *for some* $n \geq 2$, $\mathsf{IC} = \mathsf{IS}_1$, *or* $\mathsf{IC} = \mathsf{IS}_1^n$ *for some* $n \geq 2$.

Proof. The case when $\mathsf{IC} \subseteq \mathsf{IL}_2$ follows from Theorem 15 since $\mathsf{L}_2 = [x \oplus y \oplus z]$. For each case when C belongs to the infinite chains in Post's lattice, or if $\mathsf{IC} \subseteq \mathsf{IM}_2$, it is known that $\mathsf{IC} = \langle \Gamma \rangle_{\mathrm{fr}}$ for any base Γ of IC [24], which is sufficient since $\langle \Gamma \rangle_{\mathrm{fr}} \subseteq \langle \Gamma \rangle_{\exists!}$. $\square$

We now move on to the more interesting cases, and begin with the case when $\mathrm{Pol}(\Gamma)$ is essentially unary, i.e., consists of essentially unary operations. This covers $\mathsf{I}_2, \mathsf{I}_0, \mathsf{I}_1, \mathsf{I}, \mathsf{N}_2, \mathsf{N}$ from Figure 1.

Theorem 17. *Let* Γ *be a Boolean constraint language such that* $\mathrm{Pol}(\Gamma)$ *is essentially unary. Then* $\langle \Gamma \rangle$ *is* $\exists!$*-covered.*

Proof. From Theorem 13 only the two co-clones IN and IN_2 remain, where $\mathsf{IN} = \mathrm{Inv}(\{\overline{x}, 0, 1\})$ and $\mathsf{IN}_2 = \mathrm{Inv}(\overline{x})$. The two cases are similar and we for brevity concentrate only on IN_2. Hence, let $R \in \mathsf{IN}_2$ be an n-ary relation, which we without loss of generality may assume has no redundant arguments. Since $\mathsf{N}_2 = [\overline{x}]$ we begin by partitioning R into two disjoint sets S and $\overline{S}$ where $t \in S$ if and only if $\overline{t} \in \overline{S}$. Let $s = |S| = |\overline{S}|$, and construct the relation $\{\overline{x}\}(U^s)$, which is qfpp-definable over Γ according to Theorem 12. Let $i_1, \ldots, i_n \in \{1, \ldots, n\}$ be the set of indices satisfying $\mathrm{Proj}_{i_1,\ldots,i_n}(U^s) = S$ and $\mathrm{Proj}_{i_1,\ldots,i_n}(\{\overline{x}\}(U^s) \setminus U^s) = \overline{S}$. Assume there exists $i \in [2^s] \setminus \{i_1, \ldots, i_n\}$ such that i is not determined by $i_1, \ldots, i_n$. By construction, i is determined by $i_1, \ldots, i_n$ in both U^s and $\{\overline{x}\}(U^s) \setminus U^s$, implying that the only possible outcome is the existence of $t \in U^s$ and $t' \in \{\overline{x}\}(U^s) \setminus U^s$ where $t[i] \neq t'[i]$, but $\mathrm{Proj}_{i_1,\ldots,i_n}(t) = \mathrm{Proj}_{i_1,\ldots,i_n}(t')$. But then $\overline{t'} \in U^s$, and since $\mathrm{Proj}_{i_1,\ldots,i_n}(U^s) = S$ and $\mathrm{Proj}_{i_1,\ldots,i_n}(t) = \mathrm{Proj}_{i_1,\ldots,i_n}(t')$, $\overline{\mathrm{Proj}_{i_1,\ldots,i_n}(t')} \in S$, contradicting the partitioning of R into the disjoint sets S and $\overline{S}$. Then, since every argument i distinct from $i_1, \ldots, i_n$ is determined by $i_1, \ldots, i_n$, the original relation R can be upp-defined by $\{\overline{x}\}(U^s)$ using only unique existential quantification.

The IN case is similar, and the only difference is that we partition the input relation R into $S, \overline{S}$, and $\mathbf{0}, \mathbf{1}$, and proceed with $s = |S| = |\overline{S}|$. $\square$

Next, we consider the co-clone ID_2, consisting of all relations pp-definable by binary clauses.

Lemma 18. ID_2 *is* $\exists!$*-covered.*

Proof. We will show the result by using Lemma 11. According to Table 2 the relation $R_w(x_1, x_2, x_3, x_4, c_0, c_1) \equiv$

$$(x_1 \vee x_2) \wedge \mathrm{Ne}(x_1, x_3) \wedge \mathrm{Ne}(x_2, x_4) \wedge F(c_0) \wedge T(c_1)$$

is a weak base of ID_2, and the plain base Γ_p is given by $\Gamma_p = \{(x_1 \vee x_2), (\neg x_1 \vee x_2), (\neg x_1 \vee \neg x_2)\}$. Hence, we need to prove that R_w can upp-define each relation in Γ_p. Now consider the following upp-definitions:

$$(x_1 \vee x_2) \equiv \exists! x_3, x_4, c_0, c_1 \colon R_w(x_1, x_2, x_3, x_4, c_0, c_1),$$

$$(\neg x_1 \vee x_2) \equiv \exists! x_3, x_4, c_0, c_1 \colon R_w(x_3, x_2, x_1, x_4, c_0, c_1),$$

and

$$(\neg x_1 \vee \neg x_2) \equiv \exists! x_3, x_4, c_0, c_1 \colon R_w(x_3, x_4, x_1, x_2, c_0, c_1).$$

In each case it is readily verified that the existentially quantified variables are indeed uniquely determined. For example, in the upp-definition of $(\neg x_1 \vee x_2)$ the variable x_3 is uniquely determined by x_1 since $t[1] \neq t[3]$ for every $t \in R_w$. $\square$

We now tackle the cases when $\mathrm{Inv}(\{\wedge, 0, 1\}) \subseteq \mathsf{IC} \subseteq \mathrm{Inv}(\{\wedge\})$, which in Figure 1 corresponds to E, E_0, E_1, and E_2. As a first step we begin by characterising the determined arguments of relations in E_0.

