Non-uniform Boolean Constraint Satisfaction Problems with Cardinality Constraint

NADIA CREIGNOU
Aix-Marseille Université
and
HENNING SCHNOOR
Christian-Albrechts-Universität Kiel
and
ILKA SCHNOOR
Universität zu Lübeck

We study the computational complexity of Boolean constraint satisfaction problems with cardinality constraint. A Galois connection between clones and co-clones has received a lot of attention in the context of complexity considerations for constraint satisfaction problems. This connection does not seem to help when considering constraint satisfaction problems that support in addition a cardinality constraint. We prove that a similar Galois connection, involving a weaker closure operator and partial polymorphisms, can be applied to such problems. Thus, we establish dichotomies for the decision as well as for the counting problems in Schaefer's framework.

Categories and Subject Descriptors: F2.3 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Computational Complexity, Constraint Satisfaction

---

1. INTRODUCTION

The success of Boolean constraint satisfaction problems (CSPs) is due to two features: they provide a framework in which various combinatorial problems (including NP-complete ones) can be adequately expressed, and which is practically efficient since highly optimized solvers are available. Therefore, Boolean constraint satisfaction problems are an important test-bed for questions about computational complexity and algorithms. In particular the non-uniform version, $\text{Csp}(\Gamma)$, has been extensively studied from the computational complexity point of view. In this context a finite set of Boolean relations Γ, called a constraint language, is fixed. An input of such a problem is a Γ-formula. Such a formula is a conjunction of "clauses", each of which consisting in an application of some relation from Γ to

---

This work was supported by the DAAD postdoc program and the ANR *Algorithms and complexity* 07-BLAN-0327-04. Work done in part while the second and third authors worked at the University of Hannover.

variables. This framework captures many well-known combinatorial problems, as for instance the famous NP-complete problem 3SAT. The complexity study of these problems started in 1978 with the seminal paper of Schaefer [Schaefer 1978]. He proved a remarkable dichotomy theorem: CSP(Γ) is either in P or NP-complete. Since then many other algorithmic problems related to Γ-formulas have been investigated: including counting [Creignou and Hermann 1996], non-monotonic reasoning [Kirousis and Kolaitis 2003; Creignou and Zanuttini 2006], equivalence and isomorphism [Böhler et al. 2002; 2004; Bauland and Hemaspaandra 2005], optimization [Creignou 1995; Khanna et al. 2001; Reith and Vollmer 2003], parameterized complexity [Marx 2005], and many others (see [Creignou and Vollmer 2008] for a complete survey).

All the constraints that appear in non-uniform CSPs are local ones: each applies to a fixed number of variables. However in practice one is often faced with constraints of global nature, which involve all the variables occurring in the input. Due to the wide embrace of global constraints in the constraint programming community, we believe that the computational complexity of non-uniform CSPs supporting additional global constraints is worth being investigated. To this aim, we focus here on Boolean constraint satisfaction problems that support in addition a constraint of global nature, namely a cardinality constraint. Given a Γ-formula, a feasible solution is a satisfying assignment that fulfills in addition some cardinality constraint on the number of variables set to 1. More precisely we are interested in two problems BAL-CSP(Γ) and K-ONES(Γ). For the balanced constraint satisfaction problem BAL-CSP(Γ), the global constraint is that the assignment is balanced, i.e., it sets the same number of variables to 0 and 1. For K-ONES(Γ), the requirement is that exactly k variables (where k is given in the input) are set to 1. These two global constraints are well-known in constraint programming and appear in the Global constraint catalog (see http://www.emn.fr/x-info/sdemasse/gccat/index.html). The balanced constraint also arises naturally in some optimization problems. For example MIN-BISECTION can be seen as MIN-CUT with the restriction that the two sets of vertices have the same cardinality. Other optimization problems can be expressed as a Boolean constraint satisfacti on problem where a feasible solution is a balanced assignment. Recently, there was an increased interest in optimization problems supporting an additional cardinality constraint, see e.g. [Sviridenko 2001; Bazgan and Karpinski 2005; Bläser et al. 2008].

Satisfiability problems with an additional cardinality constraint first appeared in [Khanna et al. 2001]. The authors studied the problem MAX-ONES(Γ), in which a solution is a satisfying assignment that sets at least k variables to 1[1]. The problem K-ONES(Γ) was already studied from the point of view of parameterized complexity in [Marx 2005] (our results differ from his, since in our problems k, the number of variables that have to be set to true, is part of the input instance). The study of the complexity of BAL-CSP(Γ) and K-ONES(Γ) was initiated in [Bazgan and Karpinski 2005]. The authors identified a polynomial time case, obtained individual hardness results for specific constraint languages and conjectured a dichotomy classification. In this paper we prove that the conjecture holds. We prove a full complexity

[1]They studied this problem as an optimization problem, and were interested in approximability properties.

classification for the two problems BAL-CSP(Γ) and K-ONES(Γ). Moreover, we also tackle the corresponding counting problems and prove a dichotomy classification FP/#P-complete.

For this we use algebraic tools developed in [Schnoor and Schnoor 2008]. In order to obtain a complexity classification for constraint satisfaction problems, the main idea is to compare the so-called *expressive power* of constraint languages. Roughly speaking, given two constraint languages Γ_1 and Γ_2, if Γ_2 is at least as expressive than Γ_1, then any Γ_1-formula (i.e., a conjunction of Γ_1-clauses, each of which being an application of some relation from Γ_1 to variables) can be transformed into a Γ_2-formula. In the last decade, a Galois correspondence between the lattice of Boolean relations and the lattice of Boolean functions, together with Post's lattice has turned out to be one of the most successful tools to derive complexity results for Boolean constraint satisfaction problems. Indeed, this Galois correspondence relates the expressive power of a constraint language to its set of *polymorphisms*, i.e., algebraic closure properties. The structure of the polymorphism sets, so called clones, is well-known and is described by Post's lattice [Post 1941]. This Galois connection gives a procedure transforming Γ_1-formulas into satisfiability equivalent Γ_2-formulas. However, the newly constructed Γ_2-formulas contain additional existentially quantified variables and equality clauses can occur. These features make this Galois connection apparently difficult to use in order to transfer results from Post's classes to complexity when there is an additional cardinality constraint.

We prove that we can use a restricted closure, based on *partial polymorphisms* and studied in [Romov 1981]. These partial polymorphisms form a structure which is a refinement of the clone structure exhibited by Post. However, surprisingly, the complexity classification, when achieved, obeys the borders of Post's lattice.

In Section 2 we introduce the main concepts precisely, and state our results. In Section 3 we present the algebraic method from [Schnoor and Schnoor 2008] that will be used to obtain the complexity classification, providing an extended example for the required constructions at the end of the section. Section 4 is then dedicated to the hardness proofs.

2. MAIN RESULT

A *logical relation* of arity k is a relation $R \subseteq \{0,1\}^k$. A *constraint* (or *constraint application*) is a formula $R(x_1, \ldots, x_k)$, where R is a logical relation of arity k and $x_1, \ldots, x_k$ are (not necessarily distinct) variables. An assignment I of truth values to the variables *satisfies* the constraint if $\big(I(x_1), \ldots, I(x_k)\big) \in R$. If V is a subset of the variables, then with $I|_V$ we denote the restriction of I to V. A *constraint language* Γ is a finite set of logical relations. A Γ-*formula*, φ, is a conjunction of constraint applications using only logical relations from Γ, and hence is a quantifier-free first-order formula. With $\mathrm{Var}(\varphi)$ we denote the set of variables appearing in φ. A Γ-formula φ is satisfied by an assignment I if I satisfies all the constraints in φ. The satisfiability problem for Γ-formulas is denoted by CSP(Γ). Assuming a canonical order on the variables, we can regard assignments as tuples in the obvious way, and say that a quantifier-free first-order formula *defines* or *expresses* the logical relation of its solutions. We say that two quantifier-free first-order formulas φ and ψ are equivalent ($\varphi \equiv \psi$) if they have the same sets of variables and solutions.

For single-element constraint languages $\{R\}$, we often omit parenthesis and speak about R-formulas instead of $\{R\}$-formulas, etc. We also write $(0^n, 1^n)$ for the tuple $(\underbrace{0,\ldots,0}_{n},\underbrace{1,\ldots,1}_{n})$, etc.

A *balanced assignment* for a Γ-formula φ is a truth assignment I that assigns 0 to the same number of variables as 1, that means it fulfills $|\{x \in \mathrm{Var}(\varphi) \mid I(x) = 0\}| = |\{x \in \mathrm{Var}(\varphi) \mid I(x) = 1\}|$. We study here the two following problems.

Problem:	BAL-CSP(Γ)
Input:	A Γ-formula φ
Question:	Is there a balanced assignment that satisfies φ ?

Problem:	K-ONES(Γ)
Input:	A Γ-formula φ and a number $k \in \mathbb{N}$
Question:	Is there a truth assignment setting exactly k variables to true that satisfies φ?

Additionally we look at the counting version associated with each of these problems, i.e., the question of how many "acceptable" (balanced / with k ones) satisfying truth assignments a given Γ-formula has. These counting problems are denoted by #BAL-CSP(Γ) and #K-ONES(Γ).

In order to state our complexity results for counting problems, we briefly explain the reductions used in this context. Completeness in the complexity class #P is obtained through a notion of reducibility that is more general than the usual many-one reducibility. Let G, H be two counting problems, which we simply consider as functions from Σ^* to $\mathbb{N}$ for some alphabet Σ. The counting problem G is *polynomial-time Turing reducible* to H if there is a Turing machine with an oracle for H that computes G in polynomial time. A counting problem G is #P-*hard* if every counting problem in #P is polynomial-time Turing reducible to G; if in addition G is in #P, then it is #P-*complete*.

It is worth noticing that all the reductions exhibited in this paper between constraint satisfaction problems are stricter than Turing reductions, they are namely either parsimonious or counting reductions. We say that G reduces to H with a logspace-computable *parsimonious reduction*, (we write $G \leq_!^{\log} H$) if there is a logspace-computable function $f\colon \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$, $G(x) = H(f(x))$. (If $G \leq_!^{\log} H$ and $H \leq_!^{\log} G$, we write $G \equiv_!^{\log} H$.) Note that if we denote with G^d and H^d the decision versions of G, resp. H (i.e., the problems to determine whether $G(x) > 0$, resp. $H(x) > 0$ for input x), then $G \leq_!^{\log} H$ immediately implies that $G^d \leq_m^{\log} H^d$, with the same reduction function f. We often use this without explicit reference. Parsimonious reductions are the natural generalization of many-one reductions to counting problems. A less strict notion is that of a *counting reduction*, as defined by Viktória Zankó [Zankó 1991]. A logspace counting reduction consists of two logspace-computable functions f_1 and f_2. This pair forms a counting reduction from G to H (written $G \leq_c^{\log} H$), if for all x, it holds that $G(x) = f_1(H(f_2(x)))$. In particular, this holds for a transformation f such that $H(f(x)) = 2 \cdot G(x)$, as will be the case in all of our non-parsimonious counting reductions. It is immediate that both counting and parsimonious reductions are transitive, are special cases of Turing reductions, and that a parsimonious reduction is also a counting reduction. It should be noted that although the step from parsimonious to counting reductions seems subtle, there is a significant difference: Unlike parsimonious reductions, counting reductions do not close the classes of the counting hierarchy, unless the counting hierarchy collapses [Toda and Watanabe 1992].

DEFINITION 2.1. *A logical relation R is* affine with width 2 *if it is definable by a conjunction of equations, each of which being either a unary clause or a 2XOR-clause, that is of the form $l_1 \oplus l_2$, where l_1, l_2 are literals and $\oplus$ is the exclusive-or operator. A constraint language Γ is affine with width 2 if every relation in Γ is affine with width 2.*

The following is our main result:

THEOREM 2.2. *Let Γ be a constraint language.*

—*If Γ is affine with width 2, then* BAL-CSP(Γ) *(respectively,* K-ONES(Γ)*) is decidable in polynomial time. Otherwise it is* NP*-complete.*

—*If Γ is affine with width 2, then* #BAL-CSP(Γ) *(respectively,* #K-ONES(Γ)*) is computable in polynomial time. Otherwise it is* #P*-complete.*

Observe that there is an immediate parsimonious reduction from #BAL-CSP(Γ) to #K-ONES(Γ). It therefore suffices to prove polynomial-time results only for the problems K-ONES and #K-ONES, and hardness results for the problems BAL-CSP and #BAL-CSP. The polynomial side of the theorem is rather easy to prove. As suggested in [Bazgan and Karpinski 2005], if Γ is affine with width 2, then BAL-CSP(Γ) can be efficiently decided by dynamic programmimg. We show that the technique can be extended to solve the counting problem #K-ONES(Γ) in polynomial time.

PROPOSITION 2.3. *Let Γ be a constraint language that is affine with width 2. Given a Γ-formula φ and an integer k, one can compute the number of truth assignments that satisfy φ and set exactly k variables to true in polynomial time.*

Proof. Let φ be Γ-formula over the set of variables $\{x_1, \ldots, x_n\}$ and k an integer with $k \leq n$. Since Γ is affine with width 2, the formula φ can be written as a conjunction of unary and 2XOR-clauses. Thus every constraint in φ expresses equalities between literals (or between a literal and a constant). Hence, let $G(\varphi)$ be the equality graph defined over the set of vertices $\{0, 1, x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ such that: for every unary clause (l) in φ there are two edges $\{l, 1\}$ and $\{\bar{l}, 0\}$, and for every 2XOR-clause $(l_1 \oplus l_2)$ there are two edges $\{l_1, \bar{l}_2\}$ and $\{\bar{l}_1, l_2\}$. Connected components in $G(\varphi)$ (which can be computed in polynomial time) correspond to equivalence classes among literals. Observe that by construction every connected component C in $G(\varphi)$ has a dual one $\bar{C} = \{\bar{l} \mid l \in C\}$, and that φ is satisfiable if and only if no connected component in $G(\varphi)$ is equal to its dual. Suppose φ is satisfiable. Let C_0 (resp., C_1) be the connected component containing 0 (resp., 1). By construction when $\{l, 1\}$ is an edge of $G(\varphi)$, so is $\{\bar{l}, 0\}$. Therefore, the components C_0 and C_1 are dual to each other. Let $C_2, \bar{C}_2, \ldots, C_t, \bar{C}_t$ be the other connected components. A satisfying assignment of φ is obtained by assigning 1 to all the literals in C_1 and a same value to all the literals occurring in a same connected component $C_2, \ldots, C_t$. Let α be the number of positive literals in C_1. For $i =$

$2,\ldots,t$, let a_i be the number of positive literals in C_i and b_i the number of positive literals in $\bar{C}_i$. Counting the number of satisfying assignments of φ that set exactly k variables to 1 comes down to count the number of tuples $(c_2,\ldots,c_t)$ such that, $c_j = a_j$ or $c_j = b_j$ for every $j = 2,\ldots,t$, and $\sum_{j=2}^{t} c_j = k-\alpha$. This last problem can be solved by examining $w(i,L)$, the number of tuples $(c_2,\ldots,c_i)$ as above whose sum of elements is exactly L, for $i = 2,\ldots,t$ and $L = 1,\ldots k-\alpha$. Observe that $w(i+1,L) = w(i,L-a_{i+1}) + w(i,L-b_{i+1})$ (with the convention $w(i,X) = 0$ for $X < 0$) and that $w(2,L) = 2$ if $L = a_2 = b_2$, 1 if $L = a_2$ or $L = b_2$ and $a_2 \neq b_2$, and 0 otherwise. Therefore, since all integers occurring here can be encoded in unary (they correspond to number of variables in the formula given in input) the quantity we are interested in, namely $w(t,k-\alpha)$, can be computed dynamically in polynomial time. Thus we have designed a polynomial-time algorithm to count the number of satisfying assignments of an affine with width 2 formula that set exactly k variables to true. □

3. THE WEAK BASE METHOD

Since Proposition 2.3 covers the only polynomial-time case of our result, it remains to prove NP and #P-completeness of the remaining problems. We now introduce the algebraic tools that our hardness proof relies on. A detailed example will be provided at the end of the section. For more background on these notions, see [Schnoor and Schnoor 2008]. Also note that the technique is not limited to Boolean domains, however in order to simplify notation, we only state the results for this case.

