Descriptive complexity of approximate counting CSPs

Andrei Bulatov*[1], Victor Dalmau†[2], and Marc Thurley[3]

1 **School of Computing Science, Simon Fraser University, Burnaby, Canada**
`abulatov@sfu.ca`

2 **Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain**
`victor.dalmau@upf.edu`

3 **Oracle, Buenos Aires, Argentina**
`marc.thurley@googlemail.com`

Abstract

Motivated by Fagin's characterization of NP, Saluja et al. have introduced a logic based framework for expressing counting problems. In this setting, a counting problem (seen as a mapping $\mathcal{C}$ from structures to non-negative integers) is 'defined' by a first-order sentence φ if for every instance $\mathbf{A}$ of the problem, the number of possible satisfying assignments of the variables of φ in $\mathbf{A}$ is equal to $\mathcal{C}(A)$. The logic $\mathrm{RH}\Pi_1$ has been introduced by Dyer et al. in their study of the counting complexity class #BIS. The interest in the class #BIS stems from the fact that, it is quite plausible that the problems in #BIS are not #P-hard, nor they admit a fully polynomial randomized approximation scheme. In the present paper we investigate which counting constraint satisfaction problems $\#\mathrm{CSP}(\mathbf{H})$ are definable in the monotone fragment of $\mathrm{RH}\Pi_1$. We prove that $\#\mathrm{CSP}(\mathbf{H})$ is definable in monotone $\mathrm{RH}\Pi_1$ whenever $\mathbf{H}$ is invariant under meet and join operations of a distributive lattice. We prove that the converse also holds if $\mathbf{H}$ contains the equality relation. We also prove similar results for counting CSPs expressible by linear Datalog. The results in this case are very similar to those for monotone $\mathrm{RH}\Pi_1$, with the addition that $\mathbf{H}$ has, additionally, $\top$ (the greatest element of the lattice) as a polymorphism.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.4.1 Mathematical Logic

Keywords and phrases Constraint Satisfaction Problems, Approximate Counting, Descriptive Complexity.

Digital Object Identifier 10.4230/LIPIcs.CSL.2013.149

1 Introduction

Constraint Satisfaction Problems (CSPs) form a rich class of algorithmic problems with applications in many areas of computer science. In a CSP the goal is to find an assignment to variables subject to specified constraints. It has been observed by Feder and Vardi [19] that CSPs can be viewed as homomorphisms problems: given two relational structures $\mathbf{A}$ and $\mathbf{H}$, decide if there is a homomorphism from $\mathbf{A}$ to $\mathbf{H}$.

In this paper we consider *counting* constraint satisfaction problems (#CSPs), in which the problem of computing the number of solutions of a given CSP instance. Substantial amount

* supported by NSERC Discovery grant
† supported by MICINN grant TIN2010-20967-C04-02

Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca; pp. 149–164
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of attention has been paid in the last decade to the complexity and algorithms for problems of the form CSP(**H**) (see, for example, [1, 3, 5, 8, 23]) and #CSP(**H**) [4, 6, 7, 10, 16, 17, 18, 21], in which the target structure **H** is fixed. In the case of exact counting, there is a complete complexity classification of problems #CSP(**H**) [4, 17, 18], which states that every problem of this form is either solvable in polynomial time, or complete in #P. This classification was recently extended to computing partition functions of weighted homomorphisms [10, 21].

Very few non-trivial counting problems can be solved using a polynomial-time deterministic algorithm. When efficient exact counting is not possible one might try to find a good approximation. Dyer et al. [14] argued that the most natural model of efficient approximation is the one by means of fully polynomial randomized approximation schemes (FPRAS), where the desired approximation error is a part of input, randomization is allowed, and the algorithm must stop within time polynomial in the size of the input and the bound on the approximation error. The approximation complexity of counting problems is then measured through approximation preserving, or AP-reductions, designed so that the class of problems solvable by FPRAS is closed under AP-reductions (more details can be found in §2.3).

The approximation complexity of #CSP(**H**) for 2-element structures **H** is determined in [15]. It turns out that, along with problems admitting a FPRAS (indeed, even solvable exactly in polynomial-time) and #P-hard problems, there are also problems that apparently do not fall into any of these two categories. Furthermore, all the problems that seemingly lie strictly between the class of problems admitting an FPRAS and the class of #P-hard problems are interreducible with each other and with other natural and well-studied problems (see also [14]) such as the problem of counting independent sets in a bipartite graph, denoted #BIS. It is argued in [14] that the set of problems interreducible with #BIS form a separate complexity class different from both FPRAS and #P. This class includes the problem of finding the number of downsets and the problem of finding the number of antichains in a partially ordered set, SAT-based problems such as finding the number of satisfying assignments of a CNF in which every clause is an implication or a unit clause, certain graph homomorphism problems, e.g., BeachConfig [14], and many others.

