Recognizing Frozen Variables in Constraint Satisfaction Problems

Peter Jonsson
Department of Computer and Information Science
Linköpings Universitet
SE-581 83 Linköping, Sweden
email: peter.jonsson@ida.liu.se

Andrei Krokhin
Department of Computer Science
University of Durham, UK
email: andrei.krokhin@durham.ac.uk

April 28, 2005

Abstract

In constraint satisfaction problems over finite domains, some variables can be *frozen*, that is, they take the same value in all possible solutions. We study the complexity of the problem of recognizing frozen variables with restricted sets of constraint relations allowed in the instances. We show that the complexity of such problems is determined by certain algebraic properties of these relations. Under the assumption that **NP** $\neq$ **coNP** (and consequently **PTIME** $\neq$ **NP**), we characterize *all* tractable problems, and describe large classes of **NP**-complete, **coNP**-complete, and **DP**-complete problems. As an application of these results, we *completely* classify the complexity of the problem in two cases: (1) with domain size 2; and (2) when all unary relations are present. We also give a rough classification for domain size 3.

Keywords: constraint satisfaction, frozen variable, computational complexity, polymorphism

1 Introduction

The constraint satisfaction problem (CSP) is a powerful general framework in which a variety of combinatorial problems can be expressed [11, 30]. The aim in a constraint satisfaction problem is to find an assignment of values to the variables subject to specified constraints. This framework is used across a variety of research areas in artificial intelligence, including planning [26], scheduling [46], and image processing [32], and in computer science, including combinatorial optimization [11, 20], database theory [18, 28], and complexity theory [13, 16], one of the most important applications being constraint programming [30].

In constraint satisfaction problems over finite domains, some variables can be *frozen*, that is, they take the same value in all possible solutions. We note that frozen variables and closely related objects such as spines [2], frozen pairs [12], backbones [17, 31, 43], and unary prime implicates [36] appear frequently in the CSP literature and that they are actively used in, for instance, the study of phase transition phenomena. It is well-known that CSP can be viewed as the homomorphism problem for relational structures [19, 28]. The version of CSP we study is known to be equivalent to the non-uniform homomorphism problem [28]: we fix a relational structure $\mathbf{B}$ and ask whether a given structure $\mathbf{A}$ admits a homomorphism to $\mathbf{B}$. In this setting, the frozen variable problem can be expressed as follows: in the input structure $\mathbf{A}$ we choose some elements and ask whether $\mathbf{A}$ homomorphically maps to $\mathbf{B}$ and, in addition, all homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ agree on each chosen element.

The problem of recognizing frozen variables is closely related to the well-known UNIQUE SAT problem (i.e. recognizing SAT-instances with a unique satisfying assignment). This is an intriguing problem, as it is one of the few (natural) versions of the propositional satisfiability problem that are not known to be complete for any standard complexity class [25, 33]. It is easy to see that the problem of recognizing frozen variables is a generalization of UNIQUE SAT, as the number of possible values is not restricted to two, clauses are replaced by arbitrary predicates, and the uniqueness requirement may be applied to some, but not necessarily all, variables.

Another related problem was considered in connection with databases that support statistical queries. For instance, consider a relation with attributes $(\mathsf{name}, \mathsf{age}, \mathsf{salary})$ supporting statistical queries of the form 'give me the sum of salaries of all individuals whose age satisifes a certain condition'. If we assume that the projection $(\mathsf{name}, \mathsf{age})$ is publicly available, what measures suffice to protect the confidentiality of the salary information? This is the *statistical database security* problem [1] and one approach to solve this problem is to *audit* the statistical queries in order to determine when enough information has been given out so that compromise becomes possible. Kleinberg et al. [27] have studied the complexity of this problem.

By slightly generalising their formalisation to attributes having an arbitrary number of values, we have the follwing problem which we call AUDIT:

INSTANCE: A set $\{x_1, \ldots, x_n\}$ of variables taking their values from the set $D = \{0, 1, \ldots, K\}$, a family of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$ of $\{1, \ldots, n\}$, and m integers $b_1, \ldots, b_m$.
QUESTION: Is there an $i \leq n$ such that in all 0-1-...-K solutions of the system of equations $\sum_{i \in S_j} x_i = b_j$, $j = 1, \ldots, m$, the variable x_i has the same value.

For Boolean domains (where $D = \{0, 1\}$), Kleinberg et al. show that this problem is **coNP**-complete. Clearly, this problem is closely related to the frozen variable problem where the constraints are specified by the equations given above. We would like to point out one subtle difference between the two problems: in the frozen variable problem, an instance is considered to be a 'no'-instance if it has no solution; in the AUDIT problem, an instance is considered to be a 'yes'-instance if the equation system has no solution. However, it is easy to modify our results and use them for showing that AUDIT is **coNP**-complete for all finite domains with more than two elements; it also gives an alternative proof of Kleinberg et al.'s results. Furthermore, our results are applicable to a wider range of statistical queries and not only summation queries.

We will now begin our investigation of the complexity of the frozen variable recognition problem. Note that it is not clear *a priori* whether there is any dependence between efficient deciding of satisfiability and efficient recogniton of frozen variables in CSPs. For example, CSPs such as NOT-ALL-EQUAL-SAT and GRAPH k-COLORING, $k \geq 3$, (see Examples 2.3 and 2.4) are **NP**-complete, but the frozen variable problem for them is trivial because no variable can be frozen in these problems, due to certain symmetries.

Constraints are usually specified by relations, or predicates, and the standard constraint satisfaction problem can therefore be parameterized by restricting the set of allowed relations which can be used as constraints. The problem of determining (up to complete classification) the complexity of the CSP and its many variants for all possible parameter sets has attracted much attention (see, e.g., [5, 6, 11, 16]). For the Boolean (i.e., two-valued) case, the complexity of the standard constraint satisfaction problem has been studied from the above perspective [42], as well as a number of related problems (see [11, 25, 29] for a selection of those); such problems are sometimes referred to as "generalized satisfiability problems".

It is widely acknowledged that, compared to the Boolean case, one needs more advanced tools to make progress with non-Boolean constraint satisfaction problems. Such tools based on algebra, logic, and graph theory were developed in [5, 6, 8, 9, 13, 16, 18, 22, 23, 28]. The algebraic method [5, 6, 8, 9, 22, 23], which proved to be quite powerful, builds on the fact that one can extract much information about the structure and the complexity of restricted constraint satisfaction problems from knowing certain operations, called polymorphisms, connected with the constraint relations. More exactly, polymorphisms provide a covenient 'dual' language for describing relations and, more importantly, they allow one to show that one constraint can be simulated by other constraints without giving an explicit construction.

In this paper we apply the algebraic method to study the complexity of the parameterized problems of recognizing frozen variables. We characterize *all* tractable problems (i.e., those in **PTIME**) and show that in this case the unique values for all frozen variables in an instance, if there are any, can be found efficiently, and we also present large classes of problems that are **NP**-complete, **coNP**-complete, and **DP**-complete. As an application of these results, we completely classify the complexity of the problem with domain size 2, and with arbitrary domain, but in the presence of all unary relations, and also give a rough classification for domain size 3. We observe that CSP problems containing all unary relations is a generalization of the well-studied List Homomorphism problems for graphs [6, 14, 15].

The paper is organized as follows. In Section 2, we give basic definitions and discuss the algebraic method that will be used in the paper. In Section 3, we show that the algebraic technique is applicable to the problems we study and that the complexity of these problems depends in a certain way on the complexity of the basic satisfiability problem for given constraints. Section 4 is devoted to a characterization of the tractable cases of the problem, and it also contains a sufficient condition for a problem to be **coNP**-complete. In Section 5, we study **NP**-complete and **DP**-complete cases of the problem. In Section 6, we give two complete classifications of complexity for two important special cases of the problem: one is the Boolean case, and the other is when all unary relations are available. Section 7 contains some conclusions about the work we have done.

2 Preliminaries

This section is divided into two parts: we present the computational problems in the first part and provide some basics of the algebraic techniques in the second. Throughout the paper we use the standard correspondence between predicates and relations: a relation consists of all tuples of values for which the corresponding predicate holds. We will use the same symbol for a predicate and its corresponding relation, since the meaning will always be clear from the context. We will use $R_D^{(m)}$ to denote the set of all m-ary relations (or predicates) over a *fixed finite* set D, and R_D to denote the set $\bigcup_{m=1}^{\infty} R_D^{(m)}$. Note that unary relations on D are simply the subsets of D. Throughout this paper we assume that $\mathbf{NP} \neq \mathbf{coNP}$ (and consequently

PTIME $\neq$ **NP**).

2.1 Computational problems

We begin by defining the constraint satisfaction problem.

Definition 2.1 *A* constraint language *over* D *is an arbitrary subset of* R_D. *The* constraint satisfaction problem *over the constraint language* $\Gamma \subseteq R_D$, *denoted* $\mathrm{CSP}(\Gamma)$, *is the decision problem with instance* $I = (V, D, \mathcal{C})$, *where*

- V *is a finite set of variables,*
- D *is a set of values (sometimes called a domain), and*
- $\mathcal{C}$ *is a set of constraints* $\{C_1, \ldots, C_q\}$,

  in which each constraint C_i *is a pair* (s_i, ϱ_i) *with* s_i *a list of variables of length* m_i, *called* the constraint scope, *and* ϱ_i *an* m_i*-ary relation over the set* D, *belonging to* Γ, *called* the constraint relation.