DEFINITION 3.1. *Let Γ be a set of logical relations.*

—$\langle\Gamma\rangle$ *is the set of relations which can be expressed as a formula of the form $\exists x_1 \ldots \exists x_k \varphi$, where φ is a $(\Gamma \cup \{=\})$-formula as defined above in which (among others) the variables $x_1,\ldots,x_k$ appear.*

—$\langle\Gamma\rangle_{\nexists}$ *is the set of relations which can be expressed as a $(\Gamma \cup \{=\})$-formula.*

—$\langle\Gamma\rangle_{\nexists,\neq}$ *is the set of relations which can be expressed as a Γ-formula.*

If $R \in \langle\Gamma\rangle_{\nexists}$, we also say that Γ can *implement* R. The main reason why these closure operators are relevant for us is the following: Assume that $\Gamma_1 \subseteq \langle\Gamma_2\rangle$. Then a Γ_1-formula can be transformed into a satisfiability-equivalent Γ_2-formula (here we can simply drop all existential quantifiers introduced by the transformation, as in a satisfiability-problem, all variables are implicitely quantified existentially). Thus, it has been proved that $\textsc{Csp}(\Gamma_1)$ can be reduced in logarithmic space to $\textsc{Csp}(\Gamma_2)$ (see [Jeavons 1998; Allender et al. 2005]). Hence the complexity of $\textsc{Csp}(\Gamma)$ depends only on $\langle\Gamma\rangle$. The set $\langle\Gamma\rangle$ is a relational clone (or a *co-clone*). Accordingly, in order to obtain a full complexity classification for the satisfiability problem we only have to study the co-clones. Interestingly, there exists a Galois correspondence between the lattice of Boolean relations (co-clones) and the lattice of Boolean functions (clones) (see [Geiger 1968; Bodnarchuk et al. 1969]). This correspondence is established through the operators Pol(.) and Inv(.) defined below.

DEFINITION 3.2. Let $f\colon \{0,1\}^m \to \{0,1\}$ and $R \subseteq \{0,1\}^n$. We say that f is a *polymorphism* of R, if for all $x_1,\ldots,x_m \in R$, where $x_i = (x_i[1], x_i[2],\ldots,x_i[n])$, we have $\big(f\big(x_1[1],\cdots,x_m[1]\big), f\big(x_1[2],\cdots,x_m[2]\big),\ldots,f\big(x_1[n],\cdots,x_m[n]\big)\big) \in R$.

If $f \in \text{Pol}(R)$, we also say that R is *closed under* f, or f *preserves* R. For a set of relations Γ we write $\text{Pol}(\Gamma)$ to denote the set of all polymorphisms of Γ, i.e., the set of all Boolean functions that preserve every relation in Γ. For every Γ, $\text{Pol}(\Gamma)$ is a *clone*, i.e., a set of Boolean functions that contains all projections and is closed under composition. The smallest clone containing a set B of Boolean functions will be denoted by $[B]$ in the sequel (B is also called a *basis* for $[B]$). Important Boolean clones are M, containing the *monotone functions* (f is monotone if $\alpha_1 \leq \beta_1, \ldots, \alpha_n \leq \beta_n$ implies $f(\alpha_1, \ldots, \alpha_n) \leq f(\beta_1, \ldots, \beta_n)$), and for $i \in \{0, 1\}$ the clone R_i containing the i-reproducing functions (f is i-reproducing if $f(i, \ldots, i) = i$). For a set B of Boolean functions, let $\text{Inv}(B)$ denote the set of all *invariants* of B, i.e., the set of all Boolean relations that are preserved by every function in B. It can be observed that each $\text{Inv}(B)$ is a relational clone. As shown first in [Geiger 1968; Bodnarchuk et al. 1969] the operators $\text{Pol}(.) - \text{Inv}(.)$ constitute a Galois correspondence between the lattice of sets of Boolean relations and the lattice of sets of Boolean functions. In particular for every set Γ of Boolean relations and for every set B of Boolean functions $\text{Inv}(\text{Pol}(\Gamma)) = \langle \Gamma \rangle$ and $\text{Pol}(\text{Inv}(B)) = [B]$. Thus, there is a one-to-one correspondence between clones and co-clones, and hence we may compile a full list of co-clones from the list of clones obtained by Emil Post in [Post 1941]. The list of all Boolean clones with finite bases can be found e.g. in [Böhler et al. 2003]. A compilation of all co-clones with simple bases is given in [Böhler et al. 2005]. In the following, when discussing about bases for clones or co-clones we implicitly refer to these two lists, bases for some of the most relevant clones for the current paper are given in table 2. Figure 1 provides a representation of the inclusion structure of the clones, and hence also of the co-clones: For two clones $\mathcal{C}_1$ and $\mathcal{C}_2$, it holds that $\text{Inv}(\mathcal{C}_1) \subseteq \text{Inv}(\mathcal{C}_2)$ if and only if $\mathcal{C}_2 \subseteq \mathcal{C}_1$.

Unfortunately, cardinality constraints apparently make it difficult to use the Galois connection explained above. Indeed, existential variables and equality constraints that may occur when transforming a Γ_1-formula into a satisfiability-equivalent Γ_2-formula are problematic, as they can change the set of solutions. Therefore for these problems we prefer to consider the above-mentioned restricted closure $\langle . \rangle_{\nexists,\neq}$, which allows to translate formulas into equivalent ones. Since for equivalent formulas, obviously the answers to all four problems that we consider in this paper are the same, the closure operator $\langle . \rangle_{\nexists,\neq}$ directly induces reductions for our problems. Note that although our main result only proves hardness of the involved counting problems for Turing reductions, the reductions between constraint languages generating the same $\langle . \rangle_{\nexists,\neq}$-closure are in fact parsimonious. We obtain the following proposition:

PROPOSITION 3.3. *Let Γ_1 and Γ_2 be constraint languages with $\Gamma_1 \subseteq \langle \Gamma_2 \rangle_{\nexists,\neq}$. Then*

—BAL-CSP(Γ_1) $\leq_m^{\log}$ BAL-CSP(Γ_2) *and* K-ONES(Γ_1) $\leq_m^{\log}$ K-ONES(Γ_2),

—#BAL-CSP(Γ_1) $\leq_!^{\log}$ #BAL-CSP(Γ_2) *and* #K-ONES(Γ_1) $\leq_!^{\log}$ #K-ONES(Γ_2),

The main strategy to prove that for some class of constraint languages, the problems we consider are NP- or #P-hard is to prove that there is a constraint language Γ such that for every language Γ' in the studied class, $\langle \Gamma' \rangle_{\nexists,\neq}$ contains Γ, and the problem is hard for Γ. Proposition 3.3 then implies the result for every

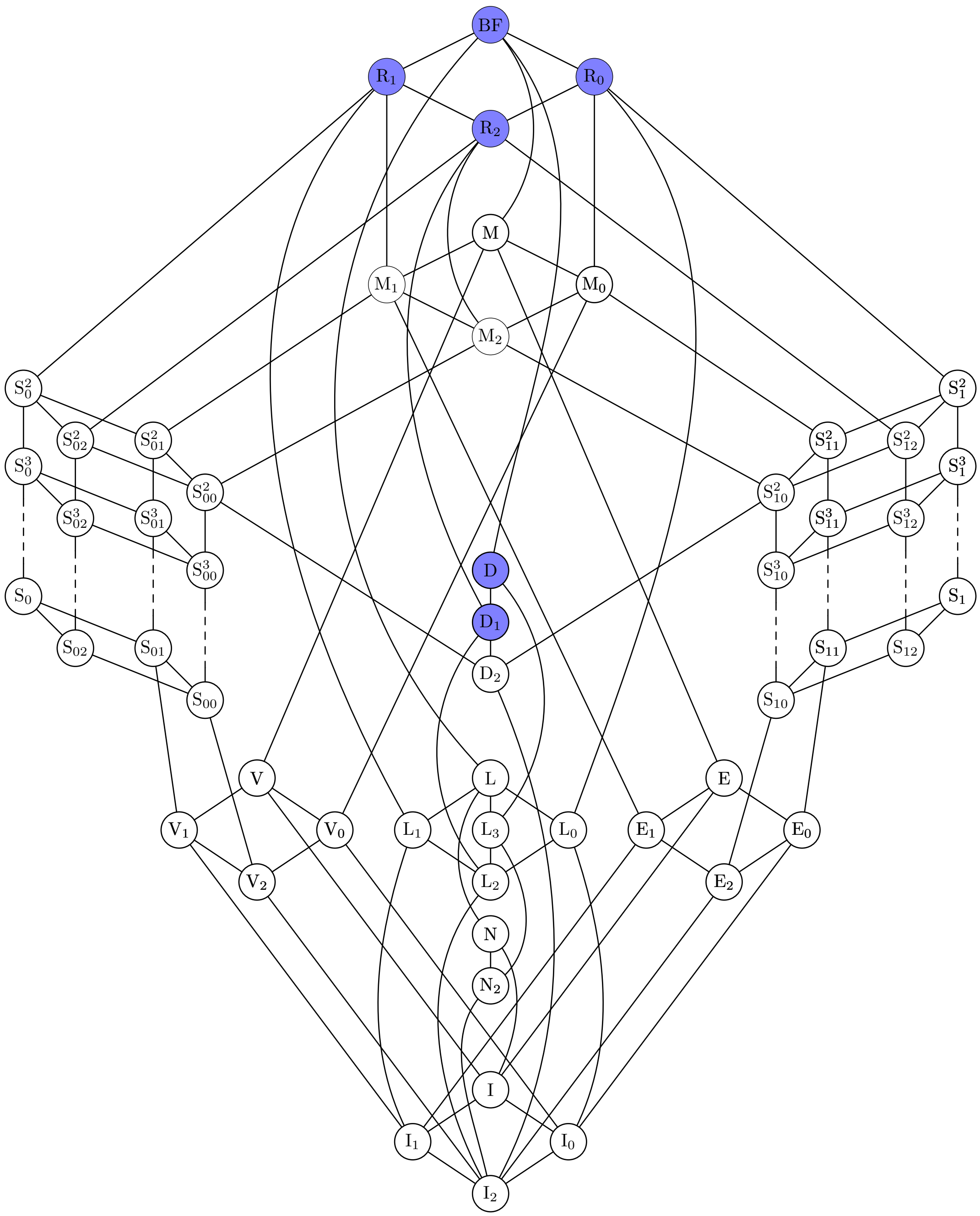

Fig. 1. Post's lattice

language in the class. In [Schnoor and Schnoor 2008], techniques were established that allow to prove results in this direction. We briefly explain the main definitions and results. The main tool is the notion of a *weak base*. A weak base is a constraint language that has the properties of the language Γ in the above description. Note that in the aforementioned paper, a different (but proven to be equivalent) definition was given.

DEFINITION 3.4 [SCHNOOR AND SCHNOOR 2008]. *Let $\mathcal{C}$ be a clone. A* weak base *for* $\mathrm{Inv}(\mathcal{C})$ *is a constraint language* Γ *such that: (i)* $\langle\Gamma\rangle = \mathrm{Inv}(\mathcal{C})$*, (ii) for any constraint language* Γ' *with* $\langle\Gamma'\rangle = \mathrm{Inv}(\mathcal{C})$*, it follows that* $\Gamma \subseteq \langle\Gamma'\rangle_{\nexists}$.

We also say that a relation R is a weak base for $\mathrm{Inv}(\mathcal{C})$ if $\{R\}$ is a weak base. Since for the balanced satisfiability problem, we need to consider the stricter closure operator $\langle.\rangle_{\nexists,\neq}$, we need an additional technical notion. In the following, we consider relations as matrices, where the rows of the matrix correspond to the tuples of the relation (technically, for uniqueness, we need to fix an order on the rows, for example lexicographical ordering). An n-ary relation R is *irredundant* if R, considered as a matrix, does not contain two identical columns, and if there is no i, $1 \leq i \leq n$, such that the value of the i^{th} variable is unconstrained. A set of relations Γ is irredundant if every relation in Γ is irredundant. The motivation for defining irreduntant relations is that one can easily represent these relations with formulas in which no equality clauses appear (as is obvious from the definition). Hence if we can express an irreduntant relation using the $\langle.\rangle_{\nexists}$-operator, we immediately obtain a representation using the $\langle.\rangle_{\nexists,\neq}$ as well. This (together with the above properties of weak bases) directly implies the following propsition:

PROPOSITION 3.5 [SCHNOOR AND SCHNOOR 2008]. *Let* Γ' *be a constraint language.*

(1) *If* $R \in \langle\Gamma'\rangle_{\nexists}$ *is an irredundant relation, then* $R \in \langle\Gamma'\rangle_{\nexists,\neq}$.
(2) *Let* Γ *be an irredundant weak base for a co-clone* $\mathrm{Inv}(\mathcal{C})$*. If* $\langle\Gamma'\rangle = \mathrm{Inv}(\mathcal{C})$*, then* $\Gamma \subseteq \langle\Gamma'\rangle_{\nexists,\neq}$.

The above proposition, together with Proposition 3.3 implies that to prove hardness results for the balanced constraint satisfaction problem, it is helpful to be able to construct weak bases for the involved coclones. Therefore, we now briefly explain how to construct weak bases. For a set of Boolean functions $\mathcal{F}$, the *$\mathcal{F}$-closure* of a relation R, denoted by $\mathcal{F}(R)$, is the minimal superset of R that is closed under every function from $\mathcal{F}$. This relation can be obtained from R by repeatedly applying all functions from $\mathcal{F}$, and adding the result to R. We say R is an *$\mathcal{F}$-core* of $\mathcal{F}(R)$.