Lemma 19. *Let $R \in \mathsf{IE}_0$ be an n-ary relation. If $i \in [n]$ is determined in R then either (1) there exists $i_1, \ldots, i_k \in [n]$ distinct from i such that $t[i] = t[i_1] \wedge \ldots \wedge t[i_k]$ for every $t \in R$, or (2) $t[i] = 0$ for every $t \in R$.*

Proof. Assume that $i \in [n]$ is determined in R. Let $R_1 = \{t_1, \ldots, t_m\} = \{t \in R \mid t[i] = 1\}$ and $R_0 = \{s_1, \ldots, s_{m'}\} = \{s \in R \mid t[i] = 0\}$. Note first that $R_0 = \emptyset$ cannot happen since R is preserved by 0, and if $R_1 = \emptyset$ then we end up in case (2). Hence, in the remainder of the proof we assume that R_0 and R_1 are both non-empty.

Consider the tuple $t_1 \wedge \ldots \wedge t_m = t$ (applied componentwise), and observe that $t \in \{t_1, \ldots, t_m\}$ since R is preserved by $\wedge$, and that $t[i] = 1$ since $t_1[i] = \ldots = t_m[i] = 1$. Furthermore, if $t[j] = 1$ for some $j \in [n]$ then it must also be the case that $t_1[j] = \ldots = t_m[j] = 1$. Let $i_1, \ldots, i_l \in [n] \setminus \{i\}$ denote the set of indices such that $t[i_j] = 1$. Then $t'[i] = t'[i_1] \wedge \ldots \wedge t'[i_l]$ for every $t' \in R_1$, and we also claim that $s[i] = s[i_1] \wedge \ldots \wedge s[i_l]$ for every $s \in R_0$, thus ending up in case (1). Note that $l > 0$, as otherwise every argument distinct from i is constantly 0 in t, which is not consistent with the fact that $\mathbf{0}^n \in R_0$, since it contradicts the assumption that i is determined. Assume that there exists $s \in R_0$ such that $s[i] = 0 \neq s[i_1] \wedge \ldots \wedge s[i_l]$. Then, clearly, $s[i_1] = \ldots = s[i_l] = 1$. But then $t \wedge s \in R$ implies that i is not determined, since $\mathrm{Proj}_{1,\ldots,i-1,i+1,\ldots,n}(t \wedge s) = \mathrm{Proj}_{1,\ldots,i-1,i+1,\ldots,n}(t)$ but $(t \wedge s)[i] \neq t[i]$. Hence, $s[i] = s[i_1] \wedge \ldots \wedge s[i_l]$ for every $s \in R$, which concludes the proof. $\square$

Lemma 19 also shows that if $R \in \mathsf{IE}$ with a determined argument i then there exists $i_1, \ldots, i_k \in [\mathrm{ar}(R)]$ such that $t[i] = t[i_1] \wedge \ldots \wedge t[i_k]$ for every $t \in R$, since the constant relation $\{(0)\} \notin \mathsf{IE}$. Before we use Lemma 19 to show the non-covering results for IE and IE_0, we will need the following lemma, relating the existence of a upp-definition to a qfpp-definition of a special form. The proof essentially follows directly from the statement of the lemma and is therefore omitted.

Lemma 20. *Let Γ be a constraint language. Then an n-ary relation $R \in \langle\Gamma\rangle_{\exists!}$ has a upp-definition $R(x_1, \ldots, x_n) \equiv \exists! y_1, \ldots, y_m \colon \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ if and only if there exists an $(n+m)$-ary relation $R' \in \langle\Gamma\rangle_{\nexists}$ such that $\mathrm{Proj}_{1,\ldots,n}(R') = R$ where each $n < i \leq n+m$ is determined by $1, \ldots, n$.*

Say that a partial operation f is *$\wedge$-closed* if $\mathrm{dom}(f)$ is preserved by $\wedge$ and that it is *0-closed* if $\mathbf{0}^{\mathrm{ar}(f)} \in \mathrm{dom}(f)$. We may now describe partial polymorphisms of $\langle\Gamma\rangle_{\exists!}$ using $\wedge$-closed and 0-closed partial polymorphisms of Γ.

Lemma 21. *Let Γ be a constraint language such that $\langle\Gamma\rangle = \mathsf{IE}_0$. If $f \in \mathrm{pPol}(\Gamma)$ is $\wedge$- and 0-closed then $f \in \mathrm{pPol}(\langle\Gamma\rangle_{\exists!})$.*

Proof. Let $R \in \langle\Gamma\rangle_{\exists!}$ be an n-ary relation and let $R' \in \langle\Gamma\rangle_{\nexists}$ be the $(n+m)$-ary relation from Lemma 20 where $\mathrm{Proj}_{1,\ldots,n}(R') = R$ and each $i \in \{n+1, \ldots, n+m\}$ is determined in R'. Since f preserves each relation in Γ it follows that f also preserves R' since $R' \in \langle\Gamma\rangle_{\nexists}$. Assume, with the aim of reaching a contradiction, that there exists $s_1, \ldots, s_k \in R$ such that $f(s_1, \ldots, s_k) \notin R$ ($\mathrm{ar}(f) = k$), and let $t_1, \ldots, t_n$ be the corresponding tuples in R' — guaranteed to exist due to the construction of R'. Now, for each $i \in \{n+1, \ldots, n+m\}$, either there exists $i_1, \ldots, i_\ell \in [n]$ such that $t[i] = t[i_1] \wedge \ldots \wedge t[i_\ell]$ for each $t \in \{t_1, \ldots, t_k\}$, or $t_j[i] = 0$ for each $t_j \in \{t_1, \ldots, t_k\}$ (from Lemma 19). But since $(t_1[i_1], \ldots, t_k[i_1]), \ldots, (t_1[i_\ell], \ldots, t_k[i_\ell]) \in \mathrm{dom}(f)$ (since $f(s_1, \ldots, s_k)$ is defined) it follows that $(t_1[i], \ldots, t_k[i]) \in \mathrm{dom}(f)$, too, due to the assumption that f is $\wedge$-closed and 0-closed. However, then $f(t_1, \ldots, t_k)$ is also defined and it follows that $f(t_1, \ldots, t_k) \notin R'$, contradicting the assumption that $f \in \mathrm{pPol}(\Gamma)$. $\square$

We now have all the technical machinery in place to prove that IE_0 and IE are not $\exists!$-covered.