The question is whether there exists a solution *to* I, *that is, a function* $\varphi : V \to D$ *such that, for each constraint in* $\mathcal{C}$, *the image of the constraint scope is a member of the constraint relation. If* I *has a solution then we also say that* I *is* satisfiable.

The size of a problem instance is the length of the encoding of all tuples in all constraints.

Definition 2.2 *We say that* $\mathrm{CSP}(\Gamma)$ *is tractable if, for every finite* $\Gamma' \subseteq \Gamma$, $\mathrm{CSP}(\Gamma')$ *is in* **PTIME**. *Similarly, we say that* $\mathrm{CSP}(\Gamma)$ *is* **NP**-, *or* **coNP**-, *or* **DP**-*complete if the problem is in the corresponding complexity class and, for some finite* $\Gamma' \subseteq \Gamma$, $\mathrm{CSP}(\Gamma')$ *has the corresponding completeness property.*

Example 2.3 *Let* N *and* N' *be the following ternary relations on* $\{0,1\}$:

$$N = \{(1,0,0),(0,1,0),(0,0,1)\}, \quad N' = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}.$$

It is easy to see that the 1-IN-3-SAT *and the* NOT-ALL-EQUAL-SAT *problems (as defined in [42]) can be expressed as* $\mathrm{CSP}(\{N\})$ *and* $\mathrm{CSP}(\{N'\})$, *respectively. Both problems are known to be* **NP**-*complete [42].*

Example 2.4 *Let* $\neq_D$ *be the binary disequality relation on any finite* D. *Then* $\mathrm{CSP}(\neq_D)$ *is exactly the* GRAPH $|D|$-COLORING *problem. It is known to be tractable if* $|D| = 2$ *and* **NP**-*complete otherwise [33].*

Many other combinatorial problems, including HOMOMORPHISM, CLIQUE, and GRAPH REACHABILITY problems, expressed as CSPs, can be found in [22].

We now define the main objects of our study in this paper.

Definition 2.5 *Let* $I = (V, D, \mathcal{C})$ *be an instance of* CSP(Γ)*, and* $x \in V$*. Then* x *is said to be* frozen *in* I *if* $|\{\varphi(x) \mid \varphi$ *is a solution to* $I\}| = 1$*. We say that* $V' \subseteq V$ *is frozen in* I *if every* $x \in V'$ *is frozen in* I*.*

Definition 2.6 *Let* $\Gamma \subseteq R_D$*. An* instance *of* FV-CSP(Γ) *is a pair* (I, V') *where* I *is an instance of* CSP(Γ)*, with a set* V *of variables, and* V' *is a non-empty subset of* V*. The* question *is whether* V' *is frozen in* I*.*

Example 2.7 *The conjunctive-query evaluation problem [28] in database theory is to find the predicate (or decide whether it is non-empty) on variables* $y_1, \ldots, y_m$ *given by a formula of the form* $(\exists x_1) \ldots (\exists x_n)\ \mathcal{C}$ *where* $\mathcal{C} = \varrho_1(s_1) \wedge \ldots \wedge \varrho_q(s_q)$*, and* $x_1, \ldots, x_n, y_1, \ldots, y_m$ *are the variables used in* $\mathcal{C}$*. It is easy to see that the problem of deciding whether a conjunctive query has a unique answer is precisely the problem of deciding whether* $\{y_1, \ldots, y_m\}$ *is frozen in* $\mathcal{C}$*.*

Example 2.8 *If we restrict* FV-CSP($R_{\{0,1\}}$) *to instances with* $V' = V$ *then we obtain the generalized* UNIQUE SAT *problem [25].*

The UNIQUE SAT problem is known to belong to **DP** (recall that **DP** is the complexity class $\{L \cap L' \mid L \in \mathbf{NP}, L' \in \mathbf{coNP}\}$ [33] and that this class contains both **NP** and **coNP**). Moreover, it is not known to belong to any weaker complexity class and it is known to be **DP**-complete only under randomized reductions [45]. A number of problems including MINIMAL UNSAT, TSP FACETS, CRITICAL CLIQUE, MAXIMAL NON-HAMILTONIAN GRAPH, and MINIMAL 3-COLORABILITY are known to be **DP**-complete [34, 35].

The ultimate goal of this investigation is to determine the complexity of FV-CSP(Γ) for all possible Γ. We start with the following basic fact that sets an upper bound for the complexity of this problem.

Proposition 2.9 FV-CSP(Γ) *is in* **DP** *for every constraint language* Γ*.*

Proof. Let $I = (V, D, \mathcal{C})$ be an instance of CSP(Γ), and $V' \subseteq V$. To prove that V' is frozen in I, we need (1) check that I has a solution (**NP**-part), and (2), check that there do not exist two solutions φ_1, φ_2 such that $\varphi_1(v) \neq \varphi_2(v)$ for some $v \in V'$ (**coNP**-part). ■

2.2 Algebraic techniques

We will now consider *operations* on the set of values. We use $O_D^{(n)}$ to denote the set of all n-ary operations on a set D (that is, the set of mappings $f: D^n \to D$), and O_D to denote the set $\bigcup_{n=1}^{\infty} O_D^{(n)}$.

Any operation on D can be extended in a standard way to an operation on tuples over D, as follows. For any operation $f \in O_D^{(n)}$, and any collection of tuples $\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n \in D^m$, where $\vec{a}_i = (\vec{a}_i(1), \ldots, \vec{a}_i(m))$, $i = 1, \ldots, n$, define

$$f(\vec{a}_1, \ldots, \vec{a}_n) = (\, f(\vec{a}_1(1), \ldots, \vec{a}_n(1)), \ldots, f(\vec{a}_1(m), \ldots, \vec{a}_n(m))\,).$$

Definition 2.10 *For any relation $\varrho \in R_D^{(m)}$, and any operation $f \in O_D^{(n)}$, if $f(\vec{a}_1, \ldots, \vec{a}_n) \in \varrho$ for all $\vec{a}_1, \ldots, \vec{a}_n \in \varrho$, then ϱ is said to be* invariant *under f, and f is called a* polymorphism *of ϱ.*

Note that unary polymorphisms of a relation can be seen as a generalization of the notion of an endomorphism (that is, homomorphism into itself) for graphs; indeed, for graphs, the two notions coincide.

The set of all relations that are invariant under each operation from some set $C \subseteq O_D$ will be denoted $\mathsf{Inv}(C)$. The set of all operations that are polymorphisms of every relation from some set $\Gamma \subseteq R_D$ will be denoted $\mathsf{Pol}(\Gamma)$. By $\mathsf{Pol}_n(\Gamma)$ we will denote the set of all n-ary members of $\mathsf{Pol}(\Gamma)$. We remark that the operators Inv and Pol form a Galois correspondence between R_D and O_D (see survey [38] or Proposition 1.1.14 in [39]). A basic introduction to this correspondence can be found in [37, 38], and a comprehensive study in [39].

It is easy to see that CSP(Γ) can be expressed as a logical problem as follows: is it true that a first-order formula $\varrho_1(s_1) \wedge \ldots \wedge \varrho_q(s_q)$, where each ϱ_i is an atomic formula involving a predicate from Γ, is satisfiable?

Definition 2.11 *For any set $\Gamma \subseteq R_D$ the set $\langle \Gamma \rangle$ consists of all predicates that can be expressed using*

1. *predicates from $\Gamma \cup \{=_D\}$,*
2. *conjunction,*
3. *existential quantification.*

A relation belongs to $\langle \Gamma \rangle$ if and only if it can be represented as the projection of the set of all solutions to some CSP(Γ)-instance onto some subset of variables [24]. Intuitively, constraints using relations from $\langle \Gamma \rangle$ are exactly those which can be 'simulated' by constraints using relations in Γ. In fact, $\langle \Gamma \rangle$ can be characterized in a number of ways [39], and one of them is most important for our purposes.

Theorem 2.12 ([39]) *For every set* $\Gamma \subseteq R_D$*,* $\langle\Gamma\rangle = \mathsf{Inv}(\mathsf{Pol}(\Gamma))$*.*

Theorem 2.12 is the corner-stone of the algebraic method, since it shows that the expressive power of constraints is determined by polymorphisms. In particular, in order to show that a relation ϱ can be expressed by relations in Γ, one does not have to give an explicit construction, but instead one can show that ϱ is invariant under all polymorphisms of Γ, which often turns out to be significantly easier. Moreover, sets of operations of the form $\mathsf{Pol}(\Gamma)$ are known as *clones*, that is, they are precisely the sets C of operations on D with the following properties:

- C contains all *projections*, i.e. all operations satisfying $f(x_1, \ldots, x_n) = x_i$ for some $1 \leq i \leq n$ and all $x_1, \ldots, x_n \in D$;
- C is closed under *superposition*, that is, for any n-ary $f \in C$ and any m-ary operations $g_1, \ldots, g_n \in C$, the operation

  $$f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))$$

  also belongs to C.