DEFINITION 3.6 [SCHNOOR AND SCHNOOR 2008]. *Let $\mathcal{C}$ be a clone and s be a positive integer. Then, s is a* core-size *for* $\mathrm{Inv}(\mathcal{C})$ *if there is a relation R such that* $\langle R\rangle = \mathrm{Inv}(\mathcal{C})$ *and R has a $\mathcal{C}$-core with cardinality s.*

As we will see in the following theorem, knowing the core-size for a coclone directly allows us to construct a weak base for it, using the following construction: The relation COLS_s is defined to be the 2^s-ary relation of cardinality s such that the columns of COLS_s contain every possible s-ary binary vector (in lexicographical order). In the following, by $\mathrm{COLS}_s(l,-)$ we denote the l-th row vector of COLS_s, and by $\mathrm{COLS}_s(-,k)$ its k-th column vector.

Theorem 3.7 [Schnoor and Schnoor 2008]. *Let $\mathcal{C}$ be a clone and s be a core-size for* $\operatorname{Inv}(\mathcal{C})$*. Then the relation* $\mathcal{C}\,(\mathrm{COLS}_s)$ *is a weak base of* $\operatorname{Inv}(\mathcal{C})$*, and* $s+1$ *is a core-size for* $\operatorname{Inv}(\mathcal{C})$ *as well.*

With this theorem one can construct weak bases for all Boolean co-clones for which we know finite bases (since finite bases give us core-sizes). This fits our purpose. Indeed, there are only 8 clones that have no finite basis, namely S_0, S_{01}, S_{02}, S_{00} and S_1, S_{11}, S_{12}, S_{10}. These clones $\mathcal{C}$ are exactly the ones for which there exists no finite constraint language Γ such that $\langle\Gamma\rangle = \operatorname{Inv}(\mathcal{C})$ (see [Böhler et al. 2005]), and therefore will not be involved in our study. Also note that the relation $\mathcal{C}\,(\mathrm{COLS}_s)$ is constructed by repeatedly applying all functions from $\mathcal{C}$ to COLS_s, but it can easily be seen that every tuple in $\mathcal{C}\,(\mathrm{COLS}_s)$ can be obtained by applying an s-ary function from $\mathcal{C}$ to COLS_s. We summarize the main properties of the weak base method that we need for the study of our cardinality constraint problems. Note that the following result is not limited to this context—in fact, it is true for every problem for which an analogous statement to Proposition 3.3 can be proven, in particular for any problem which is invariant under exchanging an input formula with an equivalent one. In most cases, we will apply the corollary for the case that Γ' contains only a single relation, and often for the case $\Gamma' = \{R\}$.

Corollary 3.8. *Let Γ be a constraint language, let s be a core-size for $\langle\Gamma\rangle$, let* $R = \operatorname{Pol}(\Gamma)\,(\mathrm{COLS}_s)$*, and let Γ' be an irredundant constraint language such that* $\Gamma' \subseteq \langle R\rangle_{\nexists}$*. Then*

(1) Bal-Csp$(\Gamma') \leq_m^{\log}$ Bal-Csp(Γ),

(2) #Bal-Csp$(\Gamma') \leq_!^{\log}$ #Bal-Csp(Γ).

Proof. It follows from Theorem 3.7 that R is a weak base for $\langle\Gamma\rangle$. Therefore $R \in \langle\Gamma\rangle_{\nexists}$, and since $\langle.\rangle_{\nexists}$ is a closure operator, it follows that $\langle R\rangle_{\nexists} \subseteq \langle\Gamma\rangle_{\nexists}$. Since $\Gamma' \subseteq \langle R\rangle_{\nexists}$, it follows that $\Gamma' \subseteq \langle\Gamma\rangle_{\nexists}$, and since Γ' is irredundant, Proposition 3.5 implies that $\Gamma' \subseteq \langle\Gamma\rangle_{\nexists,\neq}$. The reduction now follows from Proposition 3.3. □

We now present an extended example to explain the constructions given in this section. For this and for the proofs in the sequel, we will be working with some specific relations, which we define now: $C_0 = \{(0)\}$, $C_1 = \{(1)\}$, 1-in-3 $= \{(0,0,1),(0,1,0),(1,0,0)\}$, $\mathrm{Imp} = \{(0,0),(0,1),(1,1)\}$, $\mathrm{Or}^2 = \{(0,1),(1,0),(1,1)\}$, $\mathrm{Odd}^2 = \{(0,1),(1,0)\}$ and $\mathrm{Odd}^3 = \{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}$.

Example 3.9. Consider the clone N, which is generated by the set of Boolean functions $B = \{\neg, 0\}$, where $\neg$ is negation, and 0 is the constant 0-function. From [Böhler et al. 2005], it is known that the relation $\mathrm{Dup} = \{0,1\}^3 \setminus \{(0,1,0),(1,0,1)\}$ is a base for the co-clone Inv(N), i.e., Pol(Dup) = N. Now consider the relation $R = \{(0,0,1),(0,1,1)\}$. Then, the relation N(R) is obtained from R by repeatedly adding tuples to R that can be obtained by applying functions from N to tuples already in the relation (it is easy to see that it suffices to consider functions from the base B of the clone). Since $0 \in B$, we first have to add the tuple $(0,0,0)$. Since negation is in B, we have to add the coordinate-wise negation of the three tuples obtained so far, i.e., we add $(1,1,0)$, $(1,0,0)$, and $(1,1,1)$. The relation obtained in this way is Dup (as mentioned above, Dup is closed under N, and it is easy to see that it is indeed closed under B, so we do not need to add further tuples). Therefore, R is an N-core of Dup, and hence 2 is a core-size of Inv(N).

From Theorem 3.7, it now follows that $\mathrm{N}(\mathrm{COLS}_2)$ is a weak base of $\mathrm{Inv}(N)$. The relation COLS_2 contains as columns the binary numbers from 0 to 3, i.e.,

$$\mathrm{COLS}_2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

$\mathrm{N}(\mathrm{COLS}_2)$ is now obtained from COLS_2 in the same way as in the construction of $\mathrm{N}(R)$ above, i.e., we add to COLS_2 all tuples that we can generate by repeatedly applying functions from N (again, it is enough to consider functions from B, since B generated N). The resulting relation is

$$R' = \mathrm{N}(\mathrm{COLS}_2) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Here the second and third lines are the tuples from COLS_2, the first tuple is a result of the application of the constant 0-function, and tuples $4-6$ are the negations of the first three tuples. Note that R' is irredundant.

The statement that $\mathrm{N}(\mathrm{COLS}_2)$ is an irredundant weak base of Inv(N) means that for every constraint language Γ with $\mathrm{Pol}(\Gamma) = \mathrm{N}$ (which is equivalent to saying $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{N})$), there is a Γ-formula φ equivalent to $R'(w, x, y, z)$. In particular, this must hold for the constraint language $\Gamma = \{\mathrm{Dup}\}$, and in fact it can be verified that the formula $\varphi(w, x, y, z) = \mathrm{Dup}(w, y, z) \wedge \mathrm{Dup}(w, x, z) \wedge \mathrm{Dup}(x, z, y) \wedge \mathrm{Dup}(x, w, y)$ is in fact equivalent to $R'(w, x, y, z)$.

It is worth noting that we only get this "implementation result" for constraint languages that generate *exactly* the co-clone Inv(N). For example, consider the relation 1-in-3. It is known that $\mathrm{Inv}(\mathrm{N}) \subseteq \langle\text{1-in-3}\rangle$, but clearly, there is no 1-in-3-formula that is equivalent to $R(w, x, y, z)$: If φ is an 1-in-3-formula and I a solution for φ, then the negation of I is no solution for φ. Therefore 1-in-3 cannot be used to express $R'(w, x, y, z)$, whose set of solutions is closed under negation (note however that due to the above, we *can* express the relation R' using 1-in-3 if we may additionally use existential quantifiers).

4. PROOFS OF HARDNESS RESULTS

We now prove the hardness part of our main Theorem 2.2: We show that if a constraint language is not affine with width 2, then Bal-Csp(Γ) is NP-complete, and #Bal-Csp(Γ) is #P-complete. As mentioned earlier, this trivially implies the hardness results for k-Ones(Γ) and #k-Ones(Γ). The proof heavily relies on the algebraic tools described in Section 3, and follows the general idea as mentioned in the discussion following Proposition 3.3. In order to apply these algebraic techniques, we first show that our main theorem can be restated in terms of clones and coclones; this is possible since constraint languages which are affine with width 2 are exactly those contained in the coclone $\mathrm{Inv}(\mathrm{D}_1)$. Hence the main work to be done is to show, for every coclone which is not a subset of $\mathrm{Inv}(\mathrm{D}_1)$, that it has a weak base for which our problems are hard. Using the results of Sect ion 3, this implies the hardness part of our main theorem.

4.1 Another Statement of The Main Result

As discussed above we do not know how to use the Galois correspondence between relational co-clones and Post's classes for our problems. However, it turns out that the complexity classification, when achieved, obeys the border among co-clones (see Figure 1). The clones corresponding to our polynomial-time cases are highlighted in Figure 1. The following is a reformulation of our main Theorem 2.2 in terms of coclones:

THEOREM 4.1. *Let* Γ *be a constraint language.*

—*If* $\Gamma \subseteq \mathrm{Inv}(\mathrm{D}_1)$*, then* BAL-CSP($\Gamma$) *(respectively,* K-ONES(Γ)*) is decidable in polynomial time. Otherwise it is* NP*-complete.*

—*If* $\Gamma \subseteq \mathrm{Inv}(\mathrm{D}_1)$*, then* #BAL-CSP($\Gamma$) *(respectively,* #K-ONES(Γ)*) is computable in polynomial time. Otherwise it is* #P*-complete.*

Our main theorem can be reformulated as above since it is well known that $\mathrm{Inv}(\mathrm{D}_1)$ is the set of all affine with width two relations (see e.g., [Creignou et al. 2008]). Thus, Theorem 2.2 will be proved by an exhaustive examination of the clones in Post's lattice. As mentioned before, it suffices to prove hardness results for the balanced versions of our problems. These proofs are organized as follows: In Section 4.2, we prove hardness for a set of individual relations. Section 4.3 uses the algebraic techniques recalled in Section 3 to extend these results to all constraint languages generating specific co-clones, without the need to construct a concrete weak base for these. Section 4.4 then contains the results for the remaining co-clones, where the proof requires to study individual weak bases. For counting problems we will use known hardness results. Note that all these results we refer to were obtained under parsimonious or Turing reductions.

4.2 Basic Hardness Results

First we will take advantage of the symmetry of Post's lattice. The dual relation of a relation R, is given by $\mathrm{dual}(R) = \{(1-a_1, \ldots, 1-a_n) \ : \ (a_1, \ldots, a_n) \in R\}$. If Γ is a set of relations, then $\mathrm{dual}(\Gamma) = \{\mathrm{dual}(R) \ : \ R \in \Gamma\}$.

PROPOSITION 4.2. *For any constraint language* Γ*,* #BAL-CSP($\mathrm{dual}(\Gamma)$) $\equiv_!^{\log}$ #BAL-CSP(Γ) *and* BAL-CSP($\mathrm{dual}(\Gamma)$) $\equiv_m^{\log}$ BAL-CSP(Γ).

Proof. It suffices to prove the parsimonious reduction, and since $\mathrm{dual}(\mathrm{dual}(\Gamma)) = \Gamma$, it is also enough to show that #BAL-CSP(Γ) $\leq_!^{\log}$ #BAL-CSP($\mathrm{dual}(\Gamma)$). Let φ be a Γ-formula and let φ' be the $\mathrm{dual}(\Gamma)$-formula obtained by replacing every clause $R(x_1, \ldots, x_n)$ from φ by $\mathrm{dual}(R)(x_1, \ldots, x_n)$. Clearly, an assignment I satisfies φ if and only if its negation I' satisfies φ'. Since I is balanced if and only if I' is, this establishes the reduction. ☐

As mentioned, we now start with proving hardness results for some of the relations introduced at the end of Section 3.

LEMMA 4.3. BAL-CSP(Imp) *is* NP-*hard and* #BAL-CSP(Imp) *is* #P-*hard.*

Proof. We first prove NP-hardness of BAL-CSP(Imp). We reduce from the following problem:

Problem:	K-CLOSURE
Input:	a directed graph $G = (V, E)$ and $k \in \mathbb{N}$
Question:	Is there a k-closure, i.e., a set $V' \subseteq V$ such that $\lvert V'\rvert = k$ and there is no edge $(u, v) \in E$ with $u \notin V'$ and $v \in V'$?

Due to [Garey and Johnson 1979], K-CLOSURE is NP-complete. We show K-CLOSURE $\leq_m^{\log}$ BAL-CSP(Imp). Let $G = (V, E)$ be a directed graph and $k \in \mathbb{N}$. Let $n = \lvert V\rvert$. We construct an Imp-formula with variables $X = V \cup \{t_1, \ldots, t_k, f_1, \ldots, f_{n-k}\}$, where $t_1, \ldots, t_k, f_1, \ldots, f_{n-k}$ are all distinct variables and not from V. We set

$$\varphi = \bigwedge_{(u,v)\in E} \mathrm{Imp}(u, v) \wedge \bigwedge_{i=1}^{k} \bigwedge_{x\in X} \mathrm{Imp}(x, t_i) \wedge \bigwedge_{i=1}^{n-k} \bigwedge_{x\in X} \mathrm{Imp}(f_i, x).$$

We show that (G, k) is a positive instance of K-CLOSURE if and only if φ has a balanced solution. First assume that (G, k) is a positive instance, and let $V' \subseteq V$ such that $\lvert V'\rvert = k$ and for every $(u, v) \in E$ it holds that $u \in V'$ or $v \notin V'$. It is easy to see that the assignment $I\colon X \to \{0, 1\}$ defined by $I(x) = 0$ if $x \in V' \cup \{f_1, \ldots, f_{n-k}\}$ and $I(x) = 1$ otherwise is a balanced solution for φ.

Now let $I\colon X \to \{0, 1\}$ be a balanced solution for φ. Assume $I(t_i) = 0$ for an $i \in \{1, \ldots, k\}$. Then, because of the clauses $\bigwedge_{x\in X} \mathrm{Imp}(x, t_i)$, it follows that $I(x) = 0$ for every $x \in X$. This contradicts the fact that I is balanced, therefore $I(t_i) = 1$ for every $i \in \{1, \ldots, k\}$ and analogously $I(f_i) = 0$ for every $i \in \{1, \ldots, n-k\}$. Because I is balanced, I maps k variables from V to 0 and $n-k$ variables from V to 1. Let $V' = \{x \in V \mid I(x) = 0\}$, then $\lvert V'\rvert = k$, and it is straightforward to verify that V' is a k-closure. Hence the reduction is complete, and BAL-CSP(Imp) is NP-hard.