Theorem 22. *Let R_w be the weak base of IE_0 from Table 2. Then $\langle R_w \rangle_{\exists!} \subset \mathsf{IE}_0$.*

Proof. We prove that the relation $R(x_1, x_2, x_3) \equiv x_1 \leftrightarrow x_2x_3$ is not upp-definable over R_w, which is sufficient since $R \in \mathsf{IE}_0$, as evident in Table 2. Furthermore, using Lemma 20, we only have to prove that any $(3+n)$-ary R' where $\mathrm{Proj}_{1,2,3}(R') = R$, and where each other argument is determined by the three first, is not included in $\langle R_w \rangle_{\not\exists}$. Assume, without loss of generality, that R' does not contain any redundant arguments. Define the binary partial operation f such that $f(0,0) = 0$, $f(0,1) = f(1,0) = 1$. By construction, f is both 0-closed and $\wedge$-closed, and it is also readily verified that f preserves R_w, which via Lemma 21 then implies that $f \in \mathrm{pPol}(\langle R_w \rangle_{\exists!})$. To finish the proof we also need to show that $f \notin \mathrm{pPol}(R')$, which is sufficient since it implies that $R' \notin \langle R_w \rangle_{\exists!}$. Take two tuples $s, t \in R'$ such that $\mathrm{Proj}_{1,2,3}(s) = (0,0,1)$, and $\mathrm{Proj}_{1,2,3}(t) = (0,1,0)$. From Lemma 19, for each $3 < i \leq n+3$, either i is constant 0 in R' or there exists $i_1, \ldots, i_k \in \{1,2,3\}$, $k \leq 3$, such that $t[i] = t[i_1] \wedge \ldots t[i_k]$ for each $t \in R'$. But then $(s[i], t[i]) \in \mathrm{dom}(f)$ for each $3 < i \leq n+3$, since either $(s[i], t[i]) = (0,0) \in \mathrm{dom}(f)$ or $(s[i], t[i])$ is a conjunction over $(0,0,1)$ and $(0,1,0)$. However, this implies that $f(s,t) = u \notin R'$ since $\mathrm{Proj}_{1,2,3}(u) = (0,1,1)$. Hence, f does not preserve R', and $R' \notin \langle R_w \rangle_{\not\exists}$ via Theorem 4. $\square$

The proof for IE uses the same construction and we omit the details. Surprisingly, as we will now see, IE_1 and IE_2 behave entirely differently and are in fact $\exists!$-covered.

Lemma 23. *IE_1 and IE_2 are $\exists!$-covered.*

Proof. We begin with IE_1. Let $R_{\mathsf{IE}_1}(x_1, x_2, x_3, x_4) \equiv (x_1 \leftrightarrow x_2x_3) \wedge T(x_4)$ be the weak base of IE_1, and $\Gamma_p = \{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \geq 0\}$ the plain base of IE_1, from Table 2. First, note that for $k = 0$ the relation $(\neg x_1 \vee \ldots \vee x_k \vee x) \equiv T(x)$ and may be upp-defined by $T(x) \equiv R_{\mathsf{IE}_1}(x, x, x, x)$. Second, observe that if we can qfpp-define $(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \equiv (x_1 \ldots x_k \rightarrow x)$ then we can also qfpp-define $(x_1 \ldots x_k) \leftrightarrow x$, since (1) $(x_1 \rightarrow x_2) \equiv (x_1 \ldots x_1) \rightarrow x_2$ and (2) $(x_1 \ldots x_k) \leftrightarrow x \equiv x_1 \ldots x_k \rightarrow x \wedge (x \rightarrow x_1) \wedge \ldots \wedge (x \rightarrow x_k)$. We may then upp-define $(x_1 \ldots x_k \rightarrow x)$ for $k \geq 1$ as

1. $x_1 \rightarrow x \equiv \exists! y \colon R_{\mathsf{IE}_1}(x_1, x_1, x, y)$,
2. $(x_1x_2 \rightarrow x) \equiv \exists! x_3, x_4 \colon R_{\mathsf{IE}_1}(x_3, x_1, x_2, x_4) \wedge x_3 \rightarrow x$, and for $k \geq 3$
3. $(x_1 \ldots x_k \rightarrow x) \equiv \exists! y \colon (x_1 \ldots x_{k-1} \leftrightarrow y) \wedge (x_k y) \rightarrow x$,

using the upp-definable relation $(x_1 \ldots x_{k-1} \rightarrow x)$ at level $k-1$.

Let us now consider IE_2. Let $R_{\mathsf{IE}_2}(x_1, x_2, x_3, x_4, x_5) \equiv (x_1 \leftrightarrow x_2x_3) \wedge F(x_4) \wedge T(x_5)$ be the weak base of IE_2, and $\Gamma_p = \{\mathrm{NA}^n(x_1, \ldots, x_n) \mid n \in \mathbb{N}\} \cup \{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \geq 0\}$ the plain base of IE_2, from Table 2. Since the upp-definitions of $\{(\neg x_1 \vee \ldots \vee \neg x_k \vee x) \mid k \geq 0\}$ are similar to the IE_1 case we only present the upp-definitions of $\{\mathrm{NA}^n(x_1, \ldots, x_n) \mid n \in \mathbb{N}\}$. First, observe that $R_{\mathsf{IE}_1}(x_1, x_2, x_3, x_4) \equiv \exists! x_5 \colon R_{\mathsf{IE}_2}(x_1, x_2, x_3, x_5, x_4)$, implying that the relation $(x_1 \ldots x_k \rightarrow x)$ is upp-definable over R_{IE_2}. Then $\mathrm{NA}^k(x_1, \ldots, x_k)$ can be upp-defined as $\mathrm{NA}^k(x_1, \ldots, x_k) \equiv \exists! x, y \colon (x \leftrightarrow x_1 \ldots x_k) \wedge R_{\mathsf{IE}_2}(x, x, x, x, y)$. $\square$