In this paper we shed light on the complexity of approximate counting CSPs by studying its descriptive complexity. We follow Saluja et al.'s framework [27] for studying the logical definability of counting problems. Let φ be a first-order formula that can have first and second-order free variables. In the setting of [27], a counting problem $\mathcal{C}$ (seen as a mapping from structures over a finite signature τ to non-negative integers) is *defined* by formula φ if, for every structure **A** with signature τ, $\mathcal{C}(\mathbf{A})$ is equal to the number of different interpretations of the free variables that make φ true on **A**.

For example, the problem #IS of counting the number of independent sets of a graph $G = (V, E)$ is defined by the sentence

$$\forall x, y\ (\neg E(x, y) \vee \neg I(x) \vee \neg I(y)),$$

where I is a monadic second order free variable.

It is shown in [27] that, on ordered structures, the class #P coincides with the class #FO of counting problems definable by a first-order formula. Furthermore it is also shown that the expressiveness of subclasses of #FO obtained by restricting the quantifier alternation depth form the strict hierarchy

$$\#\Sigma_0 = \#\Pi_0 \subset \#\Sigma_1 \subset \#\Pi_1 \subset \#\Sigma_2 \subset \#\Pi_2 = \#\mathrm{FO},$$

where $\#L$ denotes the set of counting problems definable by a formula in L.

A different approach to expressing counting problems over graphs in logical terms has been developed in a series of papers by Makowsky et al. (see [24] and the references therein). The framework there is more liberal allowing to define a wide range of graph invariants and polynomials, such as the the chromatic polynomial, various generalizations of the Tutte polynomial, matching polynomials, interlace polynomials, and many others.

Dyer et al. have introduced the logic $\mathrm{RH}\Pi_1 \subseteq \Pi_1$ (to be defined below) in their study of the class of problems AP-interreducible with #BIS. It is shown in [14] that all problems in $\#\mathrm{RH}\Pi_1$ are AP-reducible to #BIS. Also, many problems AP-interreducible with #BIS (for example all problems listed in §2.3.1) are known to be in $\#\mathrm{RH}\Pi_1$. The logic $\mathrm{RH}\Pi_1$ contains all first-order formulas of the form $\forall \mathbf{y}\, \psi$, where ψ is a quantifier-free CNF, in which every clause has at most one occurrence of a unnegated second-order variable and at most one occurrence of a negated second-order variable.

Our main result concerns the monotone fragment of $\mathrm{RH}\Pi_1$, namely, the subset of $\mathrm{RH}\Pi_1$ containing all formulas in which every relation from τ (the signature of the input structure) appears only negatively. It is natural to consider monotone logics in the context of CSP as for every input structure $\mathbf{A}$ of $\#\mathrm{CSP}(\mathbf{H})$, every atomic formula with a predicate from τ holding in $\mathbf{A}$ makes the existence of a homomorphism from $\mathbf{A}$ to $\mathbf{H}$ less likely. Furthermore, it follows from the results of [20] that in the decision variant (that is, if our goal is merely to decide if the number of homomorphisms is greater than zero) monotone $\mathrm{RH}\Pi_1$ is as expressive as full $\mathrm{RH}\Pi_1$.

To tackle our question we consider the algebraic invariance properties of the relations in $\mathbf{H}$, namely, so-called polymorphisms of $\mathbf{H}$ (see §2.1). This approach has lead to impressive progress in the study of the complexity of $\mathrm{CSP}(\mathbf{H})$ and of $\#\mathrm{CSP}(\mathbf{H})$.