To prove hardness of the counting problem, we show #CSP(Imp) $\leq_!^{\log}$ #BAL-CSP(Imp). The result then follows because #CSP(Imp) was shown to be #P-complete in [Creignou and Hermann 1996]. Let $\varphi = \mathrm{Imp}(x_1, y_1) \wedge \cdots \wedge \mathrm{Imp}(x_n, y_n)$ be an instance of #CSP(Imp), let $k = \lvert\mathrm{Var}(\varphi)\rvert$, and let $z_1, \ldots, z_k$ be new and distinct variables. We define $\varphi' = \varphi \wedge \bigwedge_{i=1}^{k-1} \mathrm{Imp}(z_i, z_{i+1})$. There obviously is a one-to-one correspondence between solutions for φ and balanced solutions for φ', since there is exactly one way to extend a solution of φ to a balanced solution of φ', and conversely, every (balanced) solution for φ' is a solution of φ as well. Note that CSP(Imp) is solvable in polynomial time, hence this reduction does not establish hardness of the decision problem. □

NP-hardness for BAL-CSP(Or^2) was proven in [Bazgan and Karpinski 2005], we add the #P-hardness for the counting problem.

LEMMA 4.4. BAL-CSP(Or^2) *is* NP-*hard and* #BAL-CSP(Or^2) *is* #P-*hard.*

Proof. In [Bazgan and Karpinski 2005] it was shown that BAL-CSP(dual(Or^2)) is NP-hard (dual(Or^2) is the binary NAND-relation), therefore due to Proposition 4.2 it also holds that BAL-CSP(Or^2) is NP-hard. For the #P-hardness of #BAL-CSP(Or^2) it is sufficient to show #CSP(Or^2) $\leq_!^{\log}$ #BAL-CSP(Or^2), because #CSP(Or^2) is #P-complete [Creignou and Hermann 1996]. Let φ be an Or^2-formula. For every $x \in \mathrm{Var}(\varphi)$ let x' be a new variable. We define

$$\varphi' = \varphi \wedge \bigwedge_{x \in \mathrm{Var}(\varphi)} \mathrm{Or}^2(x, x').$$

It is easy to see that an assignment $I\colon \mathrm{Var}(\varphi') \to \{0,1\}$ that satisfies the subformula $\bigwedge_{x\in\mathrm{Var}(\varphi)} \mathrm{Or}^2(x,x')$ is balanced if and only if for every $x \in \mathrm{Var}(\varphi)$ holds $I(x) \neq I(x')$. Therefore every solution $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ for φ can be extended to a balanced solution for φ' only in one way: by setting $I(x') \neq_{def} I(x)$ for every $x \in \mathrm{Var}(\varphi)$. Because obviously every balanced solution for φ' satisfies φ and is (due to the above) uniquely determined by its values for the variables appearing in φ, there is a one-to-one correspondence between solutions for φ and balanced solutions for φ'. Thus, #CSP(Or^2) $\leq_!^{\log}$ #BAL-CSP(Or^2). □

NP-hardness for BAL-CSP(Odd^3) was proven in [Bazgan and Karpinski 2005], we now prove hardness of the counting problem. Note that in the proof for the #P-hardness of #BAL-CSP(Odd^3) we prove an unusual "implementation result:" Using only Odd^3-clauses, we are able to simulate the bahaviour of 1-in-3-clauses, although the latter relation is far more expressive than Odd^3 (in fact, $\langle$1-in-3$\rangle$ is the set of all Boolean relations, while $\langle \mathrm{Odd}^3 \rangle$ only contains affine relations). This shows that adding cardinality constraints has a huge impact on the expressive power of constraint languages.

LEMMA 4.5. BAL-CSP(Odd^3) *is* NP-*hard and* #BAL-CSP(Odd^3) *is* #P-*hard.*

Proof. For the #P-hardness of the counting problem it is sufficient to show that #CSP(1-in-3) $\leq_!^{\log}$ #BAL-CSP(Odd^3), because #CSP(1-in-3) is hard for #P due to [Creignou and Hermann 1996]. Note that, since CSP(1-in-3) is an NP-complete problem [Schaefer 1978], the following reduction is also an alternative NP-hardness proof for BAL-CSP(Odd^3). Let $\varphi = \bigwedge_{i=1}^n \text{1-in-3}(x_i, y_i, z_i)$ be a 1-in-3-formula. We construct an Odd^3-formula using additionally to the variables appearing in φ the following new and distinct variables: a_i, b_i, c_i, d_i for every $1 \leq i \leq n$; t^i, f^i for every $1 \leq i \leq k$ where $k = 2\,|\mathrm{Var}(\varphi)| + 4n$; and v' for every $v \in \mathrm{Var}(\varphi)$. We define

$$\begin{aligned}
\varphi' = \bigwedge_{i=1}^{n} \big\{ & \mathrm{Odd}^3(x_i, y_i, z_i) \wedge \mathrm{Odd}^3(d_i, d_i, d_i) \wedge \\
& \mathrm{Odd}^3(d_i, x_i, a_i) \wedge \mathrm{Odd}^3(d_i, y_i, b_i) \wedge \mathrm{Odd}^3(d_i, z_i, c_i) \ \big\} \\
\wedge \ & \bigwedge_{i=1}^{k} \{\mathrm{Odd}^3(t^i, t^i, t^i) \wedge \mathrm{Odd}^3(t^i, f^i, f^1)\} \wedge \bigwedge_{v \in \mathrm{Var}(\varphi)} \mathrm{Odd}^3(f^1, v, v').
\end{aligned}$$

Note that $|\mathrm{Var}(\varphi')| = 2\,|\mathrm{Var}(\varphi)| + 4n + 2k = 3k$. Let $I\colon \mathrm{Var}(\varphi') \to \{0,1\}$ be a balanced solution for φ'. We show that I is uniquely determined by its values for

$\mathrm{Var}(\varphi)$. For every $1 \leq i \leq k$, it holds that $I(t^i) = 1$, because $\mathrm{Odd}^3(t^i, t^i, t^i)$ is a clause from φ', and $I(f^i) = I(f^1)$, because $\mathrm{Odd}^3(t^i, f^i, f^1)$ is a clause from φ'. Now assume $I(f^1) = 1$. Then $I(f^1) = \cdots = I(f^k) = I(t^1) = \cdots = I(t^k) = 1$. But since $2k > \frac{1}{2}|\mathrm{Var}(\varphi')| = \frac{3}{2}k$ this contradicts the prerequisite that I is balanced. Therefore we have $I(f^1) = \cdots = I(f^k) = 0$. With this the clauses $\mathrm{Odd}^3(f^1, v, v')$ give $I(v) \neq I(v')$ for every $v \in \mathrm{Var}(\varphi)$. So I is already balanced on $\mathrm{Var}(\varphi) \cup \{v' \mid v \in \mathrm{Var}(\varphi)\}$ and as well on the set $\{t^1, \ldots, t^k, f^1, \ldots, f^k\}$. It follows that I is also balanced on the rest of the variables of φ', i.e., on $\{a_1, b_1, c_1, d_1, \ldots, a_n, b_n, c_n, d_n\}$.

Since we have the clause $\mathrm{Odd}^3(d_i, d_i, d_i)$ for every $1 \leq i \leq n$, it holds that $I(d_1) = \cdots = I(d_n) = 1$. Therefore the clauses $\mathrm{Odd}^3(d_i, x_i, a_i)$, $\mathrm{Odd}^3(d_i, y_i, b_i)$, and $\mathrm{Odd}^3(d_i, z_i, c_i)$, give us $I(x_i) = I(a_i)$, $I(y_i) = I(b_i)$, and $I(z_i) = I(c_i)$. Thus, for every variable from φ' its value under I is uniquely determined by $I|_{\mathrm{Var}(\varphi)}$.

We now show that $I|_{\mathrm{Var}(\varphi)}$ is a solution for φ. Assume there is a clause 1-in-3(x_i, y_i, z_i) in φ that is not satisfied by $I|_{\mathrm{Var}(\varphi)}$. Since $\mathrm{Odd}^3(x_i, y_i, z_i)$ is a clause in φ' and $\mathrm{Odd}^3 = \text{1-in-3} \cup \{(1,1,1)\}$, it holds that $I(x_i) = I(y_i) = I(z_i) = 1$. Due to the above it follows $I(a_i) = I(b_i) = I(c_i) = I(d_i) = 1$. We showed above that I is balanced on $\{a_1, b_1, c_1, d_1, \ldots, a_n, b_n, c_n, d_n\}$, that means there exists a $j \in \{1, \ldots, n\}$ such that $I(a_j) + I(b_j) + I(c_j) + I(d_j) < 2$, otherwise we cannot compensate that $I(a_i) = I(b_i) = I(c_i) = I(d_i) = 1$. Because $I(d_j) = 1$ and $I(x_j) = I(a_j)$, $I(y_j) = I(b_j)$, and $I(z_j) = I(c_j)$ it follows that $I(x_j) + I(y_j) + I(z_j) = 0$, which is a contradiction to the fact that $\mathrm{Odd}^3(x_j, y_j, z_j)$ is a clause from φ'. Thus, every balanced solution for φ' restricted to $\mathrm{Var}(\varphi)$ is a solution for φ.

Now let $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ be a solution for φ. We prove that exactly one extension of I to $\mathrm{Var}(\varphi')$ is a balanced solution for φ'. It holds $I(x_i)+I(y_i)+I(z_i) = 1$ for every $i \in \{1, \ldots, n\}$, because 1-in-3(x_i, y_i, z_i) is a clause from φ. So, by setting $I(a_i) = I(x_i)$, $I(b_i) = I(y_i)$, $I(c_i) = I(z_i)$ and $I(d_i) = 1$, we extend I such that it is balanced on $\{a_1, b_1, c_1, d_1, \ldots, a_n, b_n, c_n, d_n\}$. If we extend I further according to the above by setting $I(v') = \neg I(v)$, $I(t_i) = 1$, and $I(f_i) = 0$ for every $v \in \varphi$ and every $1 \leq i \leq k$, we get a balanced solution for φ'. According to the discussion above, this is the only extension of I to $\mathrm{Var}(\varphi')$ that is a balanced solution for φ'. Hence, there is a one-to-one correspondence between solutions of φ and balanced solutions of φ'. We thus proved $\#\textsc{Csp}(\text{1-in-3}) \leq_!^{\log} \#\textsc{Bal-Csp}(\mathrm{Odd}^3)$, whic h completes the proof. □

4.3 Hardness Results with Unified Proofs

We now use the hardness results for individual relations obtained in the last section to prove hardness results for classes of constraint languages. The first theorem deals with constraint languages that have both the Boolean constant functions 0 and 1 as polymorphisms, but not the negation. In this case we prove that we can implement the implication relation, which corresponds to a basic hard case identified in the previous section.

Theorem 4.6. *Let Γ be a constraint language such that $\langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{I})$ and $\langle\Gamma\rangle \nsubseteq \mathrm{Inv}(\mathrm{N}_2)$. Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP*-hard and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P*-hard.*

Proof. We show $\mathrm{Imp} \in \langle\Gamma\rangle_{\nexists,\neq}$, then hardness for decision and counting problem follow from Lemma 4.3 and Proposition 3.3. Since $\langle\Gamma\rangle \nsubseteq \mathrm{Inv}(\mathrm{N}_2)$, and since N_2 is the clone generated by Boolean negation, negation is no polymorphism of Γ. That means there exists a relation R in Γ such that $\neg \notin \mathrm{Pol}(\Gamma)$, i.e., there is a $t \in R$ such that $\neg t \notin R$. Let $t = (t[1], \ldots, t[n])$, where n is the arity of R. Since $\langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{I})$, both constant Boolean functions are polymorphisms of Γ and in particular of R. Hence, $(0,\ldots,0),(1,\ldots,1) \in R$. Note that $t \notin \{(0,\ldots,0),(1,\ldots,1)\}$, otherwise $\neg t \in R$. One can verify that $\mathrm{Imp}(x_0,x_1) = R(x_{t[1]},\ldots,x_{t[n]})$. Hence, $\mathrm{Imp} \in \langle R\rangle_{\nexists} \subseteq \langle\Gamma\rangle_{\nexists}$. Since Imp is irredundant, it follows due to Proposition 3.5 that $\mathrm{Imp} \in \langle\Gamma\rangle_{\nexists,\neq}$, which concludes the proof. □

The main strategy of the remaining proofs is to apply the weak base method as summarized in Corollary 3.8. We show that for an appropriate core-size s of a co-clone $\langle\Gamma\rangle$, the weak base $R = \mathrm{Pol}(\Gamma)\,(\mathrm{COLS}_s)$ allows us to implement a relation that is "close to" one of those covered in Section 4.2. Hence, in the rest of this section we work with weak bases. However, for the cases covered here we do not need to compute any concrete weak base—instead, our results follow from proving that weak bases for the involved co-clones share the properties that we require to implement the necessary relations. The following theorem deals with constraint languages that generate one of the following co-clones: $\mathrm{Inv}(\mathrm{M}_1)$, $\mathrm{Inv}(\mathrm{V}_1)$, $\mathrm{Inv}(\mathrm{E}_1)$, $\mathrm{Inv}(\mathrm{S}_{01}^m)$ for any $m \in \mathbb{N}$. The main idea for the proof is to show that with these constraint languages, we can "almost implement" the relation Imp, for which we already proved that our problems are hard in Lemma 4.3.

THEOREM 4.7. *Let Γ be a constraint language such that* $\mathrm{Inv}(\mathrm{M}_1) \subseteq \langle\Gamma\rangle \subsetneq \mathrm{Inv}(\mathrm{I}_1)$*. Then* BAL-CSP($\Gamma$) *is* NP*-hard and* #BAL-CSP(Γ) *is* #P*-hard.*

Proof. Let T-Imp be the relation $\mathrm{C}_1 \times \mathrm{Imp}$. We prove hardness of BAL-CSP(T-Imp) and #BAL-CSP(T-Imp). Due to Lemma 4.3, it suffices to show #BAL-CSP(Imp) $\leq_!^{\log}$ #BAL-CSP(T-Imp). Let $\varphi = \mathrm{Imp}(x_1,y_1) \wedge \cdots \wedge \mathrm{Imp}(x_n,y_n)$ be an Imp-formula. We construct a T-Imp-formula φ' such that φ' has the same number of balanced solutions as φ has. Let t and f be new and distinct variables. We define:

$$\varphi' = \text{T-Imp}(t,x_1,y_1) \wedge \cdots \wedge \text{T-Imp}(t,x_n,y_n) \wedge \bigwedge_{x \in \mathrm{Var}(\varphi)} \text{T-Imp}(t,f,x).$$

Note that every solution for φ' must map t to 1. Furthermore every balanced solution for φ' must map f to 0 because otherwise 1 is assigned to all variables. Since every balanced solution $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ for φ extended by $I(t) = 1$ and $I(f) = 0$ is a balanced solution for φ', and conversely every balanced solution for φ' restricted to $\mathrm{Var}(\varphi)$ is a balanced solution for φ, it holds that φ has the same number of balanced solutions as φ'. Thus #BAL-CSP(Imp) $\leq_!^{\log}$ #BAL-CSP(T-Imp).