The natural generalisation of the Boolean operations $\wedge$ and $\vee$ are so-called *semilattice operations*; binary operations that are idempotent, associative, and commutative. It is then tempting to conjecture that Lemma 19 can be generalized to arbitrary semilattice operations, i.e., that every determined argument can be described as a semilattice combination of other arguments, whenever a relation is preserved by a given semilattice operation. This, however, is not true. For a simple counterexample define the semilattice operation $s\colon \{0,1,2\}^2 \to \{0,1,2\}$ as $s(x,x) = x$ and $s(x,y) = 0$ otherwise. If we then consider the relation $R = \{(0,0),(1,1),(2,0)\}$ it is easily verified that s preserves R, and that the second argument is uniquely determined by the first argument but cannot be described via the operation s.

The only co-clones remaining are IS_{11} and IS_{11}^n (for $n \geq 2$). As we will see, unique existential quantification is only as powerful as frozen quantification for these co-clones. We state the following lemma only for IS_{11} but the same construction is valid also for IS_{11}^n.

Lemma 24. *Let Γ be a constraint language such that $\langle \Gamma \rangle = \mathsf{IS}_{11}$. Then $\langle \Gamma \rangle_{\exists !} = \langle \Gamma \rangle_{\mathrm{fr}}$.*

Proof. Let R be an n-ary relation in IS_{11}. Our aim is to prove that if an argument $i \in [n]$ of R is determined then i is either redundant or constant 0. This is sufficient to prove the claim since any upp-definition over Γ can then be transformed into an fpp-definition over Γ.

Hence, assume that i is neither redundant nor constant 0. Recall that $\mathsf{IS}_{11} \subset \mathsf{IE}_0$ and that $[\{x \wedge (y \vee z), 0\}] = \mathsf{S}_{11}$. Lemma 19 then implies that such an i can be described as a conjunction of other arguments, i.e., that there exists $i_1, \ldots, i_k \in [k]$ distinct from i such that $t[i] = t[i_1] \wedge \ldots \wedge t[i_k]$ for every $t \in R$. Note that $k > 1$ as otherwise i is redundant in R. Partition R into two sets R_0 and R_1 such that $R_0 = \{t \in R \mid t[i] = 0\}$ and $R_1 = \{t \in R \mid t[i] = 1\}$; both non-empty by our assumption that i is non-constant. By the assumption that i is determined by a conjunction of $i_1, \ldots, i_k$ it first follows that there exists $s \in R_1$ where $s[i_1] = \ldots = s[i_k] = 1$, and that there for each $i_j \in \{i_1, \ldots, i_k\}$ exists a tuple $s_{i_j} \in R_0$ such that $s_{i_j}[i_j] \neq s_{i_j}[i] = 0$, as otherwise $t[i] = t[i_j]$ for each $t \in R$, and i is redundant. Now, consider an application of the form $s \wedge (s_{i_j} \vee s_{i_l}) = s'$ for $i_j, i_l \in \{i_1, \ldots, i_k\}$. Since $\mathrm{Proj}_{i_1,\ldots,i_k}(s) = (1, \ldots, 1)$, we have that $\mathrm{Proj}_{i_1,\ldots,i_k}(s') = \mathrm{Proj}_{i_1,\ldots,i_k}(s_{i_j}) \vee \mathrm{Proj}_{i_1,\ldots,i_k}(s_{i_l})$, and since $s_{i_j}[i] = s_{i_l}[i] = 0$, we also know that $s'[i] = 0$. If we then consider the application $s' = s \wedge (s_{i_1} \vee (s \wedge (s_{i_2} \vee (\ldots \vee (s \wedge (s_{i_{k-1}} \vee s_{i_k})) \cdots))))$ it follows that $\mathrm{Proj}_{i_1,\ldots,i_k}(s') = \mathrm{Proj}_{i_1,\ldots,i_k}(s)$ since $\mathrm{Proj}_{i_1,\ldots,i_k}(s') = \mathrm{Proj}_{i_1,\ldots,i_k}(s_{i_1}) \vee \ldots \vee \mathrm{Proj}_{i_1,\ldots,i_k}(s_{i_k})$, and, furthermore, that $s' \in R_0$ since $s'[i] = 0$. This contradicts the assumption that i is determined by $i_1, \ldots, i_k$, and we conclude that i must be redundant or constant 0 if it is determined in R. $\square$

It thus only remains to prove that IS_{11} and IS_{11}^n do not collapse into a single frozen co-clone. Here, we state the lemma only for IS_{11}^n, but the same argument works for IS_{11}.

Lemma 25. *Let Γ_p denote the plain base and Γ_w the weak base of IS_{11}^n ($n \geq 2$) from Table 2. Then $\langle \Gamma_w \rangle_{\mathrm{fr}} \subset \langle \Gamma_p \rangle_{\mathrm{fr}}$.*