We prove that $\#\mathrm{CSP}(\mathbf{H})$ is definable by a monotone $\mathrm{RH}\Pi_1$ formula whenever $\mathbf{H}$ has, as polymorphisms, the meet and join operations of a distributive lattice. We also show that this is the best-possible considering only the algebraic invariants of $\mathbf{H}$. In particular, we prove that if $\#\mathrm{CSP}(\mathbf{H})$ is definable in monotone $\mathrm{RH}\Pi_1$ and $\mathbf{H}$ contains one relation interpreted as the equality then $\mathbf{H}$ must be invariant under the meet and join operations of a distributive lattice. Since the set of polymorphisms of a structure does not change if one adds the equality relation to it, our results imply a complete characterization for the definability of #CSPs in monotone $\mathrm{RH}\Pi_1$ under the assumption that every pair of structures $\mathbf{H}$ and $\mathbf{H}'$ with the same polymorphisms give rise to counting CSPs, $\#\mathrm{CSP}(\mathbf{H})$ and $\#\mathrm{CSP}(\mathbf{H}')$ that are both definable or both undefinable in monotone $\mathrm{RH}\Pi_1$. Although the majority of the properties of CSPs investigated so far are completely determined by the algebraic invariants (see [26]), there are some which are not [25].

As a byproduct of our main result we obtain a similar characterization of definability on the fragment of monotone $\mathrm{RH}\Pi_1$ known as linear Datalog. Linear Datalog has been investigated in the decision CSPs as a tool to show the membership in NL [2, 11, 12, 13]. For our purposes, linear Datalog is precisely the class of monotone $\mathrm{RH}\Pi_1$-formulas that do not contain free first-order variables and where every clause of its quantifier-free part contains an unnegated second-order variable. We show that $\#\mathrm{CSP}(\mathbf{H})$ is definable in linear Datalog if $\mathbf{H}$ has, as polymorphisms, the join, meet, and top operations of a distributive lattice and that the converse holds whenever $\mathbf{H}$ contains the equality relation.

2 Preliminaries

2.1 Basic definitions

Let A be a finite set. A k-ary tuple $(a_1, \ldots, a_k)$ over A is any element of A^k. We shall use boldface letters to denote tuples of any length. A k-ary relation on A is a collection of k-ary tuples over A or, alternatively, a subset of A^k. A *relational signature* (also relational vocabulary) τ is a collection of relational symbols (also called predicates), in which every symbol has an associated arity. A *(relational) structure* $\mathbf{A}$ with signature τ (also called τ*-structure*) consists of a set A called the *universe* of $\mathbf{A}$, and for each symbol $R \in \tau$, say of arity k, a k-ary relation $R^{\mathbf{A}}$ on A, called the *intepretation* of R in $\mathbf{A}$. We shall use the same boldfaced and slanted capital letters to denote a structure and its universe, respectively. In this paper all signatures and structures are finite. A *fact* of a relational structure is any atomic formula holding in it. Sometimes we will regard relational structures as a universe and a collection of facts on it.

Let R be a relation on a set A and $f\colon A^n \to A$ an n-ary operation on the same set. Operation f is said to be a *polymorphism* of R if for any choice $\mathbf{a}_1, \ldots, \mathbf{a}_n$ of tuples from R the tuple $f(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ obtained by applying f component-wise also belongs to R. Then, it is also said that R is *invariant* under f. Operation f is a polymorphism of a relational structure $\mathbf{A}$ if it is a polymorphism of every relation in $\mathbf{A}$.

Let A, B be finite sets and let $f\colon A \to B$. For every tuple $\mathbf{a}$ on A we use $f(\mathbf{a})$ to denote the tuple on B obtained by applying f to $\mathbf{a}$ component-wise. Similarly, for every relation R on A we use $f(R)$ to denote $\{f(\mathbf{a}) \mid \mathbf{a} \in R\}$. Let $\mathbf{A}, \mathbf{B}$ be relational structures of the same signature with universes A and B, respectively. Mapping f is said to be a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if for any symbol R from τ, $f(R^{\mathbf{A}}) \subseteq R^{\mathbf{B}}$. If, furthermore, $B \subseteq A$ and f acts as the identity on B then f is said to be a *retraction*. A homomorphism f from $\mathbf{A}$ to $\mathbf{B}$ is said to be an *isomorphism* if it is bijective and f^{-1} is a homomorphism from $\mathbf{B}$ to $\mathbf{A}$.

A *lattice* $\mathbf{H}$ is a structure with a universe H equipped with two binary operations $\sqcap, \sqcup\colon H \times H \to H$ (see, e.g., [22]) satisfying the following conditions for any $x, y, z \in H$: (1) $x \sqcap x = x \sqcup x = x$, (2) $x \sqcap y = y \sqcap x$, $x \sqcup y = y \sqcup x$, (3) $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$, $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$, (4) $x \sqcap (x \sqcup y) = x \sqcup (x \sqcap y) = x$. Lattice $\mathbf{H}$ is said to be *distributive* if it satisfies an additional equation $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$. Every lattice has an associated partial order $\leq$ on its universe given by $x \leq y$ if and only if $x \sqcap y = x$.