To show the result of the theorem, let s be a core-size of $\langle\Gamma\rangle$ and let $R = \mathrm{Pol}(\Gamma)\,(\mathrm{COLS}_s)$. Due to Theorem 3.7, we can assume that $s \geq 2$, and due to Corollary 3.8 and the above, it suffices to show that T-Imp $\in \langle R\rangle_{\nexists}$ (note that T-Imp is obviously irredundant). To prove this, we distinguish two cases: $\langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{V}_1)$ and $\langle\Gamma\rangle \nsubseteq \mathrm{Inv}(\mathrm{V}_1)$.

Case 1. $\langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{V}_1)$. Let S be the Boolean relation defined by

$$S(t,x,y) \equiv R(x, \underbrace{y,\ldots,y}_{2^{s-1}-1}, \underbrace{t,\ldots,t}_{2^{s-1}}).$$

We show $S = \text{T-Imp}$. Since $\langle\Gamma\rangle \subsetneq \text{Inv}(\text{I}_1)$, the constant 1-function is a polymorphism of Γ, and therefore of R. Hence, $(1,\ldots,1) \in R$ and $(1,1,1) \in S$. Because $\vee \in \text{V}_1 \subseteq \text{Pol}(\Gamma)$ it follows that the nested application of $\vee$ to all tuples of COLS_s is in R, i.e., $(0,1,\ldots,1) = \text{COLS}_s(1,-) \vee \cdots \vee \text{COLS}_s(s,-) \in R$.

To see that the disjunction of the above tuples gives the tuple $(0,1,\ldots,1)$, recall that the first column of COLS_s, contains only zeroes, and all other columns contain at least one entry with a 1. Hence the disjunction over all of these tuples gives the tuple that has a 0 in its first and a 1 in its remaining components. We therefore have shown $(0,1,\ldots,1) \in R$, and from the definition of S it follows that $(1,0,1) \in S$. Since

$$(\underbrace{0,\ldots,0}_{2^{s-1}}, \underbrace{1,\ldots,1}_{2^{s-1}}) = \text{COLS}_s(1,-) \in R,$$

it holds that $(1,0,0) \in S$, hence $\text{T-Imp} \subseteq S$. Note that $\text{Pol}(R)$ contains only functions which are monotone and 1-reproducing because $\text{Pol}(R) \subseteq \text{M}_1$, and M_1 contains exactly the Boolean functions with these two properties. Since $\text{COLS}_s(-,2^s) = (1,\ldots,1)$ and since all polymorphisms of Γ are 1-reproducing, it follows $R(-,2^s) = (1,\ldots,1)$ and therefore it holds for all $a,b \in \{0,1\}$ that $(0,a,b) \notin S$.

Finally assume $(1,1,0) \in S$. Then

$$u = (1, \underbrace{0,\ldots,0}_{2^{s-1}-1}, \underbrace{1,\ldots,1}_{2^{s-1}}) \in R.$$

Following the discussion after Theorem 3.7, there is an s-ary Boolean function $g \in \text{Pol}(\Gamma)$ such that $g(\text{COLS}_s(1,-),\ldots,\text{COLS}_s(s,-)) = u$. It holds that g is not monotone because $g(0,\ldots,0) = u[1] = 1$ and $g(0,\ldots,0,1) = u[2] = 0$. Since every function from $\text{Pol}(\Gamma)$ is monotone, this is a contradiction. Hence $(1,1,0) \notin S$, and from the case analysis over all possible tuples if follows that $\text{T-Imp} = S$ and therefore $\text{T-Imp} \in \langle R\rangle_{\nexists} \subseteq \langle\Gamma\rangle_{\nexists}$.

Case 2. $\langle\Gamma\rangle \nsubseteq \text{Inv}(\text{V}_1)$. In this case $\langle\Gamma\rangle = \text{Inv}(\text{E}_1)$ (see Figure 1). Let S be the Boolean relation defined by

$$S(t,x,y) \equiv R(\underbrace{x,\ldots,x}_{2^{s-1}}, \underbrace{y,\ldots,y}_{2^{s-1}-1}, t).$$

We show $S = \text{T-Imp}$. With the same arguments as in the first case it holds that $(1,1,1) \in S$. Because conjunction is an element of E_1, it follows that

$$(0,\ldots,0,1) = \text{COLS}_s(1,-) \wedge \cdots \wedge \text{COLS}_s(s,-) \in R,$$

which means $(1,0,0) \in S$ (the resulting tuple is $(0,\ldots,0,1)$, since in COLS_s, all columns except for the last one contain at least a single 0). Since

$$(\underbrace{0,\ldots,0}_{2^{s-1}},\underbrace{1,\ldots,1}_{2^{s-1}}) = \mathrm{COLS}_s(1,-) \in R,$$

it holds that $(1,0,1) \in S$, hence T-Imp $\subseteq S$. Again all functions from $\mathrm{Pol}(\Gamma)$ are monotone and 1-reproducing. So, using the same arguments as in the first case, we have for all $a,b \in \{0,1\}$ that $(0,a,b) \notin S$.

Finally assume $(1,1,0) \in S$. Then

$$u = (\underbrace{1,\ldots,1}_{2^{s-1}},\underbrace{0,\ldots,0}_{2^{s-1}-1},1) \in R.$$

As above, it follows that there is an s-ary Boolean function $g \in \mathrm{Pol}(\Gamma)$ such that

$$g(\mathrm{COLS}_s(1,-),\ldots,\mathrm{COLS}_s(s,-)) = u.$$

But g is not monotone because $g(0,\ldots,0) = u[1] = 1$ and $g(1,\ldots,1,0) = u[2^s - 1] = 0$. This contradicts that every function of $\mathrm{Pol}(\Gamma)$ is monotone, therefore $(1,1,0) \notin S$. Hence T-Imp $= S$ and it follows T-Imp $\in \langle R\rangle_{\nexists} \subseteq \langle\Gamma\rangle_{\nexists}$ as claimed.

□

The following theorem uses the same basic idea for the proof, with a slightly different relation used for implementation.

THEOREM 4.8. *Let* Γ *be a constraint language such that* $\mathrm{Inv}(\mathrm{M}_2) \subseteq \langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{V}_2)$. *Then* BAL-CSP($\Gamma$) *is* NP-*hard and* #BAL-CSP(Γ) *is* #P-*hard.*

Proof. We define the relation TF-Imp $= \mathrm{C}_1 \times \mathrm{C}_0 \times \mathrm{Imp}$. Obviously, transforming a formula $= \mathrm{Imp}(x_1,y_1) \wedge \cdots \wedge \mathrm{Imp}(x_n,y_n)$, into $\text{TF-Imp}(t,f,x_1,y_1) \wedge \cdots \wedge \text{TF-Imp}(t,f,x_n,y_n)$ (for new variables f and t) is a parsimonious reduction from #BAL-CSP(Imp) to #BAL-CSP(TF-Imp), thus Lemma 4.3 implies hardness for both decision and counting for TF-Imp.

We now prove the result for Γ. Let $s \geq 2$ be a core-size of $\langle\Gamma\rangle$, and let $R = \mathrm{Pol}(\Gamma)\,(\mathrm{COLS}_s)$. Due to Corollary 3.8 and the above result, it suffices to show that TF-Imp $\in \langle R\rangle_{\nexists}$ (TF-Imp is obviously irredundant). Since $\mathrm{Inv}(\mathrm{M}_2) \subseteq \langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{V}_2)$, it follows that $\mathrm{V}_2 \subseteq \mathrm{Pol}(\Gamma) \subseteq \mathrm{M}_2$. Hence the Boolean OR-function (which generates the clone V_2) is a polymorphism of Γ, and every polymorphism of Γ is monotone, 0-, and 1-reproducing. Let $n = 2^s$, be the arity of R, and let S be the relation defined by

$$S(t,f,x,y) = R(f,\underbrace{x,\ldots,x}_{\frac{n}{4}-1},\underbrace{y,\ldots,y}_{\frac{n}{4}},\underbrace{t,\ldots,t}_{\frac{n}{2}}).$$

It suffices to show $S =$ TF-Imp. Since $\mathrm{COLS}_s(1,-) = (0^{\frac{n}{2}},1^{\frac{n}{2}}) \in R$, it holds that $(1,0,0,0) \in S$ and since $\mathrm{COLS}_s(1,-) \vee \mathrm{COLS}_s(2,-) \in R$ it holds that $(1,0,0,1) \in S$. Because $(0,1,\ldots,1) = \mathrm{COLS}_s(1,-) \vee \cdots \vee \mathrm{COLS}_s(s,-) \in R$, we have that $(1,0,1,1) \in S$. Thus TF-Imp $\subseteq S$. Since the left-most column of COLS_s contains only 0s, and the rightmost one only 1s, and R is obtained from

COLS_s by applying functions from $\mathrm{Pol}(\Gamma)$ which are 0- and 1-reproducing, the left- and rightmost columns of R only contain 0s (1s, respectively) as well. It follows that $S \subseteq \{1\} \times \{0\} \times \{0,1\}^2$. Therefore it remains to prove $(1,0,1,0) \notin S$. Assume this is not the case. Then

$$u = (0, \underbrace{1, \ldots, 1}_{\frac{n}{4}-1}, \underbrace{0, \ldots, 0}_{\frac{n}{4}}, \underbrace{1, \ldots, 1}_{\frac{n}{2}}) \in R.$$

Due to the discussion after Theorem 3.7, there is an s-ary Boolean function $g \in \mathrm{Pol}(\Gamma)$ with $g(\mathrm{COLS}_s(1,-), \ldots, \mathrm{COLS}_s(s,-)) = u$. Then $g(0,\ldots,0,1) = u[2] = 1$ and $g(0,1,\ldots,1) = u[\frac{n}{2}] = 0$, i.e., g is not monotone, which is a contradiction to $g \in \mathrm{Pol}(\Gamma) \subseteq \mathrm{M}_2$. Hence TF-Imp $= S$ as required. □

The following theorem covers the cases $\mathrm{Inv}(\mathrm{S}_0^m)$ and $\mathrm{Inv}(\mathrm{S}_{02}^m)$ for all $m \geq 2$. The proof follows the same lines as the proofs for Theorems 4.7 and 4.8.

THEOREM 4.9. *Let Γ be a constraint language such that $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{S}_{02}^m)$ or $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{S}_0^m)$ for some natural number $m \geq 2$. Then* BAL-CSP(Γ) *is* NP*-hard and* #BAL-CSP(Γ) *is* #P*-hard.*

Proof. Let T-Or $= \mathrm{C}_1 \times \mathrm{Or}^2$ and TF-Or $= \mathrm{C}_1 \times \mathrm{C}_0 \times \mathrm{Or}^2$. Recall that both decision and counting are hard for Or^2 due to Lemma 4.4. Hardness for TF-Or follows trivially, since transforming $\wedge_{i=1}^n \mathrm{Or}^2(x_i, y_i)$ to $\wedge_{i=1}^n \text{TF-Or}(t, f, x_i, y_i)$ (for new variables f and t) is a parsimonious reduction. We now consider hardness of the decision problem BAL-CSP(T-Or). For that let $\varphi = \bigwedge_{i=1}^n \mathrm{Or}^2(x_i, y_i)$ be an Or^2-formula. We construct a T-Or-formula $\varphi' = \text{T-Or}(t, x_1, y_1) \wedge \cdots \wedge \text{T-Or}(t, x_n, y_n) \wedge \text{T-Or}(t, t, f)$ for new variables f and t. Obviously every balanced solution $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ for φ can be extended to a balanced solution for φ' by setting $I(t) = 1$ and $I(f) = 0$.

Now let $I\colon \mathrm{Var}(\varphi') \to \{0,1\}$ be a balanced solution of φ'. It is obvious that $I(t) = 1$. If $I(f) = 0$, the restriction of I to $\mathrm{Var}(\varphi)$ is a balanced solution for φ. If on the other hand $I(f) = 1$, then I assigns 0 to two variables more of $\mathrm{Var}(\varphi)$ than 1. Let $z \in \mathrm{Var}(\varphi)$ be a variable mapped to 0 by I and let $J\colon \mathrm{Var}(\varphi) \to \{0,1\}$ defined by $J(z) = 1$ and for all $x \neq z$: $J(x) = I(x)$. Since every Or^2-clause of φ is satisfied by $I|_{\mathrm{Var}(\varphi)}$ and because of the monotonicity of Or^2, the clauses are satisfied by J as well, so J is a balanced solution for φ. Therefore we showed BAL-CSP$(\mathrm{Or}^2) \leq_m^{\log}$ BAL-CSP(T-Or) which gives us NP-hardness of BAL-CSP(T-Or) due to Lemma 4.4.

The above reduction is not parsimonious. For #P-hardness of #BAL-CSP(T-Or) we show that #CSP$(\mathrm{Or}^2) \leq_!^{\log}$ #BAL-CSP(T-Or) (note that since CSP(Or^2) can be solved in polynomial time, this does not establish hardness of the decision problem, hence the seperate proof above). Let $\varphi = \mathrm{Or}^2(x_1, y_1) \wedge \cdots \wedge \mathrm{Or}^2(x_n, y_n)$ be an Or^2-formula. Let v' for every $v \in \mathrm{Var}(\varphi)$, and f and t be new and distinct variables. We define

$$\varphi' = \text{T-Or}(t, x_1, y_1) \wedge \cdots \wedge \text{T-Or}(t, x_n, y_n) \wedge \bigwedge_{v \in \mathrm{Var}(\varphi)} \text{T-Or}(t, v, v') \wedge \text{T-Or}(t, t, f).$$

The correctness of the reduction can be proven analogously to Lemma 4.4. It follows that #CSP$(\mathrm{Or}^2) \leq_!^{\log}$ #BAL-CSP(T-Or). With [Creignou and Hermann 1996], it follows that #BAL-CSP(T-Or) is hard for #P.

We now prove hardness for the constraint language Γ. For this, let $s \geq 2$ be a core-size of $\langle\Gamma\rangle$ and $R = \mathrm{Pol}(\Gamma)\,(\mathrm{COLS}_s)$. Due to Corollary 3.8, it suffices to show that T-Or $\in \langle R\rangle_{\nexists}$ or TF-Or $\in \langle R\rangle_{\nexists}$.