Note that there exist several equivalent definitions of a clone; this one follows [44]. Clones are well-studied objetcs in algebra (see, e.g., [39, 44] for more information including the just mentioned equivalence), and they are all known for the case $|D| = 2$ [40]. In this paper we will use the following result from [40] (see also [41], Chapter 1.4 [37], or Corollary 1.14 [44]) .

Proposition 2.13 *Let C be a clone on $\{0,1\}$. Either C consists of all projections (and then $\mathsf{Inv}(C) = R_{\{0,1\}}$), or else C contains at least one of the following 7 operations:*

(a) the constant operation 0,

(b) the constant operation 1,

(c) the negation operation $\neg x$,

(d) the disjunction operation $x \vee y$,

(e) the conjunction operation $x \wedge y$,

(f) the majority operation $(x \vee y) \wedge (x \vee z) \wedge (y \vee z)$,

(g) the affine operation $x - y + z \pmod 2$.

Example 2.14 *Reconsider the relation N from Example 2.3. It is easy to check that none of the 7 operations from Proposition 2.13 is a polymorphism of N. Since $\mathsf{Pol}(\{N\})$ is a clone, it follows from Proposition 2.13 that it consists of all projections. Moreover, it is easy to verify that any relation is invariant under every projection, and then, by Theorem 2.12, we have $\langle\{N\}\rangle = R_{\{0,1\}}$.*

Example 2.14 illustrates how Theorem 2.12 allows one to make use of known results from algebraic clone theory. Moreover, using Theorem 2.12, the following result was obtained in [22].

Theorem 2.15 ([22]) *Let Γ_1 and Γ_2 be sets of predicates over a finite set, such that Γ_1 is finite. If $\mathsf{Pol}(\Gamma_2) \subseteq \mathsf{Pol}(\Gamma_1)$ then $\mathrm{CSP}(\Gamma_1)$ is polynomial-time reducible to $\mathrm{CSP}(\Gamma_2)$.*

This result shows that finite sets of relations with the same polymorphisms give rise to constraint satisfaction problems which are mutually reducible. In other words, the complexity of $\mathrm{CSP}(\Gamma)$ is determined by the polymorphisms of Γ.

A number of results on the complexity of constraint satisfaction problems have been obtained via this approach (e.g., [3, 5, 4, 6, 7, 8, 9, 10, 13, 22, 23]). For example, it is shown in [22] that Schaefer's Dichotomy Theorem [42], when appropriately re-stated, easily follows from Theorem 2.15 and well-known algebraic results [40].

Theorem 2.16 ([42]) *For any set $\Gamma \subseteq R_{\{0,1\}}$, $\mathrm{CSP}(\Gamma)$ is tractable when $\mathsf{Pol}(\Gamma)$ contains at least one of the operations (a)-(b) or (d)-(g) from Proposition 2.13. In all other cases $\mathrm{CSP}(\Gamma)$ is **NP**-complete.*

3 Reduction and separation

In this section we prove that the complexity of $\mathrm{FV\text{-}CSP}(\Gamma)$ is determined by the polymorphisms of Γ, and hence the algebraic technique is applicable. We also show how the complexity of $\mathrm{FV\text{-}CSP}(\Gamma)$ strongly depends on the set $\mathsf{Pol}_1(\Gamma)$ of unary polymorphisms of Γ and on the complexity of $\mathrm{CSP}(\Gamma)$.

Lemma 3.1 *Let $\Gamma \subseteq R_D$ and $\varrho \in \langle\Gamma\rangle$ for some $\varrho \in R_D$. Then, the problems $\mathrm{FV\text{-}CSP}(\Gamma \cup \{\varrho\})$ and $\mathrm{FV\text{-}CSP}(\Gamma)$ are polynomial-time equivalent.*

Proof. By the remark after Definition 2.11, each occurence of ϱ in every instance I of $\mathrm{CSP}(\Gamma \cup \{\varrho\})$ can be replaced by the corresponding collection of constraints involving only relations from $\Gamma \cup \{=_D\}$ (with possible renaming of variables to avoid name clashes). The equality constraint can then be removed by identifying variables. It is easy to see that transforming an arbitrary instance (I, V') of $\mathrm{FV\text{-}CSP}(\Gamma \cup \{\varrho\})$ in the same way and keeping V' the same gives us a polynomial-time reduction from $\mathrm{FV\text{-}CSP}(\Gamma \cup \{\varrho\})$ to $\mathrm{FV\text{-}CSP}(\Gamma)$. The reduction in the other direction is trivial. ∎

Theorem 3.2 *Arbitrarily choose $\Gamma_1, \Gamma_2 \subseteq R_D$ and assume that Γ_1 is finite. If $\mathsf{Pol}(\Gamma_2) \subseteq \mathsf{Pol}(\Gamma_1)$ then $\mathrm{FV\text{-}CSP}(\Gamma_1)$ is polynomial-time reducible to $\mathrm{FV\text{-}CSP}(\Gamma_2)$.*

Proof. If $\mathsf{Pol}(\Gamma_2) \subseteq \mathsf{Pol}(\Gamma_1)$, then $\mathsf{Inv}(\mathsf{Pol}(\Gamma_2)) \supseteq \mathsf{Inv}(\mathsf{Pol}(\Gamma_1))$ since Inv is antimonotone (i.e. inclusion-reversing). This implies that $\langle \Gamma_1 \rangle \subseteq \langle \Gamma_2 \rangle$ by Theorem 2.12 so the theorem is proved by applying Lemma 3.1 a finite number of times. ∎

Similarly to Theorem 2.15, Theorem 3.2 shows that the computational complexity of FV-CSP(Γ) is determined by the polymorphisms of Γ.

Remark 3.3 *It is easy to see that if φ is a solution to an instance I of* CSP(Γ) *then so is $f\varphi$ for every $f \in \mathsf{Pol}_1(\Gamma)$. It follows that if $\varphi(x) = a$ for some variable x in I then, for every $b \in D$ with $b = f(a)$ for some $f \in \mathsf{Pol}_1(\Gamma)$, there is another solution that maps x to b.*

This remark shows that unary polymorphisms are very important in recognizing frozen variables. For example, if $f(d) \neq d$ for some $f \in \mathsf{Pol}_1(\Gamma)$ then d cannot be the value taken by a frozen variable in an instance of CSP(Γ). In fact, the condition $\{d \in D \mid f(d) = d \text{ for all } f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$ can be shown to be equivalent to saying that no variable in any instance of CSP(Γ) can be frozen.

Proposition 3.4 *Let $\Gamma \subseteq R_D$. If*

$$\{d \in D \mid f(d) = d \textit{ for all } f \in \mathsf{Pol}_1(\Gamma)\} \neq \emptyset$$

then CSP(Γ) *is polynomial-time reducible to* FV-CSP(Γ).

Proof. Let $d \in D$ be such that $f(d) = d$ for all $f \in \mathsf{Pol}_1(\Gamma)$. Then, since $f'(x, \ldots, x) \in \mathsf{Pol}_1(\Gamma)$ for all $f' \in \mathsf{Pol}(\Gamma)$, we have $f'(d, \ldots, d) = d$ for all $f' \in \mathsf{Pol}(\Gamma)$. By Theorem 2.12, $\{d\} \in \langle \Gamma \rangle$. By Lemma 3.1, we may assume that $\{d\} \in \Gamma$. Take an arbitrary instance I of CSP(Γ) and transform it to an instance $(I', \{z\})$ of FV-CSP(Γ) as follows: introduce a new variable z and add a constraint $(z, \{d\})$. It is obvious that z is frozen in I' if and only if I is satisfiable. ∎

Proposition 3.4 answers a question, mentioned in the introduction, about the dependence between efficient deciding of satisfiability and efficient frozen variable recognition. Indeed, we now see that whenever there are frozen variables in instances of FV-CSP(Γ), this problem is not easier than CSP(Γ), that is, the problems having 'symmetries', like Not-All-Equal-Sat and Graph k-coloring, $k \geq 3$, as mentioned in the introduction, are in fact the only problems where the frozen variable problem is easier (trivial) than the satisfiability problem. Note that Proposition 3.4, though seemingly easy, would be very difficult to prove without the algebraic approach.

Theorem 3.5 *Let $\Gamma \subseteq R_D$. If $\{d \in D \mid f(d) = d \textit{ for all } f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$ then* FV-CSP(Γ) *is trivial. Otherwise,* FV-CSP(Γ) *is in* **coNP** *if* CSP(Γ) *is tractable and it is* **NP***-hard if* CSP(Γ) *is* **NP***-complete.*

Proof. If there is no $d \in D$ such that $f(d) = d$ for all $f \in \mathsf{Pol}_1(\Gamma)$ then, since $f\varphi$ is a solution to an instance whenever φ is such, no variable can be frozen in any instance. Therefore, in this case FV-CSP(Γ) is trivial.

Suppose that CSP(Γ) is tractable. If (I, V') is an instance of FV-CSP(Γ) then it can be decided in polynomial time whether I is satisfiable. If it is not then V' is not frozen in I, otherwise the problem is equivalent to the one of deciding whether I does not have two solutions that are distinct on V' which is easily seen to be in **coNP**.