Proof. We will show that there exists a partial operation f such that $f(\mathbf{0}^n) = 0$, and such that f preserves Γ_w but not Γ_p, which is sufficient to prove the claim according to Theorem 12 in Nordh & Zanuttini [24]. Let f be defined on $s_1, \ldots, s_n \in \{0,1\}^n$ such that the Hamming weight of each s_i is equal to 1, and such that $f(s_1) = \ldots = f(s_n) = 1$. By definition, f does not preserve $\{0,1\}^n \setminus \{\mathbf{1}^n\}$, and does therefore not preserve Γ_p, either. We now claim that f preserves Γ_w. Indeed, consider an application $f(t_1, \ldots, t_n)$ for $t_1, \ldots t_n \in R_w$. Then either (1) there exists i such that $(t_1[i], \ldots, t_n[i])$ has Hamming weight larger than 1, or (2) $\{t_1, \ldots, t_n\} = \{\mathbf{0}^{n+2}, t_i\}$ for some $i \in [n]$. To see why the second case is true, simply note that $t[n+1] = 1$ for every $t \in \Gamma_w \setminus \{\mathbf{0}^{n+2}\}$, and if we insist that the Hamming weight of each $(t_1[i], \ldots, t_n[i])$ is smaller than or equal to 1, then the sequence $t_1, \ldots, t_n$ contains $n-1$ or n repeated applications of $\mathbf{0}^{n+2}$. We can now wrap up the proof by a simple case analysis of the tuple t_i. First, assume that $t_i = \mathbf{0}^{n+2}$. In this case $f(t_1, \ldots, t_n) = \mathbf{0}^{n+2} \in \Gamma_w$. Second, assume that $\mathrm{Proj}_{1,\ldots,n}(t_i)$ has Hamming weight between 1 and $n-1$ (the case where it is equal to n is impossible). In this case there exists j such that $(t_1[j], \ldots, t_n[j]) = \mathbf{0}^n$, implying that $f(t_1, \ldots, t_n) = t$ for a tuple where $\mathrm{Proj}_{1,\ldots,n}(t) \neq \mathbf{1}^n$, $t[n+1] = 1$, and $t[n+2] = 0$, and hence that $t \in \Gamma_w$. $\square$

Combining the results in this section we can now finally prove our dichotomy theorem.

Theorem 26. *Let $\langle\Gamma\rangle$ be a Boolean co-clone. Then $\langle\Gamma\rangle$ is not $\exists!$-covered if and only if*

1. $\langle\Gamma\rangle \in \{\mathsf{IE}, \mathsf{IE}_0, \mathsf{IV}, \mathsf{IV}_1\}$, *or*
2. $\langle\Gamma\rangle \in \{\mathsf{IS}^n_{01}, \mathsf{IS}^n_{11} \mid n \geq 2\} \cup \{\mathsf{IS}_{01}, \mathsf{IS}_{11}\}$ *(where, in addition, $\langle\Gamma\rangle_{\exists!} = \langle\Gamma\rangle_{\mathrm{fr}}$).*

Proof. Each negative case either follows immediately from Lemma 22, Lemma 24, Lemma 25, or is the dual of one of those cases. Each $\exists!$-covered co-clone is proved in Lemma 16, Theorem 17, Lemma 18, and Lemma 23. □

4 Applications in Complexity

In this section we apply Theorem 26 to study the complexity of computational problems not compatible with pp-definitions. Let us begin by defining the *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)).

INSTANCE: A tuple (V, C) where V is a set of variables and C a set of constraints of the form $R_i(x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R)}})$ for $R_i \in \Gamma$.
QUESTION: Does (V, C) have at least one model? That is, a function $f\colon V \to D$ such that $f(x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R_i)}}) \in R_i$ for each $R_i(x_{i_1}, \ldots, x_{i_{\mathrm{ar}(R_i)}}) \in C$?

For Boolean constraint languages Γ we write SAT(Γ) instead of CSP(Γ). If $\Delta \subseteq \langle\Gamma\rangle$ (or, equivalently, $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta)$) then CSP($\Delta$) is polynomial-time reducible to CSP(Γ) [14]. However, there exist many natural variants of CSPs not compatible with pp-definitions, but compatible with more restricted closure operators such as upp-definitions. One such example is the *unique satisfiability problem* over a Boolean constraint language Γ (U-SAT(Γ)).
INSTANCE: A SAT(Γ) instance I.

QUESTION: Does I have a unique model?

The unrestricted U-SAT problem, i.e., the U-SAT problem where all possible constraints are allowed, can be seen as the intersection of satisfiability (in NP), and the satisfiability problem of checking if a given instance does not admit two distinct models (in co-NP). Hence, U-SAT is included in the second level of the Boolean hierarchy, BH_2, but is not believed to be complete for this class [25]. This unclear status motivated Blass and Gurevich [2] to introduce the complexity class *unique polynomial-time*, US, the set of decision problems solvable by a non-deterministic polynomial-time Turing machine where an instance is a yes-instance if and only if there exists a unique accepting path. Blass and Gurevich then quickly observed that U-SAT is US-complete and that $\mathrm{US} \subseteq \mathrm{BH}_2$.

We will present a simple, algebraic proof of Juban's trichotomy theorem for U-SAT(Γ) [18], showing that U-SAT(Γ) for finite Γ is either tractable, co-NP-complete, or US-complete. Using our machinery we will also be able to generalise this result to arbitrary infinite constraint languages. However, for infinite Γ we first need to specify a method of representation. We assume that the elements $R_1, R_2, \ldots$ of Γ are recursively enumerable by their arity, are represented as lists of tuples, and that there exists a computable function $f\colon \mathbb{N} \to \mathbb{N}$ such that for every $k \geq 1$ and every k-ary relation R, $R \in \langle\Gamma\rangle_{\exists!}$ if and only if $R \in \langle\Gamma \cap \mathsf{REL}^{\leq f(k)}_{\{0,1\}}\rangle_{\exists!}$. Thus, if a relation is upp-definable it is always possible to bound the arities of the required relations in the definition. The complexity of U-SAT(Γ) is then determined by $\langle\Gamma\rangle_{\exists!}$ in the following sense.

Theorem 27. *Let Γ and Δ be Boolean constraint languages. If $\Delta \subseteq \langle\Gamma\rangle_{\exists!}$ is finite then* U-SAT(Δ) *is polynomial-time many-one reducible to* U-SAT(Γ).