Case 1. $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{S}_0^m)$ for some $m \geq 2$. We show T-Or $\in \langle R\rangle_{\nexists}$. Let S be the Boolean relation defined by

$$S(t,x,y) \equiv R(\underbrace{x,\ldots,x}_{2^{s-1}},\underbrace{y,\ldots,y}_{2^{s-2}},\underbrace{t,\ldots,t}_{2^{s-2}}).$$

We show $S =$ T-Or. Since $\langle\Gamma\rangle \subseteq \mathrm{Inv}(\mathrm{I}_1)$, it holds that the constant 1-functions is a polymorphism of Γ, and therefore $(1,\ldots,1) \in R$ and $(1,1,1) \in S$. Note that $\mathrm{COLS}_s(1,-) = (0^{2^{s-1}}, 1^{2^{s-1}})$, and $\mathrm{COLS}_s(2,-) = (0^{2^{s-2}}, 1^{2^{s-2}}, 0^{2^{s-2}}, 1^{2^{s-2}})$. Because Boolean implication is an element of $\mathrm{S}_0^m \subseteq \mathrm{Pol}(\Gamma)$, it follows that

$$(\underbrace{1,\ldots,1}_{2^{s-1}},\underbrace{0,\ldots,0}_{2^{s-2}},\underbrace{1,\ldots,1}_{2^{s-2}}) = \mathrm{COLS}_s(1,-) \rightarrow \mathrm{COLS}_s(2,-) \in R,$$

and therefore $(1,1,0) \in S$. Since $\mathrm{COLS}_s(1,-) \in R$, it holds that $(1,0,1) \in S$, hence T-Or $\subseteq S$. Since $\mathrm{S}_0^m \subseteq \mathrm{R}_1$, every polymorphism of Γ is 1-reproducing. Since the right-most column of COLS_s contains only 1s, and every element in R is obtained by the application of 1-reproducing functions to COLS_s, the rightmost column of R also contains only 1s. Therefore, for all $a, b \in \{0,1\}$ it follows that $(0,a,b) \notin S$. It remains to show that $(1,0,0) \notin S$: Assume $(1,0,0) \in S$. As before, it follows that there is some s-ary function $g \in \mathrm{S}_0^m$ such that

$$(\underbrace{0,\ldots,0}_{2^{s-1}},\underbrace{0,\ldots,0}_{2^{s-2}},\underbrace{1,\ldots,1}_{2^{s-2}}) = g(\mathrm{COLS}_s(1,-),\ldots,\mathrm{COLS}_s(s,-)).$$

That means $g(\mathrm{COLS}_s(-,i)) = 1$ if and only if $2^{s-1}+2^{s-2} < i \leq 2^s$, and therefore we have $g(a_1,\ldots,a_s) = a_1 \wedge a_2$. Therefore, g is essentially Boolean conjunction, which generates the clone E_2. Since $g \in \mathrm{Pol}(\Gamma)$, it therefore follows that $\mathrm{E}_2 \subseteq \mathrm{Pol}(\Gamma) = \mathrm{S}_0^m$, which is not true (see Figure 1). Hence, $S =$ T-Or and therefore T-Or $\in \langle R\rangle_{\nexists}$ as required.

Case 2. $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{S}_{02}^m)$ for some $m \geq 2$. We show TF-Or $\in \langle R\rangle_{\nexists}$. Let S be the Boolean relation defined by

$$S(t,f,x,y) \equiv R(\underbrace{f,\ldots,f}_{2^{s-2}},\underbrace{x,\ldots,x}_{2^{s-2}},\underbrace{y,\ldots,y}_{2^{s-2}},\underbrace{t,\ldots,t}_{2^{s-2}}).$$

We show $S =$ TF-Or. Since $\mathrm{COLS}_s(1,-) \in R$ and $\mathrm{COLS}_s(2,-) \in R$, it holds that $(1,0,0,1) \in S$ and $(1,0,1,0) \in R$. Note that $\vee \in \mathrm{V}_2 \subseteq \mathrm{Pol}(\Gamma)$ (see Figure 1). That means $\mathrm{COLS}_s(1,-) \vee \mathrm{COLS}_s(2,-) \in R$ and therefore $(1,0,1,1) \in S$. Hence, TF-Or $\subseteq S$.

Since $\mathrm{S}_{02}^m \subseteq \mathrm{R}_0$ and $\mathrm{S}_{02}^m \subseteq \mathrm{R}_1$ every polymorphism of Γ is 0-reproducing and 1-reproducing. Therefore the first column of R equals $(0,\ldots,0)$ and the last column of R equals $(1,\ldots,1)$, that means all tuples from S have the form $(1,0,a,b)$ for

Clone	Base
D_2	$(x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$
L	$\{x \oplus y, 1\}$
L_1	$\{x \oplus y \oplus 1\}$
L_2	$\{x \oplus y \oplus z\}$
L_3	$\{x \oplus y \oplus z \oplus 1\}$
I_0	$\{id, 0\}$

Fig. 2. Bases for selected clones.

some $a, b \in \{0, 1\}$. Assume $(1, 0, 0, 0) \in S$. Then according to Proposition 3.5 there is some s-ary function $g \in S_{02}^m$ such that

$$(\underbrace{0, \ldots, 0}_{2^{s-2}}, \underbrace{0, \ldots, 0}_{2^{s-2}}, \underbrace{0, \ldots, 0}_{2^{s-2}}, \underbrace{1, \ldots, 1}_{2^{s-2}}) = g(\mathrm{COLS}_s(1, -), \ldots, \mathrm{COLS}_s(s, -)).$$

As in Case 1, it follows that $E_2 \subseteq \mathrm{Pol}(\Gamma)$, which again is contradiction. Thus, $S = \text{TF-Or}$ and therefore $\text{TF-Or} \in \langle R \rangle_{\nexists}$, which concludes the proof.

□

4.4 Hardness Results with Non-Unified Proofs

For the convenience of the reader we have recalled in Fig. 2 a list of bases for selected clones, which will be of use in the proofs of the next theorems.

In this section we work with concrete irredundant weak bases in all proofs. Formally, to apply Corollary 3.8, it is necessary to know a core-size for the co-clone in question beforehand. However, knowing a core-size for the co-clone under consideration can sometimes be avoided: Suppose that s is a number such that $\mathcal{C}\,(\mathrm{COLS}_s)$ is a base for the co-clone $\mathrm{Inv}(\mathcal{C})$ in the usual sense, i.e., $\langle \mathcal{C}\,(\mathrm{COLS}_s) \rangle = \mathrm{Inv}(\mathcal{C})$ (this is equivalent to saying that $\mathrm{Pol}(\mathcal{C}\,(\mathrm{COLS}_s)) = \mathcal{C}$, i.e., that $\mathcal{C}\,(\mathrm{COLS}_s)$ is not closed under any function *not* in $\mathcal{C}$). Since COLS_s is an $\mathcal{C}$-core of $\mathcal{C}\,(\mathrm{COLS}_s)$, it then immediately follows that s is a core-size for $\mathrm{Inv}(\mathcal{C})$, and hence $\mathcal{C}\,(\mathrm{COLS}_s)$ is a weak base of $\mathrm{Inv}(\mathcal{C})$. In the proof of Theorem 4.12 and Theorem 4.18, we proceed in this way. The core-sizes we need in the proofs of the other theorems are listed in the following lemma.

LEMMA 4.10. 3 *is a core-size for the co-clones* $\mathrm{Inv}(D_2)$, $\mathrm{Inv}(L_3)$, $\mathrm{Inv}(L_2)$, $\mathrm{Inv}(I_2)$, *and* $\mathrm{Inv}(N_2)$, 2 *is a core-size for the co-clones* $\mathrm{Inv}(L_1)$ *and* N.

Proof. The proofs here are very simple and all follow the same pattern: To prove that some number s is a core-size for a co-clone $\mathrm{Inv}(\mathcal{C})$, it suffices to exhibit a relation R that has s elements, and for which $\langle \mathcal{C}(R) \rangle = \mathrm{Inv}(\mathcal{C})$. Note that determining the minimal core-size for a co-clone with a finite base can be done automatically [Schnoor and Schnoor 2008]. As an example let us prove that 3 is a core-size of the co-clone $\mathrm{Inv}(L_3)$.

By definition of core-size, we need to show that there is a relation R with $|R| = 3$ such that $L_3(R)$, which is obtained from R by applying functions from L_3 and adding the result until the relation does not grow anymore, has polymorphism set L_3. By construction, it is clear that *every* relation of the form $L_3(R)$ is closed under every function in L_3, the important point is to prove that there are no additional polymorphisms.

In [Böhler et al. 2005], it was shown that $\mathrm{Pol}(\mathrm{Odd}^4) = \mathrm{L}_3$. Now consider the relation

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

and denote the rows of R with v_1, v_2, v_3. The clone L_3 is generated by the function $f(x,y,z) = x \oplus y \oplus z \oplus 1$. If we apply this function coordinate-wise to the tuples v_1, v_2, and v_3, then the result is the tuple $v_4 = (1,0,0,0)$, and hence this is an element of $\mathrm{L}_3(R)$. Since $f(x,x,x) = \overline{x}$, $\mathrm{L}_3(R)$ also contains the negations of the tuples v_1, v_2, v_3, and v_4. These 8 tuples together form the relation Odd^4, and since Odd^4 is closed under L_3 due to the above result, it follows that no further tuples are added. Hence we have proven that $\mathrm{L}_3(R) = \mathrm{Odd}^4$, and therefore 3 is a core-size of $\mathrm{Inv}(\mathrm{L}_3(R))$, as claimed. □

THEOREM 4.11. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{D}_2)$. *Then* BAL-CSP($\Gamma$) *is* NP-*hard and* #BAL-CSP(Γ) *is* #P-*hard.*

Proof. From Fig. 2 it follows that D_2 is generated by the the ternary majority function, maj. Due to Lemma 4.10, 3 is a core-size for $\mathrm{Inv}(\mathrm{D}_2)$. Let $R = \mathrm{maj}(\mathrm{COLS}_3)$. It can be verified that

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Note that the second row is generated by the coordinate-wise application of maj to the other three rows, which build COLS_3. Define the Boolean relations $S(t,f,x,y) = R(f,f,x,x,y,y,t,t)$ and $T(t,f,v,w,x,y) = R(f,f,v,w,y,x,t,t)$, then by definition $\{S,T\} \subseteq \langle R\rangle_{\nexists}$. Since both relations are irredundant, Corollary 3.8 states that it suffices to show #BAL-CSP(Imp) $\leq_!^{\log}$ #BAL-CSP($\{S,T\}$) (hardness for both counting and decision then follows from Lemma 4.3). The following equivalences can be verified:

$$S(t,f,x,y) \equiv \mathrm{C}_1(t) \wedge \mathrm{C}_0(f) \wedge \mathrm{Odd}^2(x,y)$$
$$T(t,f,v,w,x,y) \equiv \mathrm{C}_1(t) \wedge \mathrm{C}_0(f) \wedge \mathrm{Imp}(v,w) \wedge \mathrm{Odd}^2(v,x) \wedge \mathrm{Odd}^2(w,y)$$

Let $\varphi = \bigwedge_{i=1}^n \mathrm{Imp}(x_i,y_i)$ be an Imp-formula. We construct an $\{S,T\}$-formula as follows: Let f, t, and z' and z'' for every $z \in \mathrm{Var}(\varphi)$ be new and distinct variables. We define $\varphi' = \bigwedge_{i=1}^n T(t,f,x_i,y_i,x_i',y_i') \wedge S(t,f,x_i',x_i'') \wedge S(t,f,y_i',y_i'')$, then

$$\varphi' \equiv \varphi \wedge \bigwedge_{i=1}^{n} \mathrm{Imp}(x_i,y_i) \wedge \bigwedge_{z\in\varphi} \mathrm{Odd}^2(z,z') \wedge \mathrm{Odd}^2(z',z'') \wedge \mathrm{C}_1(t) \wedge \mathrm{C}_0(f)$$

Every balanced solution $I\colon \mathrm{Var}(\varphi') \to \{0,1\}$ of φ' is already balanced on $\mathrm{Var}(\varphi')\setminus \mathrm{Var}(\varphi)$, because $I(t) \neq I(f)$ and because the clauses of the form $\mathrm{Odd}^2(z', z'')$ provide that for every $z \in \varphi$ it holds $I(z') \neq I(z'')$. It follows that I is balanced on $\mathrm{Var}(\varphi)$ as well and therefore $I|_{\mathrm{Var}(\varphi)}$ is a balanced solution for φ.

Conversely, every balanced solution $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ for φ can be extended to a balanced solution of φ by setting $I(t) = 1$, $I(f) = 0$, $I(z') = \neg I(z)$ and $I(z'') = I(z)$ for every $z \in \mathrm{Var}(\varphi)$. Since every other extension of I to $\mathrm{Var}(\varphi')$ does not satisfy φ', we have a one-to-one correspondence between balanced solutions of φ and the balanced solutions of φ'. Hence, $\#\textsc{Bal-Csp}(\mathrm{Imp}) \leq_{!}^{\log} \#\textsc{Bal-Csp}(\{S,T\})$ as required. $\square$

With the next four theorems we cover the cases $\mathrm{Inv}(\mathrm{L})$, $\mathrm{Inv}(\mathrm{L}_1)$, $\mathrm{Inv}(\mathrm{L}_2)$ and $\mathrm{Inv}(\mathrm{L}_3)$. Note that the hardness-results transfer to $\mathrm{Inv}(\mathrm{L}_0)$ due to Proposition 4.2, because $\mathrm{Inv}(\mathrm{L}_0) = \mathrm{dual}(\mathrm{Inv}(\mathrm{L}_1))$. The proofs of the following theorems are very similar, however they differ in so many details that there does not seem to be a way to combine them.

THEOREM 4.12. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{L})$. *Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP*-hard and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P*-hard.*

Proof. Let $R = \mathrm{L}(\mathrm{COLS}_2)$. It can be verified that

$$R = \begin{pmatrix} 0&0&0&0\\ 0&0&1&1\\ 0&1&0&1\\ 0&1&1&0\\ 1&0&0&1\\ 1&0&1&0\\ 1&1&0&0\\ 1&1&1&1 \end{pmatrix} = \mathrm{Even}^4.$$

(The second and the third row are from COLS_2, the other ones result from the application of functions from L to them.) Due to [Böhler et al. 2005], $\langle\mathrm{Even}^4\rangle = \mathrm{Inv}(\mathrm{L})$. It therefore follows that 2 is a core-size of $\mathrm{Inv}(\mathrm{L})$. Since Even^4 is obviously irredundant, Corollary 3.8 shows that it suffices to prove the hardness result for Even^4. We show $\#\textsc{Bal-Csp}(\mathrm{Odd}^3) \leq_c^{\log} \#\textsc{Bal-Csp}(\mathrm{Even}^4)$ with a reduction that also is a many-one reduction, the result then follows from Lemma 4.5. Let $\varphi = \bigwedge_{i=1}^n \mathrm{Odd}^3(x_i, y_i, z_i)$ be an Odd^3-formula. Let $k = |\mathrm{Var}(\varphi)|$ and let $t, t_1, \ldots, t_k, f, f_1, \ldots, f_k$ be new and distinct variables. We define:

$$\varphi' = \bigwedge_{i=1}^{n} \mathrm{Even}^4(t, x_i, y_i, z_i) \wedge \bigwedge_{i_1}^{k} \mathrm{Even}^4(t,t,t,t_i) \wedge \mathrm{Even}^4(f,f,f,f_i).$$

We prove that φ' has exactly twice as many balanced solutions as φ, which implies that the above is both a counting and a many-one reduction. First note that the clauses $\mathrm{Even}^4(t,t,t,t_i)$ and $\mathrm{Even}^4(f,f,f,f_i)$ for all $i \in \{1, \ldots, k\}$ imply that every solution of φ' maps $t_1, \ldots, t_k$ to the same value as t and $f_1, \ldots, f_k$ to the same value as f. Further, every balanced solution of φ' must map f and t to different values, because otherwise this value would be assigned to at least $2k + 2$ variables, which is more than half of the variables of φ. Now let $I\colon \mathrm{Var}(\varphi) \to \{0, 1\}$ be a balanced solution of φ. It can be verified that we get a balanced solution for φ' by setting $I(f) = I(f_1) = \ldots = I(f_k) = 0$ and $I(t) = I(t_1) = \ldots = I(t_k) = 1$. This is the only extension of I to $\mathrm{Var}(\varphi')$ that is balanced and satisfies φ', because if we map t to 0, then $I(t) + I(x_i) + I(y_i) + I(z_i)$ would be odd for every $i \in \{1, \ldots, n\}$.