The last part part of the theorem follows from Proposition 3.4. ∎

For $\varrho \in R_D$ and a unary $f \in O_D$, let $f(\varrho) = \{f(\vec{a}) \mid \vec{a} \in \varrho\}$. Also, let $f(\Gamma) = \{f(\varrho) \mid \varrho \in \Gamma\}$. Note that if $\varrho \in \Gamma$ and $f \in \mathsf{Pol}_1(\Gamma)$ then $f(\varrho) \subseteq \varrho$. It is known [9, 22] and easy to show that if $f \in \mathsf{Pol}_1(\Gamma)$ then CSP(Γ) is polynomial-time equivalent to CSP$(f(\Gamma))$. This fact is often used in the analysis of constraint satisfaction problems to reduce the set of possible values to a minimum. Unfortunately, this equivalence works in neither direction for the problem FV-CSP(Γ) as the following examples show:

Example 3.6 *Let $\Gamma \subseteq R_{\{0,1\}}$ consist of all relations that contain a tuple $(1, \ldots, 1)$. The operation f that maps both 0 and 1 to 1 is a polymorphism of Γ. The problem* FV-CSP$(f(\Gamma))$ *is trivial, but* FV-CSP(Γ) *is* **coNP**-*complete, as shown in Section 6.*

Example 3.7 *Let $D = \{0, 1, 2, 3\}$ and let $f : D \to \{0, 1\}$ be such that $f(0) = f(2) = 0$ and $f(1) = f(3) = 1$. Take $\Gamma \subseteq R_{\{0,1\}}$ as in Example 3.6 and let $\Gamma' \subseteq R_D$ be the set $\{f^{-1}(\varrho) \mid \varrho \in \Gamma\}$ where $\vec{a} \in f^{-1}(\varrho)$ if and only if $f(\vec{a}) \in \varrho$. No instance of* FV-CSP(Γ') *has frozen variables because the constraints in Γ' cannot distinguish 0 and 2, and 1 and 3. So* FV-CSP(Γ') *is trivial. It is obvious that $f \in \mathsf{Pol}_1(\Gamma')$ and $f(\Gamma') = \Gamma$, and, as shown in Section 6,* FV-CSP(Γ) *is* **coNP**-*complete.*

4 Tractable and coNP-complete problems

In this section we *completely* characterize *all* tractable problems FV-CSP(Γ) and give examples of **coNP**-complete problems. To state our theorem we need to introduce some notation.

Let $\sqsubseteq$ denote the quasi-order on D defined by the following rule: $a \sqsubseteq b$ if and only if $f(a) = b$ for some $f \in \mathsf{Pol}_1(\Gamma)$. It is well known and easy to show that the relation θ, such that $a \mathrel{\theta} b$ if and only if $a \sqsubseteq b$ and $b \sqsubseteq a$, is an equivalence relation on D. Let $[a]$ denote the θ-class containing a. It is also well known and easy to show that the relation $\leq$, on the set of all θ-classes, such that $[a] \leq [b]$ if and only if $a \sqsubseteq b$, is well-defined and is a partial order.

Let P denote the corresponding poset. We will often omit θ and call the elements of P classes. The intuition behind the poset P is simple: if, in some instance, a variable can take some value a in a solution then by Remark 3.3 it also takes, in some other solution, any other value lying in the same class as a or in a class that is above $[a]$ in P. In particular, values taken by frozen variables must belong to maximal classes in P that are one-element. Note that the condition $\{d \in D \mid f(d) = d \text{ for all } f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$ mentioned in Theorem 3.5 and in the next theorem means that each maximal class in P is not a singleton.

Let $\mathcal{Z} = \{B_1, \ldots, B_k\}$ be the set of all θ-classes B with the following property: there exists a maximal class $[a] \in P$ such that $[a] = \{a\}$ and $[a]$ is the only class in P with $B < [a]$; let a_i denote the element a corresponding to B_i in this definition. Finally, if $\mathcal{Z} \neq \emptyset$, fix arbitrary $b_i \in B_i$, $i = 1, \ldots, k$, and let Γ_i denote $\Gamma \cup \{\{b_i\}\}$.

Now we are ready to present the complete characterization of tractability of FV-CSP(Γ). Note that Proposition 3.4 implies that any such characterization must, for all non-trivial problems FV-CSP(Γ), contain the tractability condition for the corresponding CSP(Γ). We begin by proving a technical lemma that is used in the proof of Theorem 4.2.

Lemma 4.1 *If T is a maximal class in P or if $T = B_i \cup \{a_i\}$ then $T \in \langle\Gamma\rangle$.*

Proof. Let T be a maximal class in P and let $t' = f(t_1, \ldots, t_n)$ for some $f \in \mathsf{Pol}_n(\Gamma)$ and $t_1, \ldots, t_n \in T$. We have that for each $1 \leq i \leq n$ there exist $f_i \in \mathsf{Pol}_1(\Gamma)$ with $f_i(t_1) = t_i$. One can see that the function $f'(x) = f(f_1(x), \ldots, f_n(x))$ is a member of $\mathsf{Pol}_1(\Gamma)$ and $f'(t_1) = t'$. Using maximality of T we infer that $t' \in T$. Therefore, $T \in \mathsf{Inv}(\mathsf{Pol}(\Gamma))$, and, by Theorem 2.12, the result follows.

Let $T = B_i \cup \{a_i\}$. Since $\{a_i\}$ is a maximal class in P, the argument above shows that $f(a_i, \ldots, a_i) = a_i$ for all $f \in \mathsf{Pol}(\Gamma)$. Let $t' = f(t_1, \ldots, t_n)$ where $f \in \mathsf{Pol}_n(\Gamma)$, $t_1, \ldots, t_n \in T$, and at least one of the t_i, say t_1, belongs to B_i. By the definition of B_i, there exist $f_1, \ldots, f_n \in \mathsf{Pol}_1(\Gamma)$ such that $f_i(t_1) = t_i$ for all $1 \leq i \leq n$. Since $f'(x) = f(f_1(x), \ldots, f_n(x)) \in \mathsf{Pol}_1(\Gamma)$, $t' = f'(t_1)$, and the fact that $\{a_i\}$ is the only element in P above B_i, we conclude that $t' \in B_i \cup \{a_i\}$. As above, it follows that $T \in \langle\Gamma\rangle$. ∎

Theorem 4.2 *1.* FV-CSP(Γ) *is tractable if and only if one of the following conditions holds:*

(a) $\{d \in D \mid f(d) = d \text{ for all } f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$,

(b) CSP(Γ) *is tractable and* $\mathcal{Z} = \emptyset$,

(c) CSP(Γ_i) *is tractable for all* $1 \leq i \leq k$.

Input: An instance (I, V') of FV-CSP(Γ) and a variable $v \in V'$
Output: 'Yes' if v is frozen in I and 'No' otherwise.

1 for each maximal T in P **solve** I_T
2 if I_T is satisfiable for no T **then Output** 'No'
3 if I_T is satisfiable for more than one T **then Output** 'No'
4 if I_T is satisfiable for exactly one T and $|T| > 1$ **then Output** 'No'
5 if I_T is satisfiable for exactly one T and $T = \{a\}$ **then**
6   **if** $\mathcal{Z} = \emptyset$ or $\{B_j \mid a_j = a\} = \emptyset$ **then Output** 'Yes'
7   **else for** each j with $a_j = a$ **solve** I_j
8   **if** some I_j with $a_j = a$ is satisfiable **then Output** 'No'
9   **else Output** 'Yes'.

Figure 1: Algorithm for deciding tractable FV-CSP(Γ)

2. *If* CSP(Γ) *is tractable and* CSP(Γ_i) *is* **NP**-*complete for some* $1 \leq i \leq k$ *then* FV-CSP(Γ) *is* **coNP**-*complete.*

Proof. First we prove that conditions (a)-(c) in part 1) are sufficient.

If (I, V') is an instance of FV-CSP(Γ), $v \in V'$ is fixed, and T is a maximal class in P, let I_T denote the instance of CSP(Γ) obtained by adding the constraint (v, T) to I. Note that by Lemma 4.1 and Lemma 3.1, we may assume that $T \in \Gamma$. Furthermore, if $\mathcal{Z} \neq \emptyset$, let I_j, $1 \leq j \leq k$, denote the instance of CSP(Γ_j) obtained by adding the constraint $(v, \{b_j\})$ to I.

We assume that $\mathsf{Pol}_1(\Gamma)$, the poset P, and all maximal classes in P are already computed. It is easy to see that, since D is fixed, this can be done in polynomial time.

If condition (a) holds then FV-CSP(Γ) is trivial by Theorem 3.5. Suppose now that one of conditions (b) and (c) holds. We prove that the polynomial-time algorithm shown in Fig. 1 correctly decides whether a given variable is frozen in an instance of FV-CSP(Γ). Obviously, this is sufficient to prove tractability of this problem.