2.2 Constraint satisfaction problem

For a relational structure $\mathbf{H}$ an instance of the *constraint satisfaction problem* CSP($\mathbf{H}$) is a structure $\mathbf{A}$ of the same signature. The goal in CSP($\mathbf{H}$) is to decide whether or not there is a homomorphism from $\mathbf{A}$ to $\mathbf{H}$. In the *counting constraint satisfaction problem* #CSP($\mathbf{H}$) the objective is to find the number of such homomorphisms.

▶ **Example 1.** In a 3-SAT problem we are given a propositional formula φ in conjunctive normal form whose clauses contain 3 literals (3-CNF). The task is to decide if φ is satisfiable. As is easily seen, the 3-SAT problem is equivalent to CSP($\mathbf{H}_{3-\mathsf{SAT}}$), where $\mathbf{H}_{3-\mathsf{SAT}}$ is the relational structure with universe $\{0,1\}$ that contains, for every $a, b, c \in \{0,1\}$, the relation $R_{a,b,c} = \{0,1\}^3 \setminus \{(a,b,c)\}$. In the counting version of 3-SAT, denoted #3-SAT, the goal is to find the number of satisfying assignments of a 3-CNF formula. Clearly, this problem can be represented as #CSP($\mathbf{H}_{3-\mathsf{SAT}}$).

▶ **Example 2.** Let F be a finite field. The LINEAR SYSTEM(F) problem over F is the problem of checking the consistency of a given system of linear equations over F. This

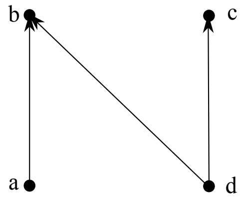

Figure 1 The $\mathbf{H}_{\mathsf{BIS}}$ graph.

problem cannot be represented as a CSP because the set of possible linear equations that can appear in an instance is infinite, while we only allow finite structures. However, for any system of linear equations one can easily obtain an equivalent system in which every equation contains at most 3 variables (although it may be necessary to introduce new variables). Hence, LINEAR SYSTEM(F) reduces to the restricted problem 3-LINEAR SYSTEM(F), in which only equations with 3 variables are allowed. For every $\alpha, \beta, \gamma, \delta \in F$, denote by $R_{\alpha\beta\gamma\delta}$, the ternary relation that contains all tuples $(x, y, z) \in F^3$ satisfying the equation $\alpha x + \beta y + \gamma z = \delta$. Then the 3-LINEAR SYSTEM(F) problem can be represented as CSP($\mathbf{H}_{\mathsf{LIN}}$), where $\mathbf{H}_{\mathsf{LIN}}$ is the relational structure with universe F equipped with all relations $R_{\alpha\beta\gamma\delta}$, $\alpha, \beta, \gamma, \delta \in F$. The counting version of this problem, #CSP($\mathbf{H}_{\mathsf{LIN}}$), concerns finding the number of solutions of a system of linear equations.

2.3 Counting and approximation

Counting CSPs is a particular case of *counting problems*. For every problem $\mathcal{L}$ in NP, one can associate a corresponding counting problem; namely, the problem of counting the accepting paths of a nondeterministic Turing machine deciding $\mathcal{L}$ in polynomial time. The set of problems defined this way is denoted by #P.

▶ **Example 3.** In the counting Bipartite Independent Set problem (#BIS) we are given a bipartite graph $\mathbf{G}$ and asked to find the number of independent sets in $\mathbf{G}$. Let $\mathbf{H}_{\mathsf{BIS}}$ be the digraph shown in Fig. 1. Given a bipartite graph $\mathbf{G}$ with bipartition (V_1, V_2) let $\mathbf{G}'$ be the digraph obtained by orienting all edges from V_1 to V_2. As is easily seen, the number of homomorphisms from $\mathbf{G}'$ to $\mathbf{H}_{\mathsf{BIS}}$ equals the number of independent sets in $\mathbf{G}$, as the preimage of $\{a, c\}$ is an independent set of $\mathbf{G}$. Thus, #BIS can be easily 'reduced' to #CSP($\mathbf{H}_{\mathsf{BIS}}$), but it is not clear if it can be represented as a counting CSP.