Proof. By assumption every $R \in \Delta$ is upp-definable over Γ. First let $k = \max\{f(\mathrm{ar}(R)) \mid R \in \Delta\}$. We then begin by computing a upp-definition of R over $\Gamma \cap \mathsf{REL}^{\leq k}_{\{0,1\}}$, and store this upp-definition in a table. Since Δ is finite this can be done in constant time. Next, given an instance $I = (V, C)$ of U-SAT(Δ), we similar to the ordinary CSP case simply replace each constraint in C by its upp-definition over Γ, and identify any potential variables occurring in equality constraints. This procedure might introduce additional variables, but since they are all determined by V, the existence of a unique model is preserved. $\square$

Theorem 28. *Let Γ be a Boolean constraint language. Then* U-SAT(Γ) *is co-NP-complete if $\langle\Gamma\rangle \in \{\mathsf{II}_0, \mathsf{II}_1\}$, US-complete if $\langle\Gamma\rangle = \mathsf{II}_2$, and is tractable otherwise.*

Proof. We begin with the tractable cases and assume that $\langle\Gamma\rangle \notin \{\mathsf{II}_0, \mathsf{II}_1, \mathsf{II}_2\}$. If $\langle\Gamma\rangle = \mathsf{IN}_2$ or $\langle\Gamma\rangle = \mathsf{IN}$ then any instance of U-SAT(Γ) is either unsatisfiable or has at least two models, since any $R \in \mathsf{IN} \subset \mathsf{IN}_2$ is closed under complement. Then, trivially, U-SAT(Γ) can be solved in constant time by always answering no. Similarly, if $\langle\Gamma\rangle = \mathsf{II}$ then any instance of U-SAT(Γ) has at least two models, and we simply answer no. Every other case can then be solved efficiently by enumerating models wih polynomial delay [29], by answering no if more than one model is found.

For the intractable cases assume first that $\langle\Gamma\rangle = \mathsf{II}_0$. Membership in co-NP is clear since a no-instance can be verified by any non-constant model. Let UNSAT(Γ) denote the unsatisfiability problem over a Boolean constraint language Γ, and consider the problem UNSAT$(\{R\})$ where $R = \{(0,0,1,0,1), (0,1,0,0,1), (1,0,0,0,1)\}$. It is readily seen that $\langle R\rangle = \mathsf{II}_2$ since R is only preserved by projections, implying that UNSAT$(\{R\})$ is co-NP-complete, and we will show co-NP-hardness of U-SAT(Γ) by a polynomial-time many-one reduction from UNSAT$(\{R\})$ to U-SAT$(\{R \cup \{(0,0,0,0,0)\}\})$. Given an instance (V, C) of UNSAT$(\{R\})$ we begin by introducing one fresh variable c_1, and for each variable x_i^5 occurring in a constraint $R(x_i^1, x_i^2, x_i^3, x_i^4, x_i^5) \in C$ we replace x_i^5 with c_1. Then, we for each constraint $R(x_i^1, x_i^2, x_i^3, x_i^4, c_1)$ replace it by $(R \cup \{(0,0,0,0,0)\})(x_i^1, x_i^2, x_i^3, x_i^4, c_1)$, and let I' be the resulting instance of U-SAT$(\{R \cup \{(0,0,0,0,0)\}\})$. The fresh variable c_1 which occurs in every constraint ensures that if a constraint is satisfied by assigning all variables 0, then all other variables have to be assigned 0 as well. It follows that I' admits a unique model, namely the model where each variable is assigned 0, if and only if I is unsatisfiable. Theorem 26 and Theorem 27 then gives co-NP-hardness for every other base Γ of I_0 for free. The case when $\langle\Gamma\rangle = \mathsf{I}_1$ is entirely analogous and we therefore omit it.

Last, assume that $\langle\Gamma\rangle = \mathsf{II}_2$. Let UNIQUE-$k$-SAT denote the U-SAT problem restricted to constraints formed by k-ary clauses, and recall that U-SAT is US-complete. Following the succinct upp-definition provided in Example 9 we may then conclude that UNIQUE-k-SAT is also US-complete for every $k \geq 3$. Since each k-ary clause can be represented by a k-ary relation, Theorem 26 and Theorem 27 then shows US-completeness for every base Γ of II_2. $\square$

A complexity classification akin to Theorem 28 is useful since it clearly separates tractable from intractable cases. However, in the last decade, a significant amount of research has been devoted to better understanding the "fine-grained" complexity of intractable problems, with a particular focus on ruling out algorithms running in $O(c^{|V|})$ time for every $c > 1$, so-called *subexponential time.* This line of research originates from Impagliazzo et al. [13] who conjectured that 3-SAT is not solvable in subexponential time; a conjecture known as the *exponential-time hypothesis* (ETH). Lower bounds for U-SAT(Γ) can then be proven using the ETH and the results from Section 3.

Theorem 29. *Let Γ be a Boolean constraint language such that* U-SAT(Γ) *is US-complete or co-NP-complete. Then* U-SAT(Γ) *is not solvable in subexponential time, unless the ETH is false.*

Proof. We begin with the case when $\langle\Gamma\rangle = \mathsf{II}_0$ or $\langle\Gamma\rangle = \mathsf{II}_1$. First, observe that U-SAT(Γ) is the complement of determining whether there exists a non-constant model. The latter problem, in turn, can be seen as a special case of the problem of determining if there exists a surjective model, and is in the literature referred to as SUR-SAT(Γ) or SAT$^*(\Gamma)$ [10]. It is furthermore known that if SUR-SAT(Γ) for $\langle\Gamma\rangle \in \{\mathsf{II}_0, \mathsf{II}_1\}$ is solvable in subexponential time then the ETH is false [17][Section 5].