It is easy to see that this algorithm runs in polynomial time because all the I_T and all the I_j (if $\mathcal{Z} \neq \emptyset$) are instances of tractable problems. Let us prove correctness of the algorithm. It is not hard to verify that the conditions checked in the algorithm are jointly exhaustive and pairwise incompatible. Therefore, it is sufficient to show that every line produces the correct output, if any. Fix an instance (I, V') of FV-CSP(Γ) and a variable $v \in V'$. It follows from Remark 3.3 that if I is satisfiable then it has a solution φ with $\varphi(v) \in T$ for some maximal class T in P. Consequently, the algorithms outputs 'No' in line **2** if and only if I is not satisfiable. Since different classes in P do not intersect, fulfillment of conditions of line **3** implies that v is not frozen, so this line outputs the right answer. If the conditions in line **4** are satisfied then, by definition of T, v takes all values in T in solutions to I, that is, v is not frozen.

Assume now that the condition in line **5** is satisfied. Then we know that, for any solution φ to I, $\varphi(v)$ can take only values b such that $\{a\}$ is the only maximal class in P satisfying $[b] \leq \{a\}$. Assume that $\varphi(v) = b \neq a$ for some solution φ, and let T' be a maximal class among those satisfying $[b] \leq T' < \{a\}$. By Remark 3.3, there is a solution φ' to I such that $\varphi'(v) \in T'$. Since $\{a\}$ is the only maximal class above T', we conclude that T' must be B_j for some $1 \leq j \leq k$ such that $a_j = a$ (and, in particular, then $\mathcal{Z} \neq \emptyset$). Note that this justifies line **6** of the algorithm. Further, Remark 3.3 implies that any element in B_j (and, in particular, b_j) is the value of v in some solution to I, and, consequently, I_j is satisfiable. Clearly, if I_j is satisfiable for some $1 \leq j \leq k$ such that $a_j = a$, then v is not frozen in I. Thus, under conditions of line **5**, v is not frozen in I if and only if there is $1 \leq j \leq k$ such that $a_j = a$ and I_j is satisfiable. This argument justifies lines **7**-**9** of the algorithm.

We now prove the necessity of conditions (a)-(c). Assume, for contradiction, that none of these conditions holds. If condition (a) does not hold then, by Proposition 3.4, CSP(Γ) reduces to FV-CSP(Γ). Therefore, if CSP(Γ) is intractable, so is FV-CSP(Γ). Suppose now that CSP(Γ) is tractable. Since neither of conditons (b) and (c) holds, it follows that CSP(Γ_i) is intractable for some i, say $i = 1$. We present a polynomial-time reduction from the complement of this problem to FV-CSP(Γ). Note that $\{a_1\} \in \langle\Gamma\rangle$ and $B_1 \cup \{a_1\} \in \langle\Gamma\rangle$ by Lemma 4.1. By Lemma 3.1, we may assume that both relations are in Γ. Let I be an arbitrary instance of CSP(Γ_1). Modify I to an instance I' of CSP(Γ) as follows:

1. introduce a new variable z and a constraint $(z, B_1 \cup \{a_1\})$,
2. for every constraint of the form $(v, \{b_1\})$ in I,
   - remove this constraint from I,
   - identify all occurences of v in I (if they exist) with z.

Since CSP(Γ) is tractable, I' can be decided in polynomial time. It is easy to see that if I' is not satisfiable then neither is I. If this is the case, map I to the one-constraint instance $((x, \{a_1\}), \{x\})$ of FV-CSP(Γ).

If I' is satisfiable, then map I to the instance $(I', \{z\})$ of FV-CSP(Γ). We show that I is not satisfiable if and only if z is frozen in I'. If z is frozen in I' then z is assigned a_1 in all solutions to I' because $b \sqsubseteq a_1$ for any $b \in B_1$ (see Remark 3.3). We conclude that I is not satisfiable. If z is not frozen in I' then, since I' is satisfiable, it takes some value $b \in B_1$ in some solution φ.

By the definition of B_1, there is $f \in \mathsf{Pol}_1(\Gamma)$ such that $f(b) = b_1$. Therefore, $f\varphi$ is a solution to I.

We conclude that FV-CSP(Γ) cannot be tractable since, otherwise, the problem CSP(Γ_1) is tractable which contradicts the assumption made. Note that the second part of the theorem also follows from the reduction given above. ∎

The next corollary says that, whenever FV-CSP(Γ) is tractable, not only can the frozen variables be recognized efficiently, but also the unique values for them can be found in polynomial time.

Corollary 4.3 *Let* FV-CSP(Γ) *be tractable. Then the unique values for all frozen variables in any instance of* CSP(Γ) *can be found efficiently.*

Proof. If condition (a) of part 1 of Theorem 4.2 is satisfied then no variable in any instance of CSP(Γ) is frozen. Otherwise, for a given instance I of CSP(Γ), apply the algorithm in Fig. 1 to each variable v in I and, if v is frozen in I, it takes value a from line **5** of the algorithm. ∎

We now give some examples to show how Theorem 4.2 works.

Example 4.4 *If* CSP(Γ) *is tractable and all* $f \in \mathsf{Pol}_1(\Gamma)$ *are permutations then* FV-CSP(Γ) *is tractable. Indeed, since* $\mathsf{Pol}_1(\Gamma)$ *is a permutation group, the poset* P *is an antichain (all classes are pairwise incomparable). Therefore,* $\mathcal{Z} = \emptyset$ *and we can apply Theorem 4.2.*

Example 4.5 *Let* $D = \{0, 1, 2\}$ *and* Γ *consist of two relations,* ϱ_1 *and* ϱ_2*, on* D *where* $\varrho_1 = \{(0,0,0), (0,1,1), (0,1,2)\}$ *and*

$$\varrho_2 = \{(0,0,0,0), (1,0,0,1), (1,0,1,0), (1,1,0,0), (1,0,0,2)\}.$$

Obviously, CSP(Γ) *is tractable, since one can always satisfy all constraints by assigning 0 to all variables. It is easy to calculate that* $\mathsf{Pol}_1(\Gamma)$ *consists of three operations* f_1, f_2, f_3 *whose values on* $(0, 1, 2)$ *are the tuples in* ϱ_1*. In particular,* f_1 *is the 0 operation and* $f_3 = \mathrm{id}_D$ *is the identity operation on* D*. Then the quasi-order* $\sqsubseteq$ *satisfies* $2 \sqsubseteq 1 \sqsubseteq 0$ *and* $0 \not\sqsubseteq 1 \not\sqsubseteq 2$*. Therefore, we have* $\mathcal{Z} = \{B_1\}$ *where* $B_1 = \{1\}$ *and* $a_1 = 0$*. Then* $\Gamma_1 = \{\varrho_1, \varrho_2, \{1\}\}$*. Consider the relation* ϱ' *defined by*

$$\varrho'(x, y, z) \equiv (\exists u)(\exists v)(\exists w)(u = 1 \wedge \varrho_2(u, x, y, z) \wedge \varrho_1(v, z, w)).$$

One can easily check that ϱ' *is exactly the relation* N *defined in Example 2.3. So,* $N \in \langle \Gamma_1 \rangle$*, and, by Theorems 2.12, 2.15 and Example 2.3,* CSP(Γ_1) *is* **NP***-complete. Thus, by Theorem 4.2,* FV-CSP(Γ) *is* **coNP***-complete.*

Example 4.6 *In Example 4.5, replace the only occurence of 2 in ϱ_2 by 0. Let ϱ_2' denote the obtained relation and let $\Gamma' = \{\varrho_1, \varrho_2'\}$. Again, it is easy to compute that $\mathsf{Pol}_1(\Gamma')$ coincides with $\mathsf{Pol}_1(\Gamma)$, and, therefore, the quasi-order $\sqsubseteq$ is the same as in Example 4.5, and so $\Gamma_1' = \{\varrho_1, \varrho_2', \{1\}\}$. One can straightforwardly check that the binary operation $\min(x, y)$ on D, which takes the minimum of x and y with respect to the natural order on D, belongs to $\mathsf{Pol}(\Gamma_1')$. It follows from [23] that $\mathrm{CSP}(\Gamma_1')$ is tractable. Thus, by Theorem 4.2, FV-CSP(Γ') is tractable.*

Note that if the conjecture that every CSP(Γ) is either tractable or **NP**-complete holds (and there is strong evidence that it does [5, 4, 6, 7, 8, 9, 10, 16, 21, 22, 42]) then Theorems 3.5 and 4.2 also give a complete characterization of **coNP**-complete problems FV-CSP(Γ) (under the assumption that **NP** $\neq$ **coNP**). It was proved in [5] that, for $|D| \leq 3$, this conjecture is true which implies the following result.

Corollary 4.7 *Let $|D| \leq 3$. Then, for every $\Gamma \subseteq R_D$,* FV-CSP(Γ) *is either tractable,* **coNP***-complete, or* **NP***-hard.*

Proof. The problem CSP(Γ) is either tractable or **NP**-complete by [5] since $|D| \leq 3$. If $\{d \in D \mid f(d) = d \text{ for all } f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$ then FV-CSP(Γ) is tractable by Theorem 4.2. Assume that this set is non-empty. Now, if CSP(Γ) is **NP**-complete then FV-CSP(Γ) in **NP**-hard by Proposition 3.4. Otherwise, CSP(Γ) is tractable and, since every problem CSP(Γ_i) is either tractable or **NP**-complete, Theorem 4.2 implies that FV-CSP(Γ) is either tractable or **coNP**-complete. ∎

Bulatov [5] also exhibits a polynomial-time algorithm which determines, for a given finite $\Gamma \subseteq R_D$, into which case the problem FV-CSP(Γ) falls. In Section 6 we will give a more precise classification for the cases when $|D| = 2$ and when Γ contains all unary relations.