Algorithms and the complexity of counting problems, including counting CSPs, have attracted considerable amount of attention starting from the seminal paper by Valiant [28]. The complexity of exact counting CSPs is largely known, see, [4, 17, 18]. Every problem of the form #CSP($\mathbf{H}$) is either solvable in polynomial time or is complete in #P under polynomial time reductions[1].

The approximation complexity of #CSP($\mathbf{H}$) is much more diverse. Let $\mathcal{C}$ be a counting problem and, for an instance I of $\mathcal{C}$, let us denote the solution of I by $\#I$. For $\varepsilon > 0$, a randomized algorithm Alg is said to be an *ε-approximating algorithm* for the problem $\mathcal{C}$ if for any instance I of $\mathcal{C}$ it returns a number $\mathsf{Alg}(I)$ such that

$$\mathsf{Pr}\left[e^{-\varepsilon} < \frac{\mathsf{Alg}(I)}{\#I} < e^{\varepsilon}\right] \geq \frac{2}{3}.$$

[1] In fact, the class #P is not closed under polynomial time reductions; therefore it is technically more correct to say that these problems are complete in $\mathrm{FP}^{\#\mathrm{P}}$

Arguably, the most general, but still practical type of approximation algorithm for counting problems is *fully polynomial randomized approximation schemes* (*FPRAS*): An algorithm Alg is said to be an FPRAS for a counting problem $\mathcal{C}$ if it takes as input an instance I of $\mathcal{C}$ and a number $\varepsilon > 0$, outputs a number $\mathsf{Alg}(I, \varepsilon)$ satisfying the inequality above, and works in time polynomial in $|I|$ and $\log \frac{1}{\varepsilon}$. To compare the relative complexity of approximating counting problems one uses *approximation preserving reduction* (or *AP-reduction* for short). If $\mathcal{A}$ and $\mathcal{B}$ are counting problems, an AP-reduction from $\mathcal{A}$ to $\mathcal{B}$ is a probabilistic algorithm Alg, using $\mathcal{B}$ as an oracle, that takes as input a pair (I, ε) where I is an instance of $\mathcal{A}$ and $0 < \varepsilon < 1$, and satisfies the following three conditions: (i) every oracle call made by Alg is of the form (I', δ), where I' is an instance of $\mathcal{B}$, and $0 < \delta < 1$ is an error bound such that $\log \frac{1}{\delta}$ is bounded by a polynomial in the size of I and $\log \frac{1}{\varepsilon}$; (ii) the algorithm Alg meets the specifications for being approximation scheme for $\mathcal{A}$ whenever the oracle meets the specification for being approximation scheme for $\mathcal{B}$; and (iii) the running time of Alg is polynomial in the size of I and $\log \frac{1}{\varepsilon}$. If an AP-reduction from $\mathcal{A}$ to $\mathcal{B}$ exists we write $\mathcal{A} \leq_{\mathrm{AP}} \mathcal{B}$, and say that $\mathcal{A}$ is *AP-reducible to* $\mathcal{B}$.

2.3.1 The class of problems AP-interreducible with #BIS

The two most natural approximation complexity classes are FPRAS, the class of problems solvable by an FPRAS, and the class $\mathrm{FP}^{\#\mathrm{P}}$, the class of problems AP-interreducible with #SAT (note that #P is not closed under AP-reductions). In [14] Dyer et al. argued that #BIS (see Example 3) defines a class of its own: No FPRAS is known for this problem, and it is not believed to be interreducible with #SAT. There are many natural and well studied problems that are AP-interreducible with #BIS. The following list contains some examples:

- #DOWNSET. A downset in a partial order $(P, \leq)$ is a set $A \subseteq P$ such that whenever $b \in A$ and $a \leq b$, the element a belongs to A. The #DOWNSET problem asks, given a partial order $(P, \leq)$ to find the number of downsets in P.
- #ANTICHAIN. An antichain in a partial order $(P, \leq)$ is a set $C \subseteq P$ such that $a \leq b$ for no $a, b \in C$. In the #ANTICHAIN problem we are required, given a partial order $(P, \leq)$, to find the number of antichains. #ANTICHAIN and #DOWNSET are essentially the same problem. Clearly, every downset $A \subseteq P$ is determined by the set of its maximal elements that form an antichain. Conversely, if $C \subseteq P$ is an antichain then the set $\{a \in P \mid a \leq b \text{ for some } b \in C\}$ is a downset.
- #IMPLICATION. Let φ be a 2-CNF, in which every clause is of the form $\neg x \vee y$, or, equivalently, $x \to y$. In the #IMPLICATION problem, given such a 2-CNF, the goal is to compute the number of its satisfying assignments. There are easy AP-reductions between #DOWNSET and #IMPLICATION. In one direction, the downsets of a partial order $(P, \leq)$ are exactly the satisfying assignments of the formula that includes clause $b \to a$ for every pair $a, b \in P$ with $a \leq b$. For the opposite direction, every instance φ of #IMPLICATION can be represented as a digraph $G(\varphi)$, in which the nodes are the variables of φ and edges (x, y) correspond to clauses $x \to y$. The set of strongly connected components $P(\varphi)$ of $G(\varphi)$ can be equipped with the natural partial order: $U_1 \leq U_2$ for $U_1, U_2 \in P(\varphi)$ if and only if there is a directed path from a node from U_2 to a node from U_1. It is straightforward to see that the number of satisfying assignments of φ equals the number of downsets in $P(\varphi)$.