Second, assume that $\langle\Gamma\rangle = \mathsf{II}_2$. Assume that U-SAT($\Gamma$) is solvable in subexponential time. Results from Jonsson et al. [16] then imply that U-SAT($R^{\neq\neq\neq 01}_{1/3}$), where

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

is solvable in subexponential time, too. It is furthermore known that the ETH is false if and only if SAT($R^{\neq\neq\neq 01}_{1/3}$) restricted to instances containing at most $2|V|$ constraints, is solvable in subexponential time [16]. Hence, it suffices to show that the original assumption implies that SAT($R^{\neq\neq\neq 01}_{1/3}$), restricted to instances with at most $2|V|$ constraints, is solvable in subexponential time, in order to contradict the ETH. Define the 9-ary relation $R_\vee$ as $R_\vee = \{(x_1, \ldots, x_8, b) \mid (x_1, \ldots, x_8) \in R^{\neq\neq\neq 01}_{1/3}, b \in \{0,1\}\} \cup \{(x_1, \ldots, x_8, 1) \mid (x_1, \ldots, x_8) \notin R^{\neq\neq\neq 01}_{1/3}\}$. Let

$$R_\vee(x_1, \ldots, x_8, x_9) \equiv \exists! y_1, \ldots, y_D \colon \varphi_{R_\vee}(x_1, \ldots, x_8, x_9, y_1, \ldots, y_D)$$

be a upp-definition of $R_\vee$ over $R^{\neq\neq\neq 01}_{1/3}$; this is possible due to Theorem 26. Similarly, let $(x_1 \to x_2) \equiv \exists! y_1, \ldots, y_E \colon \varphi_{x_1 \to x_2}(x_1, x_2, y_1, \ldots, y_E)$ be a upp-definition of $(x_1 \to x_2)$ over $R^{\neq\neq\neq 01}_{1/3}$. The reduction then proceeds as follows. Let $I = (V, C)$ be an instance of SAT($R^{\neq\neq\neq 01}_{1/3}$) where $|C| \leq 2|V|$. We introduce one fresh variable x together with the constraints $\bigwedge_{i=1}^{|V|} \varphi_{\to}(x, x_i, y_i^1, \ldots, y_i^E)$, where $y_1^1, \ldots, y_1^E, \ldots, y_n^1, \ldots, y_n^E$ are fresh variables. For each constraint $c_i = R^{\neq\neq\neq 01}_{1/3}(x_i^1, \ldots, x_i^8)$ we then replace it by $\varphi_{R_\vee}(x_i^1, \ldots, x_i^8, x, z_i^1, \ldots, z_i^D)$, where $z_i^1 \ldots, z_i^D$ are fresh variables.

Let $I' = (V', C')$ be the resulting instance of U-SAT($R^{\neq\neq\neq 01}_{1/3}$), where $V' = V \cup \{x\} \cup Y$ such that Y consists of the variables introduced when replacing constraints in C by their upp-definitions over $R^{\neq\neq\neq 01}_{1/3}$. We claim that I is unsatisfiable if and only if I' admits a unique model. First assume that I is unsatisfiable. In this case any model f of I' must satisfy $f(x) = 1$. However, due to the constraints $\bigwedge_{i=1}^{|V|} \varphi_{\to}(x, x_i, y_i^1, \ldots, y_i^E)$ this also implies that $f(x_i) = 1$ for each $x_i \in V$. But since each $y \in Y$ is determined by a set of variables in V, it follows that I' has a unique model. For the other direction, assume that I' has a unique model f. Assume first that $f(x) = 1$. Then the constraints $\bigwedge_{i=1}^{|V|} \varphi_{\to}(x, x_i, y_i^1, \ldots, y_i^E)$ force $f(x_i) = 1$ for each $x_i \in V$, and it is trivial to verify that f does not satisfy I, and that the existence a model of I would contradict the uniqueness of f. Second, assume that $f(x) = 0$. Define $g \colon V' \to \{0,1\}$ such that $g(x) = 1$, $g(x_i) = 1$ for each $x_i \in V'$, and $g(y_i)$ for $y_i \in Y$ according to the value prescribed by g in the constraint containing y_i. This is possible since each variable in Y is determined by $V \cup \{x\}$. However, then g is also a model of I', contradicting the uniqueness assumption of f.

The above construction can clearly be carried out in polynomial time with respect to $|V|$ and $|C|$. For the time complexity, the constraints $\bigwedge_{i=1}^{|V|} \varphi_{\to}(x, x_i, y_i^1, \ldots, y_i^E)$ require $|V| \cdot E$ fresh variables, and the constraints $\bigwedge_{i=1}^{|C|} \varphi_{R_\vee}(x_i^1, \ldots, x_i^8, x, z_i^1, \ldots, z_i^D)$ require $|C| \cdot D$ fresh variables, which is bounded by $2|V|$. Hence, $|V'|$ is bounded by $|V| + |V|E + 2|V|D$, and since D and E are both constant it follows that SAT($R^{\neq\neq\neq 01}_{1/3}$) is solvable in subexponential time, which contradicts the ETH. $\square$

Using our algebraic framework, hardness results can effortlessly be proven for the CSP generalisation of U-SAT, i.e., the problem U-CSP(Γ) of answering yes if and only if the given instance of CSP(Γ) admits a unique model.

Theorem 30. *Let Γ be a constraint language over a finite domain D. If $\langle\Gamma\rangle = \mathsf{REL}_D$ then* U-CSP(Γ) *is US-complete, and if* $\mathrm{Pol}(\Gamma) = [\{f\}]$ *for a constant operation f, then* U-CSP(Γ) *is co-NP-complete.*

Proof. First, assume that $\langle\Gamma\rangle = \mathsf{REL}_D$. Inclusion in US follows directly from the definition of U-CSP(Γ). To show hardness we take a Boolean Δ such that U-SAT(Δ) is US-complete, and since $\langle \Delta \rangle_{\exists!} \subseteq \langle \Gamma \rangle_{\exists!}$ by Theorem 13, we can perform a standard gadget reduction from U-SAT(Δ) by replacing constraints by their upp-definitions over Γ.