5 NP-complete and DP-complete problems

In this section, we exhibit sufficient conditions for FV-CSP(Γ) to be **DP**-complete and **NP**-complete.

Assume that Γ can express all relations of the form $\{a\}$, that is, $\{a\} \in \langle\Gamma\rangle$ for all $a \in D$. This is equivalent to saying that $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$. Indeed, if $\{a\} \in \langle\Gamma\rangle$ for all $a \in D$ then $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$ by Theorem 2.12 and the definition of a polymorphism. Conversely, since $f'(x, \ldots, x) \in \mathsf{Pol}_1(\Gamma)$ for all $f' \in \mathsf{Pol}(\Gamma)$, it follows that, for all $a \in D$, $f'(a, \ldots, a) = a$, and, by Theorem 2.12, $\{a\} \in \langle\Gamma\rangle$. By Lemma 3.1, we may assume that Γ contains all unary relations of the form $\{a\}$, $a \in D$. We know from Example 4.4 that in this case FV-CSP(Γ) is tractable whenever CSP(Γ) is tractable. We will show that if CSP(Γ) is **NP**-complete then in many cases FV-CSP(Γ) is **DP**-complete.

Lemma 5.1 *Let $\Gamma \subseteq R_D$ and $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$. Suppose that there exists a subset $A \subseteq D$ such that $A \in \langle \Gamma \rangle$, and an equivalence relation $\epsilon \in \langle \Gamma \rangle$ on A, with exactly two classes A_0 and A_1, such that the ternary relation*

$$\varrho_\epsilon = \{(a,b,c) \in A^3 \mid |\{a,b,c\} \cap A_1| = 1\}$$

(with some choice of A_1) belongs to $\langle \Gamma \rangle$. Then CSP(Γ) *is* **NP***-complete.*

Proof. It is easy to see that ϱ_ϵ is the 1-IN-3-SAT relation N, as defined in Example 2.3, with elements from A_1 playing the role of 1, and elements of A_0 playing the role of 0. Since CSP($\{N\}$) is **NP**-complete (see Example 2.3), CSP($\{\varrho_\epsilon\}$) is **NP**-complete as well. By Theorems 2.12 and 2.15, CSP(Γ) is **NP**-complete. ∎

It was proved in [8, 9] that, to classify the complexity of problems CSP(Γ) over all finite domains, it is enough to look at the problems CSP(Γ) with $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$, and it was conjectured (with appropriate reformulation in terms of universal algebras) [5, 8, 9] that all **NP**-complete problems CSP(Γ) with $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$ are described in Lemma 5.1, and that all other problems CSP(Γ) with the given property are tractable. This conjecture is confirmed in many special cases, in particular, it is true when $|D| \leq 3$ [5, 22, 42] or when Γ contains all unary relations [6].

Let $f : A \to \{0,1\}$ be such that $f^{-1}(1) = A_1$ and $f^{-1}(0) = A_0$. For a relation $\varrho \in R_{\{0,1\}}$ define a relation $f^{-1}(\varrho)$ over A by letting $\vec{a} \in f^{-1}(\varrho)$ if and only if $f(\vec{a}) \in \varrho$.

Lemma 5.2 *The set $\langle \{\varrho_\epsilon\} \rangle$ contains all relations of the form $f^{-1}(\varrho)$ where $\varrho \in R_{\{0,1\}}$.*

Proof. Consider the relation N from Example 2.3 and notice that $\varrho_\epsilon = f^{-1}(N)$. Pick arbitrary $\varrho \in R_{\{0,1\}}$. By Example 2.14, we have $\varrho \in \langle \{N\} \rangle$, that is, ϱ is expressible by a first-order formula Φ using only predicates N and $=$, conjunction, and existential quantifiers (see Definition 2.11). We may assume that the formula does not use the equality predicate on $\{0,1\}$, since one can check that

$$x = y \quad \Leftrightarrow \quad (\exists v)(\exists u)(\exists w)(N(v,u,u) \wedge N(x,u,w) \wedge N(y,u,w)).$$

Also, we may without loss of generality assume that the formula Φ is prenex normal form. It is not hard to see that if one replaces N by $\varrho_\epsilon = f^{-1}(N)$ in Φ then the new formula will produce $f^{-1}(\varrho)$. By Definition 2.11, this proves the lemma. ∎

It follows from Lemma 5.2 that, in Lemma 5.1, if $\varrho_\epsilon \in \langle \Gamma \rangle$ for one choice of A_1 then the same is true for the other choice of A_1 too.

Theorem 5.3 *Suppose that $\Gamma \subseteq R_D$ satisfies the conditions of Lemma 5.1, and A, ϵ, and A_1 can be chosen so that A_1 is one-element. Then* FV-CSP(Γ) *is* **DP**-*complete.*

Proof. By Lemma 3.1, we may assume that $\varrho_\epsilon \in \Gamma$. An instance of the SAT-UNSAT problem is given by a pair (F, F') of propositional formulas. The question is whether it is true that F is satisfiable and F' is not. This problem is known to be **DP**-complete [33, 35]. A very similar proof shows that 3SAT-3UNSAT, a restriction of the above problem with F and F' having three literals per clause, is also **DP**-complete. We will reduce 3SAT-3UNSAT to FV-CSP(Γ).

Let $A_1 = \{a\}$ and Δ the set of all at most 4-ary relations in $\langle \varrho_\epsilon \rangle$. Consider the function $f : A \to \{0, 1\}$ given by $f(a) = 1$ and $f(b) = 0$ if $b \neq a$. Arbitrarily choose a 3SAT-3UNSAT instance (F, F') and construct an FV-CSP(Δ) instance $(I, \{u\})$ with $I = (V, D, \mathcal{C})$ defined as follows. Let V be the set of variables used in (F, F') together with one new variable u. For every clause $c(x, y, z)$ in F introduce constraint $((x, y, z), \varrho_c)$ where

$$(a_1, a_2, a_3) \in \varrho_c \Leftrightarrow c(f(a_1), f(a_2), f(a_3)) = 1.$$

Then, every clause in $c'(x, y, z)$ in F', introduce a constraint $((x, y, z, u), \sigma_{c'})$ where

$$(a_1, a_2, a_3, a_4) \in \sigma_{c'} \Leftrightarrow (c'(f(a_1), f(a_2), f(a_3)) = 1 \text{ or } f(a_4) = 1).$$

Let $\mathcal{C}$ be the collection of constraints obtained in this way. Note that the constraints built from F mimic clauses in F with a playing the role of 1, and all (indistinguishable) elements of A_0 playing the role of 0. The constraints built from F' effectively say that if F' is satisfied then u can take any value, but if F' is not satisfied then u must take value a. Note that, by Lemma 5.2, all constraint relations ϱ_c and $\sigma_{c'}$ belong to $\langle \{\varrho_\epsilon\} \rangle$, and, therefore, to Δ. Moreover, since D is fixed, the transformation can be performed in polynomial time.

We prove that u is frozen in I if and only if (F, F') is a 'yes'-instance of 3SAT-3UNSAT. Assume that F is not satisfiable. Then neither is I. Assume that both F and F' are satisfiable. Then I has one solution in which u takes value a and one in which u takes some other value. Finally, assume that F is satisfiable and F' is not. In this case, $\varphi(u) = a$ for every solution φ to I. Hence, FV-CSP(Δ) is **DP**-complete. By Lemma 3.1, we conclude that FV-CSP($\{\varrho_\epsilon\}$) (and, hence, FV-CSP(Γ)) is **DP**-complete. ■

Example 5.4 *Reconsider relation N from Example 2.3. In this case,* **DP**-*completeness of* FV-CSP($\{N\}$) *follows immediately from Theorem 5.3. Let us now consider the* AUDIT *problem which was defined in Section 1. First,* *the fact that $N(x,y,z)$ holds if and only if $x+y+z=1$ suggests that the* AUDIT *problem could be viewed as a subproblem of* FV-CSP($\{N\}$)*. However, if the given set of equations has no solution, then the corresponding* AUDIT *instance is considered to be a 'yes'-instance (and not a 'no'-instance). A straightforward modification of the proof of Theorem 5.3 solves this problem: simply do the reduction from the* **coNP**-*complete problem* 3UNSAT *(instead of the* **DP**-*complete problem* 3SAT-3UNSAT*) by removing the* 3SAT-*formula F from the transformed instance (F,F'). This reduction shows that the* AUDIT *problem is* **coNP**-*complete.*

We would like to point out that the difference in problem definition between FV-CSP and AUDIT (i.e. in the AUDIT problem, we check whether *at least* one variable is frozen or not) does not affect the complexity in the example above. We also would like to point out that the AUDIT problem is **coNP**-complete for a wider range of statistical queries than summation queries. In fact, as soon as the set of relations which are expressible by the statistical queries satisfies the preconditions of Theorem 5.3, then the AUDIT problem for such queries is **coNP**-complete.