Finally let $I\colon \mathrm{Var}(\varphi') \to \{0, 1\}$ be a balanced solution for φ'. Since I is already balanced on the newly introduced variables, it holds that I is also balanced on $\mathrm{Var}(\varphi)$. For every $1 \le i \le n$ we know that $I(t) + I(x_i) + I(y_i) + I(z_i)$ is even, therefore $I(x_i) + I(y_i) + I(z_i)$ is odd if and only if $I(t) = 0$. That means $I|_{\mathrm{Var}(\varphi)}$ is a balanced solution for φ if and only if $I(t) = 1$.

Since $\neg \in \mathrm{N}_2 \subseteq \mathrm{L}$, it holds that I', defined by $I'(x) = \neg I(x)$, is a balanced solution of φ' as well. That means exactly half of that balanced solutions of φ' can be restricted to a balanced solution of φ. Thus, φ' has exactly twice as many balanced solutions as φ as claimed. □

Theorem 4.13. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{L}_3)$*. Then* Bal-Csp($\Gamma$) *is* NP*-hard and* #Bal-Csp(Γ) *is* #P*-hard.*

Proof. Due to Lemma 4.10, 3 is a core-size for $\mathrm{Inv}(\mathrm{L}_3)$. From Corollary 3.8, it follows that it suffices to prove the hardness results for $R = \mathrm{L}_3(\mathrm{COLS}_3)$ (since R is irredundant). It can be verified that

$$R = \begin{pmatrix} 0&0&0&0&1&1&1&1\\ 0&0&1&1&0&0&1&1\\ 0&1&0&1&0&1&0&1\\ 0&1&1&0&1&0&0&1\\ 1&0&0&1&0&1&1&0\\ 1&0&1&0&1&0&1&0\\ 1&1&0&0&1&1&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}.$$

Note that the first three tuples of R form COLS_3. It holds that

$$R(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \equiv$$

$$\mathrm{Even}^4(x_1, x_2, x_3, x_4) \wedge \mathrm{Odd}^2(x_1, x_8) \wedge \mathrm{Odd}^2(x_2, x_7) \wedge \mathrm{Odd}^2(x_3, x_6) \wedge \mathrm{Odd}^2(x_4, x_5).$$

In the proof of Theorem 4.12 we showed that Bal-Csp(Even^4) is NP-hard and #Bal-Csp(Even^4) is #P-hard. Therefore it suffices to show #Bal-Csp(Even^4) $\le_!^{\log}$ #Bal-Csp(R) to prove this theorem. Let $\varphi = \bigwedge_{i=1}^{n} \mathrm{Even}^4(w_i, x_i, y_i, z_i)$ be an Even^4-formula. Without loss of generality assume that $k = |\mathrm{Var}(\varphi)|$ is at least 2. For every $v \in \mathrm{Var}(\varphi)$ let $v^1, \ldots, v^k$ be new and distinct variables. We set:

$$\varphi' = \bigwedge_{i=1}^{n} R(w_i, x_i, y_i, z_i, z_i^1, y_i^1, x_i^1, w_i^1) \wedge \cdots \wedge R(w_i, x_i, y_i, z_i, z_i^k, y_i^k, x_i^k, w_i^k).$$

Then the following equivalence holds:

$$\varphi' \equiv \varphi \wedge \bigwedge_{v \in \mathrm{Var}(\varphi)} \mathrm{Odd}^2(v, v^1) \wedge \cdots \wedge \mathrm{Odd}^2(v, v^k).$$

Let $I\colon \mathrm{Var}(\varphi') \to \{0,1\}$ be a balanced solution of φ'. Clearly $I|_{\mathrm{Var}(\varphi)}$ is a solution of φ. We show that $I|_{\mathrm{Var}(\varphi)}$ is balanced. Let l_0 be the number of variables mapped to 0 and l_1 be the number of variables mapped to 1 by $I|_{\mathrm{Var}(\varphi)}$. It holds for every $v \in \mathrm{Var}(\varphi)$ and for every $1 \leq i \leq k$ that $I(v) \neq I(v^i)$ in order to satisfy $\mathrm{Odd}^2(v, v^i)$. Therefore I maps $l_0 + kl_1$ variables to 0 and $l_1 + kl_0$ variables to 1. Since I is balanced, it follows $l_0 + kl_1 = l_1 + kl_0$, which implies $l_0 = l_1$ because $k \geq 2$. Hence, $I|_{\mathrm{Var}(\varphi)}$ is a balanced solution for φ. Obviously I is uniquely determined by $I|_{\mathrm{Var}(\varphi)}$.

Now let $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$ be a balanced solution for φ. For every $i \in \{1, \ldots, k\}$ and every $v \in \mathrm{Var}(\varphi)$ we set $I(v^i) \neq_{def} I(v)$. Clearly this extension satisfies φ' and with the above equations it can be seen that it is balanced on $\mathrm{Var}(\varphi')$.

Thus we have a one-to-one correspondence between balanced solutions from φ and balanced solutions from φ'. It follows that $\#\textsc{Bal-Csp}(\mathrm{Even}^4) \leq_!^{\log} \#\textsc{Bal-Csp}(R)$, which completes the proof. □

Theorem 4.14. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{L}_1)$. *Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP*-hard and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P*-hard.*

Proof. Due to Lemma 4.10, 2 is a core-size of $\mathrm{Inv}(\mathrm{L}_1)$. Let $R = \mathrm{L}_1(\mathrm{COLS}_2)$, it can be verified that

$$R = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Note that the first two rows of R are the ones from COLS_2. It is easy to see that $R = \mathrm{Odd}^3 \times \mathrm{C}_1$ and that R is irredundant. From Lemma 4.5, we know that our problems are hard for Odd^3, hence Corollary 3.8 shows that it suffices to prove $\#\textsc{Bal-Csp}(\mathrm{Odd}^3) \leq_!^{\log} \#\textsc{Bal-Csp}(R)$. For this, let $\bigwedge_{i=1}^{n} \mathrm{Odd}^3(x_i, y_i, z_i)$ be an Odd^3-formula, let $k = \mathrm{Var}(\varphi)$, and let $t, t_1, \ldots, t_k, f, f_1, \ldots, f_k$ be new and distinct variables. We define:

$$\varphi' = \bigwedge_{i=1}^{n} R(x_i, y_i, z_i, t) \wedge \bigwedge_{i=1}^{k} R(f, f_i, t_i, t_i)$$

The correctness of the reduction can be shown with similar arguments as in the proof of Theorem 4.12. Note that here we get a parsimonious reduction (which then also is many-one) because the variable t must be mapped to 1 by every solution. □

Theorem 4.15. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{L}_2)$. *Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP*-hard and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P*-hard.*

Proof. Due to Lemma 4.10, 3 is a core-size of $\mathrm{Inv}(\mathrm{L}_2)$. Let $R = \mathrm{L}_2(\mathrm{COLS}_3)$, i.e.,

$$R = \begin{pmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&1&1&0&1&0&0&1 \end{pmatrix}.$$

The first three tuples of R are the ones from COLS_3. It can be verified that the following equivalence is true:

$$R(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \equiv$$
$$\mathrm{Even}^3(x_2, x_3, x_4) \wedge \mathrm{Odd}^2(x_2, x_7) \wedge \mathrm{Odd}^2(x_3, x_6) \wedge \mathrm{Odd}^2(x_4, x_5) \wedge \mathrm{C}_0(x_1) \wedge \mathrm{C}_1(x_8)$$

From Lemma 4.5 and by duality, we know that $\textsc{Bal-Csp}(\mathrm{Even}^3)$ is NP-hard and $\#\textsc{Bal-Csp}(\mathrm{Even}^3)$ is #P-hard under parsimonious reductions. It remains to show $\#\textsc{Bal-Csp}(\mathrm{Even}^3) \leq_!^{\log} \#\textsc{Bal-Csp}(R)$: Since R is irredundant, the theorem then follows from Corollary 3.8. Let $\varphi = \bigwedge_{i=1}^{n} \mathrm{Even}^3(x_i, y_i, z_i)$ be an Even^3-formula. Without loss of generality assume that $k = |\mathrm{Var}(\varphi)|$ is at least 2. Let f, t and $v^1, \ldots, v^k$ for every $v \in \mathrm{Var}(\varphi)$ be new and distinct variables. We define:

$$\varphi' = \bigwedge_{i=1}^{n} R(t, x_i, y_i, z_i, z_i^1, y_i^1, x_i^1, f) \wedge \cdots \wedge R(t, x_i, y_i, z_i, z_i^k, y_i^k, x_i^k, f).$$

Then the following equivalence holds:

$$\varphi' \equiv \varphi \wedge \mathrm{C}_0(f) \wedge \mathrm{C}_1(t) \wedge \bigwedge_{v \in \mathrm{Var}(\varphi)} \mathrm{Odd}^2(v, v^1) \wedge \cdots \wedge \mathrm{Odd}^2(v, v^k)$$

The correctness of this reduction can be proven with using similar arguments as in the proof of Theorem 4.13. □

Finally we prove NP-hardness and #P-hardness for the non-Schaefer constraint languages (i.e., the ones that only have constants, negation and identity as polymorphisms). Note that the case $\mathrm{Inv}(\mathrm{I})$ is already covered in Theorem 4.6 and that $\mathrm{Inv}(\mathrm{I}_0)$ is dual to $\mathrm{Inv}(\mathrm{I}_1)$. For the remaining four co-clones the proofs again are very similar, but differ in so many details that it seems easier to present them separately.

Theorem 4.16. *Let Γ be a constraint language such that $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{I}_2)$. Then $\textsc{Bal-Csp}(\Gamma)$ is NP-complete and $\#\textsc{Bal-Csp}(\Gamma)$ is #P-complete.*

Proof. Due to Lemma 4.10, 3 is a core-size of $\mathrm{Inv}(\mathrm{I}_2)$. Let $R = \mathrm{I}_2(\mathrm{COLS}_3)$. Because I_2 is generated by the identity function, it follows that $R = \mathrm{COLS}_3$, i.e.,

$$R = \begin{pmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \end{pmatrix}.$$

It can be verified that the following equivalence is true:

$$R(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \equiv \text{1-in-3}(x_2, x_3, x_5) \wedge \mathrm{C}_0(x_1) \wedge \mathrm{C}_1(x_8)$$
$$\wedge\, \mathrm{Odd}^2(x_2, x_7) \wedge \mathrm{Odd}^2(x_3, x_6) \wedge \mathrm{Odd}^2(x_4, x_5).$$

Since R is irredundant, Corollary 3.8 states that it suffices to prove the hardness result for R. From [Schaefer 1978], we know that $\textsc{Csp}$(1-in-3) is NP-hard and from [Creignou and Hermann 1996], it follows that $\#\textsc{Csp}$(1-in-3) is #P-hard. Hence, showing $\#\textsc{Csp}(\text{1-in-3}) \leq_!^{\log} \#\textsc{Bal-Csp}(R)$ completes the proof. Let $\varphi = \bigwedge_{i=1}^n \text{1-in-3}(x_i, y_i, z_i)$ be a 1-in-3-formula. Let f, t and v' for every $v \in \text{Var}(\varphi)$ be new and distinct variables. We define an R-formula $\varphi' = \bigwedge_{i=1}^n R(f, x_i, y_i, z_i', z_i, y_i', x_i', t)$. According to the above it holds that

$$\varphi' \equiv \varphi \wedge \bigwedge_{v \in \text{Var}(\varphi)} \text{Odd}^2(v_i, v_i') \wedge \text{C}_0(f) \wedge \text{C}_1(t).$$

Obviously every balanced solution of φ' satisfies φ as well and every solution of φ can be extended uniquely to a balanced solution for φ'. It follows that $\#\textsc{Csp}(\text{1-in-3}) \leq_!^{\log} \#\textsc{Bal-Csp}(R)$. □

THEOREM 4.17. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \text{Inv}(\text{N}_2)$. *Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP-*complete and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P-*complete.*

Proof. Due to Lemma 4.10, 3 is a core-size of $\text{Inv}(\text{N}_2)$. Let $R = \text{N}_2(\text{COLS}_3)$, then

$$R = \begin{pmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 1&0&1&0&1&0&1&0 \\ 1&1&0&0&1&1&0&0 \\ 1&1&1&1&0&0&0&0 \end{pmatrix}.$$

Since R is irredundant, Corollary 3.8 shows that it suffices to prove the hardness result for R. As mentioned in the previous proof, $\textsc{Csp}$(1-in-3) is NP-hard and $\#\textsc{Csp}$(1-in-3) is #P-hard. Hence showing $\#\textsc{Csp}(\text{1-in-3}) \leq_c^{\log} \#\textsc{Bal-Csp}(R)$ with a reduction from which we can derive a many-one reduction between the decision problems completes the proof. Let $\varphi = \bigwedge_{i=1}^n \text{1-in-3}(x_i, y_i, z_i)$ be a 1-in-3-formula. Let f, t and v' for every $v \in \text{Var}(\varphi)$ be new and distinct variables. We define an R-formula ψ as follows:

$$\psi = \bigwedge_{i=1}^n R(f, x_i, y_i, z_i', z_i, y_i', x_i', t)$$

We show that the number of balanced solutions for ψ is exactly twice the number of solutions for φ. Let φ' be the formula obtained from φ as in the proof of Theorem 4.16:

$$\varphi' \equiv \varphi \wedge \bigwedge_{v \in \text{Var}(\varphi)} \text{Odd}^2(v_i, v_i') \wedge \text{C}_0(f) \wedge \text{C}_1(t).$$

Observe that the relation R in the current proof is obtained from the corresponding relation in the previous one by adding all negations. Thus, it is easy to see that a solution for ψ maps f to 0 if and only if it is a solution for φ'. Since the number of solutions for φ and balanced solutions for φ' is exactly the same (see proof of

Theorem 4.16), it holds that the number of balanced solutions for ψ is exactly twice the number of solutions for φ. In particular, the process provides also a many-one reduction between the decision problems.