Second, assume that $\mathrm{Pol}(\Gamma) = [\{f\}]$ for a constant operation f, and let $\{d\}$ be the image of f. Then a no-instance of U-CSP(Γ) can be verified by any model distinct from the constant model where each variable is assigned d, implying that U-CSP(Γ) is included in co-NP. The hardness argument is similar to Theorem 28 and we only provide a sketch of the construction. Let $d_1, \ldots, d_k$ be an enumeration of D and let $R = \{(0,0,1),(0,1,0),(1,0,0)\} \times \{(d_1, \ldots, d_k)\}$. Then the problem of checking whether a CSP(R) instance does not admit a model, UNCSP(R), is co-NP-complete, and we then reduce UNCSP(R) to U-CSP($R \cup \{(d, \ldots, d)\}$) by replacing each constraint by the corresponding constraint over $R \cup \{(d, \ldots, d)\}$. Since $R \cup \{(d, \ldots, d)\} \in \langle \Gamma \rangle_{\exists!}$ by Theorem 13, co-NP-hardness carries over to U-CSP(Γ). ☐

5 Concluding Remarks and Future Research

We have studied unique existential quantification in pp-definitions, with a particular focus on finding constraint languages where existential quantification and unique existential quantification coincide. In general, this question appears highly challenging, but we have managed to find several broad classes of languages where this is true, and established a complete dichotomy theorem in the Boolean domain. We also demonstrated that upp-definitions can be applied to obtain complexity theorems for problems in a more systematic manner than what has earlier been possible. Many interesting open question hinge on the possibility of finding an algebraic characterisation of upp-closed sets of relations. For example, it would be interesting to determine the cardinality of the set $\{\langle \Gamma \rangle_{\exists!} \mid \Gamma \subseteq \mathrm{II}_2\}$, and hopefully describe all such upp-closed sets. By our classification theorem it suffices to investigate the Boolean co-clones that are not $\exists!$-covered, but even this question appears difficult to resolve using only relational tools. Similarly, a continued description of the $\exists!$-covered co-clones over finite domains would be greatly simplified by an algebraic characterisation. Thus, given a set of relations Γ, what is the correct notion of a "polymorphism" of a upp-definable relation over Γ? This question also has a strong practical motivation: essentially all complexity classifications for CSP related problems over non-Boolean domain require stronger algebraic tools than pp-definitions, and this is likely the case also for problems that can be studied with upp-definitions.

Another interesting topic is the following computational problem concerning upp-definability. Fix a constraint language Γ, and let R be a relation. Is it the case that R is upp-definable over Γ? The corresponding problem for pp-definitions is tractable for Boolean constraint languages Γ [9] while the corresponding problem for qfpp-definitions is co-NP-complete [19, 22]. Note that if $\langle \Gamma \rangle$ is $\exists!$-covered (which can be checked in polynomial time) then $R \in \langle \Gamma \rangle_{\exists!}$ can be answered by checking whether $R \in \langle \Gamma \rangle$. Thus, only the co-clones that are not $\exists!$-covered would need to be investigated in greater detail.

Last, it is worth remarking that our notion of uniqueness quantification in pp-definitions is not the only one possible. Assume that we in $\exists! x_i \colon R(x_1, \ldots, x_i, \ldots, x_n)$ over a domain D do not require that x_i is determined by $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ but instead simply obtain the relation $\{(d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_n) \mid \exists! d_i \in D \text{ such that } (d_1, \ldots, d_{i-1}, d_i, d_{i+1}, \ldots, d_n) \in R)\}$. This notion of unique existential quantification is in general *not* comparable to existential quantification, since if we e.g. let $R = \{(0,0),(0,1),(1,0)\}$ then $T(x) \equiv \exists! y \colon R(y,x)$ even though $T \notin \langle R \rangle$, i.e., is not even pp-definable by R (where $T = \{(1)\}$). Thus, it would be interesting to determine the resulting closed classes of relations and see in which respect they differ from the ordinary co-clone lattice.

Acknowledgements

We thank Andrei Bulatov for helpful discussions concerning the topic of the paper.

References

[1] L. Barto, A. Krokhin, and R. Willard. Polymorphisms, and How to Use Them. In Andrei Krokhin and Stanislav Zivny, 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.

[2] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1):80 – 88, 1982.

[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] A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS-2017)*. IEEE Computer Society, 2017.

[7] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1 – 12, 1996.

[8] N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of Boolean constraint satisfaction problems*. SIAM Monographs on Discrete Mathematics and Applications, 2001.

[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] V. Dalmau. *Computational complexity of problems over generalized formulas*. PhD thesis, Department of Computer Science, Polytechnical University of Catalonia, 2000.

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

[13] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.

[14] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[15] P. Jeavons, D. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings of the First International Conference in Principles and Practice of Constraint Programming (CP-1995)*, pages 276–291, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[16] 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.

[17] P. Jonsson, V. Lagerkvist, J. Schmidt, and H. Uppman. Relating the time complexity of optimization problems in light of the exponential-time hypothesis. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS-14)*, pages 408–419, Berlin, Heidelberg, 2014. Springer-Verlag.

[18] L. Juban. Dichotomy theorem for the generalized unique satisfiability problem. In *Proceedings of the 12th International Symposium of Fundamentals of Computation Theory (FCT-1999)*, volume 1684 of *Lecture Notes in Computer Science*, pages 327–337. Springer, 1999.

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

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

[21] V. Lagerkvist. *Strong Partial Clones and the Complexity of Constraint Satisfaction Problems: Limitations and Applications*. PhD thesis, Linköping University, The Institute of Technology, 2016.

[22] V. Lagerkvist and B. Roy. A dichotomy theorem for the inverse satisfiability problem. In *Proceedings of the 37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS-2017)*, volume 93, pages 39:39–39:14, 2018.

[23] 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.

[24] G. Nordh and B. Zanuttini. Frozen Boolean partial co-clones. In *Proceedings of the 39th International Symposium on Multiple-Valued Logic (ISMVL-2009)*, pages 120 –125, 2009.

[25] C.H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244 – 259, 1984.

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

[27] S. Reith and K. W. Wagner. The complexity of problems defined by Boolean circuits. In *Proceedings International Conference Mathematical Foundation of Informatics (MFI-1999)*, pages 25–28, 1999.

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

[29] 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.

[30] 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.

[31] D. Zhuk. The proof of CSP dichotomy conjecture. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS-2017)*. IEEE Computer Society, 2017.