We now give some examples of **NP**-complete problems FV-CSP(Γ). Let $f: D \to E$ be onto and such that $|f^{-1}(e)| > 1$ for all $e \in E$. Take any $\Gamma' \subseteq R_E$ such that CSP(Γ') is **NP**-complete, and consider $\Gamma \subseteq R_D$ consisting of all relations $f^{-1}(\varrho), \varrho \in \Gamma'$, where $\vec{a} \in f^{-1}(\varrho)$ if and only if $f(\vec{a}) \in \varrho$.

Proposition 5.5 FV-CSP($\Gamma \cup \{\{d\} \mid d \in D\}$) *is* **NP**-*complete.*

Proof. It is easy to see that CSP(Γ) is **NP**-complete which implies that CSP($\Gamma \cup \{\{d\} \mid d \in D\}$) is **NP**-complete. In every solution φ of every satisfiable instance of CSP(Γ), the value $\varphi(x)$ of any variable x can be changed to any other value a such that $f(a) = f(\varphi(x))$. Take an arbitrary instance (I, V') of FV-CSP($\Gamma \cup \{\{d\} \mid d \in D\}$). If there is a variable in $v \in V'$ on which no constraint of the form $(v, \{d\})$ is imposed, or if there are two such constraints with the same v but different d, then V' is not frozen in I, since no solution (if it exists) is unique on V'. If, in I, there is a unique constraint of the form $(v, \{d\})$ for every $v \in V'$ then the problem is equivalent to deciding whether I is satisfiable. Hence, FV-CSP($\Gamma \cup \{\{d\} \mid d \in D\}$) is in **NP**. Now the result follows from Proposition 3.4. ∎

Example 5.6 *Let $D = \{0,1,\ldots,k\}$, $k \geq 3$, and $f: D \to \{0,1\}$ be such that $f(0) = f(1) = 0$ and $f(a) = 1$ otherwise. Reconsider relation N defined in Example 2.3, and let $\varrho = f^{-1}(N)$. It follows from Proposition 5.5 and Example 2.3 that* FV-CSP($\{\varrho, \{0\}, \{1\}, \ldots, \{k\}\}$) *is* **NP**-*complete.*

Proposition 5.7 *Fix $a \in D$ and let $D' = D \backslash \{a\}$. Let $\Gamma \subseteq R_{D'}$ be such that* $\mathrm{CSP}(\Gamma)$ *is* **NP***-complete and* $\{d \in D' \mid f(d) = d$ *for all* $f \in \mathsf{Pol}_1(\Gamma)\} = \emptyset$. *Then* FV-$\mathrm{CSP}(\Gamma \cup \{\{a\}\})$ *is* **NP***-complete.*

Proof. Similar to that of Proposition 5.5. ∎

Example 5.8 *Let $D = \{0, 1, 2\}$ and let N' be as defined in Example 2.3. It is easy to see that the permutation that swaps 0 and 1 is a polymorphism of N'. Therefore,* FV-$\mathrm{CSP}(\{N', \{2\}\})$ *is* **NP***-complete by Proposition 5.7.*

6 Two complete classifications

In this section we show how results from the previous sections work. First, we obtain the following classification result.

Theorem 6.1 *Let $\Gamma \subseteq R_{\{0,1\}}$. Then*

1. *if* $\mathsf{Pol}(\Gamma)$ *contains both constant operations, 0 and 1, or one of the operations (c)-(g) from Proposition 2.13 then* FV-$\mathrm{CSP}(\Gamma)$ *is tractable;*

2. *else, if exactly one of 0 and 1 is in* $\mathsf{Pol}(\Gamma)$ *then* FV-$\mathrm{CSP}(\Gamma)$ *is* **coNP***-complete;*

3. *else,* FV-$\mathrm{CSP}(\Gamma)$ *is* **DP***-complete.*

Proof. 1) The cases when $\mathsf{Pol}(\Gamma)$ contains both constant operations or the negation operation are trivial by Theorem 3.5. In cases (d)-(g), the problem $\mathrm{CSP}(\Gamma \cup \{\{0\}, \{1\}\})$ is tractable by Theorem 2.16, and we can apply Theorem 4.2.

2) Assume that $1 \in \mathsf{Pol}_1(\Gamma)$ and $0 \notin \mathsf{Pol}_1(\Gamma)$. In this case, $\mathrm{CSP}(\Gamma)$ is tractable by Theorem 2.16. We have $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_{\{0,1\}}, 1\}$. Therefore the quasi-order defined before Theorem 4.2 satisfies $0 \sqsubseteq 1$ and $1 \not\sqsubseteq 0$, and we have $\mathcal{Z} = \{B_1\}$ where $B_1 = \{0\}$ and $a_1 = 1$. It follows from Theorem 2.16 that $\mathrm{CSP}(\Gamma \cup \{\{0\}\})$ is **NP**-complete. By Theorem 4.2, we conclude that FV-$\mathrm{CSP}(\Gamma)$ is **coNP**-complete.

3) If Γ does not satisfy any of the conditions mentioned in 1) and 2) then, by Proposition 2.13 and Theorem 2.12, we have $\langle \Gamma \rangle = R_{\{0,1\}}$. It follows from Lemma 3.1 and Example 5.4 that FV-$\mathrm{CSP}(\Gamma)$ is **DP**-complete. ∎

It is easy to see that all conditions in Theorem 6.1 can be verified efficiently for any finite $\Gamma \subseteq R_{\{0,1\}}$. Another interesting consequence of Theorem 6.1 (and of Propositions 5.5 and 5.7) is that **NP**-complete problems FV-$\mathrm{CSP}(\Gamma)$, $\Gamma \subseteq R_D$, exist if and only if $|D| > 2$.

We note that the first two parts of Theorem 6.1 coincide (when appropriately re-stated) with the corresponding parts of the classification for

UNIQUE SAT(Γ) [25], while the last part of Theorem 6.1 gives more precise information than the corresponding part in [25].

Now let us consider *conservative* CSPs that are a generalization of the well-studied LIST HOMOMORPHISM problems for graphs [6, 14, 15]. Let D be arbitrary finite, and suppose that $R_D^{(1)} \subseteq \Gamma \subseteq R_D$, that is Γ contains all unary relations. This condition means that one can specify, for each variable in any instance of CSP(Γ), its own domain within D. It also follows that, for every $f \in \mathsf{Pol}(\Gamma)$ and every $B \subseteq D$, $f(b_1, \ldots, b_n) \in B$ whenever $b_1, \ldots, b_n \in B$, and we can therefore consider restrictions of polymorphisms onto arbitrary subsets. The complexity of CSP(Γ) for such sets Γ is completely classified in [6]. Let $f|_B$ denote the restriction of a function f onto a set B.

Theorem 6.2 ([6]) *Let D be arbitrary finite and $R_D^{(1)} \subseteq \Gamma \subseteq R_D$. If, for every two-element $B \subseteq D$, there is an at most ternary $f \in \mathsf{Pol}(\Gamma)$ such that $f|_B$ (up to the names of elements) is one of the functions (d)-(g) from Proposition 2.13 then* CSP(Γ) *is tractable. Otherwise it is* **NP***-complete. Moreover, the tractable cases can be recognized efficiently.*

The previous results can be combined in order to prove the following classification result.

Theorem 6.3 *Let D be finite and $R_D^{(1)} \subseteq \Gamma \subseteq R_D$. Then* FV-CSP($\Gamma$) *is tractable if* CSP($\Gamma$) *is tractable, and it is* **DP***-complete otherwise.*

Proof. Since $\{d\} \in \Gamma$ for each $d \in D$, we have $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$. If CSP($\Gamma$) is tractable then, by Theorem 4.2, so is FV-CSP(Γ). Assume now that CSP(Γ) is **NP**-complete. By Theorem 6.2, there exists a two-element $B \subseteq D$ (say, $B = \{0,1\}$) such that, for each $f \in \mathsf{Pol}(\Gamma)$, $f|_B$ is not of the form (d)-(g) of Proposition 2.13. Using the superposition-based definition of a clone, it is easy to show that $C = \{f|_B \mid f \in \mathsf{Pol}(\Gamma)\}$ is a clone on $\{0,1\}$. Since $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_D\}$, it follows that for each $f \in \mathsf{Pol}(\Gamma)$, $f|_B$ is not of the form (a)-(c) of Proposition 2.13. Hence, by Proposition 2.13, C consists of all projections on $\{0,1\}$. Since every relation on $\{0,1\}$ is invariant under all projections on $\{0,1\}$, we have $N \in \mathsf{Inv}(C)$. By the definition of C and the fact that every operation in $\mathsf{Pol}(\Gamma)$ preserves $\{0,1\}$, we have $N \in \mathsf{Inv}(\mathsf{Pol}(\Gamma)) = \langle\Gamma\rangle$. The problem FV-CSP($\{N\}$) is **DP**-complete by Theorem 5.3 (see also Example 5.4). It follows from Lemma 3.1 that FV-CSP(Γ) is **DP**-complete. ∎

Moreover, as noticed in [6], all conditions in Theorem 6.2 can be verified efficiently for any finite $\Gamma \subseteq R_D$.