Since the formula ψ can be constructed in logarithmic space it follows that $\#\text{Csp}(\text{1-in-3}) \leq_c^{\log} \#\text{Bal-Csp}(R)$ and $\text{Csp}(\text{1-in-3}) \leq_m^{\log} \text{Bal-Csp}(R)$. □

THEOREM 4.18. *Let* Γ *be a finite constraint language over* $\{0,1\}$ *such that* $\langle\Gamma\rangle = \text{Inv}(\text{I}_0)$. *Then* $\text{Bal-Csp}(\Gamma)$ *is* NP*-complete and* $\#\text{Bal-Csp}(\Gamma)$ *is* #P*-complete.*

Proof. Let

$$R = \text{I}_0(\text{COLS}_2) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

By construction, R is closed under I_0. R is not closed under the constant 1-function, disjunction, exclusive-or, or conjunction (consider applying these functions to the second and third tuples), hence $\text{Pol}(R)$ is not a superset of I, V_0, L_0, or E_0. Since these three clones are the direct upper neighbours of I_0 in Post's lattice, it follows that $\text{Pol}(R) = \text{I}_0$, and hence $\langle R\rangle = \text{Inv}(\text{I}_0)$. Therefore, following the discussion after Theorem 3.7, it follows that 2 is a coresize of $\text{Inv}(\text{I}_0)$. Since R is irredundant, Corollary 3.8 implies that it suffices to prove the hardness result for R. Since $\langle R\rangle = \text{Inv}(\text{I}_0)$ and since R_1 is the Boolean clone such that $\langle \text{C}_1\rangle = \text{Inv}(\text{R}_1)$, it holds that

$$\langle\{R, \text{C}_1\}\rangle = \langle \text{Inv}(\text{I}_0) \cup \text{Inv}(\text{R}_1)\rangle = \text{Inv}(\text{I}_2).$$

Therefore it follows from [Schaefer 1978] that $\text{Csp}(\{R, \text{C}_1\})$ is NP-hard and from [Creignou and Hermann 1996] that $\#\text{Csp}(\{R, \text{C}_1\})$ is hard for #P under parsimonious reductions. Hence, showing $\#\text{Csp}(\{R, \text{C}_1\}) \leq_!^{\log} \#\text{Bal-Csp}(R)$ completes the proof. Let $\varphi = \bigwedge_{i=1}^n R(w_i, x_i, y_i, z_i) \wedge \bigwedge_{i=1}^m \text{C}_1(v_i)$ be a $\{R, \text{C}_1\}$-formula. Let $k = |\text{Var}(\varphi)|$ be the number of variables appearing in φ and let $f, f_1, \ldots, f_k, t, t_1, \ldots, t_k$ and x' for every $x \in \text{Var}(\varphi)$ be new and distinct variables. We define an R-formula φ' as follows:

$$\varphi' = \bigwedge_{i=1}^{n} R(w_i, x_i, y_i, z_i) \wedge \bigwedge_{x\in\text{Var}(\varphi)} R(f, x, x', t)$$
$$\wedge \bigwedge_{i=1}^{k} R(f_i, f_i, t, t_i) \wedge \bigwedge_{i=1}^{m} R(f, f, t, v_i)$$

Note that $|\text{Var}(\varphi')| = 4k + 2$. Let $I\colon \text{Var}(\varphi') \to \{0,1\}$ be a balanced solution for φ'. The clauses of the form $R(f_i, f_i, t, t_i)$ give us that $I(f_1) = \cdots = I(f_k) = 0$ and $I(t) = I(t_1) = \cdots = I(t_k)$. Since f appears in some of the clauses in the first position, we have $I(f) = 0$. It follows $I(t) = I(t_1) = \cdots = I(t_k) = 1$, otherwise I would map at least $2k + 2$ variables to 0 which means that I would not be balanced. Because of the clauses of the type $R(f, f, t, v_i)$, it holds that $I(v_1) = \cdots = I(v_m) = I(t) = 1$, therefore I satisfies φ.

From the clauses of the type $R(f,x,x',t)$ it follows that for all $x \in \mathrm{Var}(\varphi)$ it holds $I(x) \neq I(x')$. That means I is uniquely determined by $I|_{\mathrm{Var}(\varphi)}$. So for every solution of φ we can define at most one extension that is a balanced solution for φ. Now let $I\colon \mathrm{Var}(\varphi) \to \{0,1\}$. We extend I by defining $I(f) = I(f_1) = \ldots = I(f_k) = 0$, $I(t) = I(t_1) = \ldots = I(t_k) = 1$, and $I(x') = \neg I(x)$ for every $x \in \mathrm{Var}(\varphi)$. Obviously this extension is balanced and, since for all $i \in 1,\ldots,m$ we have $I(v_i) = 1$, it satisfies φ'. Hence, the balanced solutions for φ' are unique extensions of solutions for φ. It follows that $\#\textsc{Csp}(\{R, \mathrm{C}_1\}) \leq^{\log}_{!} \#\textsc{Bal-Csp}(R)$. □

Theorem 4.19. *Let* Γ *be a constraint language such that* $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{N})$. *Then* $\textsc{Bal-Csp}(\Gamma)$ *is* NP-*complete and* $\#\textsc{Bal-Csp}(\Gamma)$ *is* #P-*complete.*

Proof. Due to Example 3.9, 2 is a core-size of $\mathrm{Inv}(\mathrm{N})$. Let $R = \mathrm{N}(\mathrm{COLS}_2)$ (also see Example 3.9). Since $\langle R\rangle = \mathrm{Inv}(\mathrm{N})$ and $\langle \mathrm{C}_0, \mathrm{C}_1\rangle = \mathrm{Inv}(\mathrm{R}_2)$, it holds that $\langle\{R, \mathrm{C}_0, \mathrm{C}_1\}\rangle = \langle \mathrm{Inv}(\mathrm{N}) \cup \mathrm{Inv}(\mathrm{R}_2)\rangle = \mathrm{Inv}(\mathrm{I}_2)$. Therefore it follows from [Schaefer 1978] and [Creignou and Hermann 1996] that $\textsc{Csp}(\{R, \mathrm{C}_0, \mathrm{C}_1\})$ is NP-hard and $\#\textsc{Csp}(\{R, \mathrm{C}_0, \mathrm{C}_1\})$ is #P-hard. Hence, showing that $\#\textsc{Csp}(\{R, \mathrm{C}_0, \mathrm{C}_1\}) \leq^{\log}_{c}$ $\#\textsc{Bal-Csp}(R)$ (with a counting reduction that also provides a many-one reduction between the decision problems) completes the proof with an application of Corollary 3.8, since R is irredundant. Let $\varphi = \bigwedge_{i=1}^{n} R(w_i, x_i, y_i, z_i) \wedge \bigwedge_{i=1}^{m_0} \mathrm{C}_0(v_i) \wedge \bigwedge_{i=1}^{m_1} \mathrm{C}_1(u_i)$ be a $\{R, \mathrm{C}_0, \mathrm{C}_1\}$-formula, let $k = |\mathrm{Var}(\varphi)|$ and let $f, f_1, \ldots, f_k, t, t_1, \ldots, t_k$ and x' for every $x \in \mathrm{Var}(\varphi)$ be and distinct variables. We define an R-formula φ' as follows:

$$
\begin{aligned}
\varphi' = {} & \bigwedge_{i=1}^{n} R(w_i, x_i, y_i, z_i) \wedge \bigwedge_{x \in \mathrm{Var}(\varphi)} R(f, x, x', t) \\
& \wedge \bigwedge_{i=1}^{k} R(f, f, f, f_i) \wedge R(t, t, t, t_i) \\
& \wedge \bigwedge_{i=1}^{m_0} R(f, f, f, v_i) \wedge \bigwedge_{i=1}^{m_1} R(t, t, t, u_i).
\end{aligned}
$$

This construction is very similar to the one in the proof of Theorem 4.18. The correctness of this reduction can be shown with analogous arguments. Note that R is closed under negation, therefore the number of balanced solutions for φ' is exactly twice the number of solutions for φ, and hence the reduction is both counting and many-one. □

By inspection of Figure 1, it can be verified that with the results in this section, together with our polynomial-time result, we have covered all classes in Post's lattice (indeed the cases that are apparently not treated are dual to known cases), and hence completed the proof of Theorem 4.1, which as mentioned is equivalent to our main Theorem 2.2.

5. CONCLUSION

We have obtained complete complexity classifications for constraint satisfaction problems that mix local constraints with a global one. We have demonstrated that the weak base method is indeed a useful tool in order to get complexity results for these hybrid CSPs. Our contribution is twofold. On the one hand, our work shows an application of a new Galois connection for studying the complexity of constraint satisfaction problems. This is interesting on its own. This illuminates the potential of this new Galois connection and hopefully will popularize it.

On the other hand, our results represent a first encouraging step in the study of global constraints in the framework of non-uniform CSPs. A systematic treatment of global constraints will require an appropriate framework and to develop adequate tools. It is somewhat surprising that for the two global constraints considered in this paper, namely balanced solutions and solutions with a variable number of 1s, we achieve the same complexity classification. This is unexpected, since being able to specify the number of ones required in the solution as part of the input seems to be a much stronger requirement than only to demand that the solutions are balanced. The complexity remains the same even if we consider the counting versions of these problems. This suggests that a comparison of the expressive power and related complexity of different global constraints might be very interesting. The discussion before Lemma 4.5 also shows that this issue is worth investigating.

Finally, as we said in the introduction, balanced assignments arise naturally in many optimization problems. For this reason, as discussed in [Bazgan and Karpinski 2005], it is natural to investigate the approximability of the balanced optimization problem, BAL-MAX-CSP(Γ). The classification of the approximability of this problem is still an open question. We believe that if such a non-trivial complete classification can be achieved, then it will not follow Post's lattice (as it is already proved for the MAX-CSP(Γ) problem with no balance requirement, see [Creignou 1995; Creignou and Vollmer 2008]).

Acknowledgment

We thank the anonymous referees for comments and corrections that helped to improve the presentation of our results.

REFERENCES

ALLENDER, E., BAULAND, M., IMMERMAN, N., SCHNOOR, H., AND VOLLMER, H. 2005. The complexity of satisfiability problems: Refining Schaefer's theorem. In *Proceedings Mathematical Foundations of Computer Science*. Lecture Notes in Computer Science, vol. 3618. Springer-Verlag, 71–82.

BAULAND, M. AND HEMASPAANDRA, E. 2005. Isomorphic implication. In *Proceedings Mathematical Foundations of Computer Science*. Lecture Notes in Computer Science, vol. 3618. Springer-Verlag, Berlin Heidelberg, 119–130. To appear in *Theory of Computing Systems*.

BAZGAN, C. AND KARPINSKI, M. 2005. On the complexity of global constraint satisfaction. In *Proceedings of ISAAC 2005*. Springer, 624–633.

BLÄSER, M., HEYNEN, T., AND MANTHEY, B. 2008. Adding cardinality constraints to integer programs with applications to maximum satisfiability. *Information Processing Letters 105*, 194–198.

BODNARCHUK, V. G., KALUŽNIN, L. A., KOTOV, V. N., AND ROMOV, B. A. 1969. Galois theory for Post algebras. I, II. *Cybernetics 5*, 243–252, 531–539.

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

Böhler, E., Hemaspaandra, E., Reith, S., and Vollmer, H. 2002. Equivalence and isomorphism for Boolean constraint satisfaction. In *Computer Science Logic*. Lecture Notes in Computer Science, vol. 2471. Springer Verlag, Berlin Heidelberg, 412–426.

Böhler, E., Hemaspaandra, E., Reith, S., and Vollmer, H. 2004. The complexity of Boolean constraint isomorphism. In *Proceedings 21st Symposium on Theoretical Aspects of Computer Science*. Lecture Notes in Computer Science, vol. 2996. Springer Verlag, Berlin Heidelberg, 164–175.

Böhler, E., Reith, S., Schnoor, H., and Vollmer, H. 2005. Bases for Boolean co-clones. *Information Processing Letters 96*, 59–66.

Creignou, N. 1995. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences 51*, 511–522.

Creignou, N. and Hermann, M. 1996. Complexity of generalized satisfiability counting problems. *Information and Computation 125*, 1–12.

Creignou, N., Kolaitis, P., and Zanuttini, B. 2008. Structure identification for Boolean relations and plain bases for co-clones. *Journal of Computer and System Sciences 74*, 1103–1115.

Creignou, N. and Vollmer, H. 2008. Boolean constraint satisfaction problems: when does Post's lattice help? In *Complexity of Constraints*, N. Creignou, P. G. Kolaitis, and H. Vollmer, Eds. Vol. 5250. Springer Verlag, Berlin Heidelberg, 3–37.

Creignou, N. and Zanuttini, B. 2006. A complete classification of the complexity of propositional abduction. *SIAM Journal on Computing 36,* 1, 207–229.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman, New York.

Geiger, D. 1968. Closed systems of functions and predicates. *Pac. J. Math 27,* 1, 95–100.

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

Khanna, S., Sudan, M., Trevisan, L., and Williamson, D. 2001. The approximability of constraint satisfaction problems. *SIAM Journal on Computing 30*, 1863–1920.

Kirousis, L. M. and Kolaitis, P. G. 2003. The complexity of minimal satisfiability problems. *Information and Computation 187,* 1, 20–39.

Marx, D. 2005. Parameterized complexity of constraint satisfaction problems. *Computational Complexity 14,* 2, 153–183.

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

Reith, S. and Vollmer, H. 2003. Optimal satisfiability for propositional calculi and constraint satisfaction problems. *Information and Computation 186,* 1, 1–19.

Romov, B. A. 1981. The algebras of partial functions and their invariants. *Cybernetics and Systems Analysis 17,* 2, 157–167.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*. ACM Press, 216–226.

Schnoor, H. and Schnoor, I. 2008. Partial polymorphisms and constraint satisfaction problems. In *Complexity of Constraints*, N. Creignou, P. G. Kolaitis, and H. Vollmer, Eds. Vol. 5250. Springer Verlag, Berlin Heidelberg, 229–254.

Sviridenko, M. I. 2001. Best possible approximation algorithm for MAX-SAT with cardinality constraint. *Algorithmica 30,* 3, 398–405.

Toda, S. and Watanabe, O. 1992. Polynomial time 1-Turing reductions from #PH to #P. *Theoretical Computer Science 100*, 205–221.

Zankó, V. 1991. #P-completeness via many-one reductions. *International Journal of Foundations of Computer Science 2*, 77–82.

Received September 2008; accepted January 2009