7 Conclusion

We have continued the study of connections between algebraic theory and the computational complexity of constraint satisfaction problems. This idea was originally developed for studying the standard constraint satisfaction problems where the question is to decide the existence of a solution. This paper clearly shows that this approach leads to general results for a wider range of problems with different computational properties. For example, the frozen variable problems in constraint satisfaction can be tractable, **NP**-, **coNP**- and **DP**-complete.

One of the results in this paper is a characterization of the tractable cases of the frozen variable problem, which also provides a characterization of **coNP**-complete cases if the dichotomy conjecture for the standard CSP holds. We have shown that further progress in classifying the complexity of the frozen variable problem will strongly, though not completely, depend on the progress with the standard CSP. Indeed, even assuming dichotomy for the standard CSP and also that hard CSPs give rise only to **NP**-complete or **DP**-complete cases of FV-CSP(Γ), it remains open to separate the cases for non-Boolean domains, which seems to be quite a challenging task. For $|D| = 3$, this can probably be accomplished by refining the techniques used in this paper and combining them with the algebraic results achieved in the process of completely classifying the complexity of the standard CSP problem over a three-element domain [5].

An interesting direction of future research would also be to find out to what extent our results on the frozen variable problem can be applied to the UNIQUE CSP problem which is the problem of recognizing CSP instances with a unique solution. Is it true that tractable cases of the two problems are the same? Obviously, every tractable case of the frozen variable problem gives rise to a tractable case of UNIQUE CSP, and implication in the other direction is also true for $|D| = 2$, as mentioned in Section 6.

Acknowledgements

We would like to thank Andrei Bulatov and Peter Jeavons who provided comments on an earlier version of this paper, and Jon M. Kleinberg who provided a number of clarifying remarks. Peter Jonsson is partially supported by the *Center for Industrial Information Technology* (CENIIT) under grant 04.01, and the *Swedish Research Council* (VR) under grant 621-2002-4126. Andrei Krokhin is partially supported by the UK EPSRC grant GR/R29598 and by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

References

[1] N. Adam, J. Wortman, Security-control methods for statistical databases: a comparative study, ACM Computing Surveys 21 (4) (1989) 515–556.

[2] B. Bollobás, C. Borgs, J. Chayes, J. Kim, D. Wilson, The scaling window of the 2-SAT transition, Random Structures and Algorithms 18 (3) (2001) 201–256.

[3] F. Börner, A. Bulatov, P. Jeavons, A. Krokhin, Quantified constraints: algorithms and complexity, in: Proceedings 17th Workshop on Computer Science Logic, CSL'03, 2003, pp. 58-70.

[4] A. Bulatov, Mal'tsev constraints are tractable, Tech. Rep. PRG-RR-02-05, Computing Laboratory, University of Oxford, UK (2002).

[5] A. Bulatov, A dichotomy theorem for constraints on a three-element set, in: Proceedings of 43rd IEEE Symposium on Foundations of Computer Science, FOCS'02, 2002, pp. 649–658.

[6] A. Bulatov, Tractable conservative constraint satisfaction problems, in: Proceedings 18th IEEE Symposium on Logic in Computer Science, LICS'03, 2003, pp. 321–330.

[7] A. Bulatov, P. Jeavons, Tractable constraints closed under a binary operation, Tech. Rep. PRG-TR-12-00, Computing Laboratory, University of Oxford, UK (2000).

[8] A. Bulatov, P. Jeavons, A. Krokhin, Classifying complexity of constraints using finite algebras, submitted for publication.

[9] A. Bulatov, A. Krokhin, P. Jeavons, Constraint satisfaction problems and finite algebras, in: Proceedings 27th International Colloquium on Automata, Languages and Programming, ICALP'00, Vol. 1853 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 272–282.

[10] A. Bulatov, A. Krokhin, P. Jeavons, The complexity of maximal constraint languages, in: Proceedings 33rd ACM Symposium on Theory of Computing, STOC'01, 2001, pp. 667–674.

[11] N. Creignou, S. Khanna, M. Sudan, Complexity Classifications of Boolean Constraint Satisfaction Problems, Vol. 7 of SIAM Monographs on Discrete Mathematics and Applications, 2001.

[12] J. Culberson, I. Gent, Frozen development in graph coloring, Theoretical Computer Science 265 (1-2) (2001) 227–264.

[13] V. Dalmau, Constraint satisfaction problems in non-deterministic logarithmic space, in: Proceedings 29th International Colloquium on Automata, Languages and Programming, ICALP'02, Vol. 2380 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 414–425.

[14] T. Feder, P. Hell, J. Huang, List homomorphisms and circular arc graphs, Combinatorica 19 (1999) 487–505.

[15] T. Feder, P. Hell, J. Huang, Bi-arc graphs and the complexity of list homomorphisms, Journal of Graph Theory 42 (2003) 61–80.

[16] T. Feder, M. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory, SIAM Journal on Computing 28 (1998) 57–104.

[17] C. Gomes, B. Selman, H. Kautz, Heavy-tailed phenomena in satisfiability and constraint satisfaction, Journal of Automated Reasoning 24 (2000) 67–100.

[18] G. Gottlob, L. Leone, F. Scarcello, Hypertree decomposition and tractable queries, Journal of Computer and System Sciences 64 (3) (2002) 579–627.

[19] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, in: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS'2003), 2003, pp. 552–561.

[20] J. Håstad, Some optimal inapproximability results, Journal of the ACM 48 (2001) 798–859.

[21] P. Hell, J. Nešetřil, On the complexity of H-coloring, Journal of Combinatorial Theory, Ser.B 48 (1990) 92–110.

[22] P. Jeavons, On the algebraic structure of combinatorial problems, Theoretical Computer Science 200 (1998) 185–204.

[23] P. Jeavons, D. Cohen, M. Gyssens, Closure properties of constraints, Journal of the ACM 44 (1997) 527–548.

[24] P. Jeavons, D. Cohen, M. Gyssens, How to determine the expressive power of constraints, Constraints 4 (1999) 113–131.

[25] L. Juban, Dichotomy theorem for generalized unique satisfiability problem, in: Proceedings 12th Conference on Fundamentals of Computation Theory, FCT'99, Vol. 1684 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 327–337.

[26] H. Kautz, B. Selman, Planning as satisfiability, in: Proceedings 10th European Conference on Artificial Intelligence, ECAI'92, 1992, pp. 359–363.

[27] J. Kleinberg, C. H. Papadimitriou, P. Raghavan, Auditing Boolean attributes, Journal of Computer and System Sciences 66 (2003) 244–253.

[28] P. Kolaitis, M. Vardi, Conjunctive-query containment and constraint satisfaction, Journal of Computer and System Sciences 61 (2000) 302–332.

[29] A. Krokhin, A. Bulatov, P. Jeavons, Functions of multiple-valued logic and the complexity of constraint satisfaction: A short survey, in: Proceedings 33rd IEEE International Symposium on Multiple-Valued Logic, ISMVL'03, 2003, pp. 343–351.

[30] K. Marriott, P. Stuckey, Programming with Constraints: an Introduction, MIT Press, 1998.

[31] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, 2+p-SAT: Relation of typical-case complexity to the nature of phase transition, Random Structures and Algorithms 15 (3-4) (1999) 414–440.

[32] U. Montanari, Networks of constraints: Fundamental properties and applications to picture processing, Information Sciences 7 (1974) 95–132.

[33] C. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[34] C. Papadimitriou, D. Wolfe, The complexity of facets resolved, Journal of Computer and System Sciences 37 (1) (1988) 2–13.

[35] C. Papadimitriou, M. Yannakakis, The complexity of facets (and some facets of complexity), Journal of Computer and System Sciences 28 (2) (1982) 244–259.

[36] A. Parkes, Clustering at the phase transition, in: Proceedings 14th National (US) Conference on Artificial Intelligence, AAAI'97, 1997, pp. 340–345.

[37] N. Pippenger, Theories of Computability, Cambridge University Press, Cambridge, 1997.

[38] R. Pöschel, Galois connections for operations and relations, Tech. Rep. MATH-AL-8-2001, Technische Universität Dresden, Germany (2001).

[39] R. Pöschel, L. Kalužnin, Funktionen- und Relationenalgebren, DVW, Berlin, 1979.

[40] E. Post, The two-valued iterative systems of mathematical logic, Vol. 5 of Annals Mathematical Studies, Princeton University Press, 1941.

[41] I. Rosenberg, Minimal clones I: the five types, in: Lectures in Universal Algebra (Proc. Conf. Szeged 1983), Vol. 43 of Colloq. Math. Soc. Janos Bolyai, North-Holland, 1986, pp. 405–427.

[42] T. Schaefer, The complexity of satisfiability problems, in: Proceedings 10th ACM Symposium on Theory of Computing, STOC'78, 1978, pp. 216–226.

[43] J. Singer, I. Gent, A. Smaill, Backbone fragility and the local search cost peak, Journal of Artificial Intelligence Research 12 (2000) 235–270.

[44] A. Szendrei, Clones in Universal Algebra, Vol. 99 of Seminaires de Mathematiques Superieures, University of Montreal, 1986.

[45] L. Valiant, V. Vazirani, NP is as easy as detecting unique solutions, Theoretical Computer Science 47 (3) (1986) 85–93.

[46] P. van Beek, Reasoning about qualitative temporal information, Artificial Intelligence 58 (1992) 297–326.