  Also, #IMPLICATION is precisely $\#\mathrm{CSP}(\mathbf{H}_{\mathsf{IMP}})$, where $\mathbf{H}_{\mathsf{IMP}}$ is the digraph shown in Fig. 2.

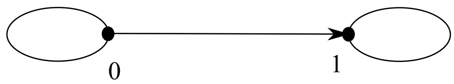

Figure 2 The $\mathbf{H}_{\mathsf{IMP}}$ digraph.

A classification of problems of the form #CSP($\mathbf{H}$) for 2-element structures $\mathbf{H}$ according to their approximation complexity given in [15] provides another evidence of the significance of #BIS. Indeed, every such problem turns out to be either solvable exactly in polynomial time (and so belongs to FPRAS), or is AP-interreducible with #SAT, or else is AP-interreducible with #BIS.

2.4 Descriptive complexity of (approximate) counting problems

Motivated by Fagin's characterization of NP, Saluja et al. [27] have introduced a logic based framework for expressing counting problems. In what follows we describe the setting of [27].

Let τ and σ be finite signatures, let $\mathcal{C}$ be a counting problem (seen as a mapping from τ-structures to non-negative integers), let $\varphi(\mathbf{z})$ be a first-order formula with signature $\tau \cup \sigma$ with free (first-order) variables $\mathbf{z}$, and let $\mathbf{A}$ be a τ-structure. Formula φ is *monadic* if all predicates in σ have arity at most one. An $\mathbf{A}$-*assignment* for φ (or just an assignment, if $\mathbf{A}$ and φ are clear) is a pair $(\mathbf{T}, \mathbf{a})$ where $\mathbf{T}$ and $\mathbf{a}$ are interpretations of σ and $\mathbf{z}$, respectively, over the universe, A, of $\mathbf{A}$. We write $(\mathbf{A}, \mathbf{T})$ to denote the $(\tau \cup \sigma)$-structure with universe A where every $R \in \tau$ is interpreted as in $\mathbf{A}$ and every $I \in \sigma$ is interpreted as in $\mathbf{T}$. We say that $(\mathbf{T}, \mathbf{a})$ *satisfies* φ if $(\mathbf{A}, \mathbf{T}) \models \varphi(\mathbf{a})$ that is, if $\varphi(\mathbf{a})$ is true on the structure $(\mathbf{A}, \mathbf{T})$. We say that φ *defines* $\mathcal{C}$ if for every τ-structure $\mathbf{A}$

$$\mathcal{C}(\mathbf{A}) = |\{(\mathbf{T}, \mathbf{a}) \mid (\mathbf{A}, \mathbf{T}) \models \varphi(\mathbf{a})\}|$$

We note here that we deviate—although only formally—from the framework in Sajula et al. in the following sense: we use predicate symbols in σ to represent second-order variables. Hence, our first-order formulas only have, formally, first-order free variables.

We shall denote by #FO the set of all counting problems definable by a first-order formula. For every fragment L of FO we define $\#L$ as the set of all counting problems definable by a formula in L. An structure $\mathbf{A}$ is ordered if it has a binary relation that is interpreted as a total order on the universe.

▶ **Theorem 4** ([27]). *On ordered structures, the class #P coincides with the class #FO. In fact, #P is the class of all counting problems definable with a* Π_2 *formula.*

Saluja et al. [27] study the expressiveness of subclasses of #FO obtained by restricting the quantifier alternation depth obtaining the strict hierarchy

$$\#\Sigma_0 = \#\Pi_0 \subset \#\Sigma_1 \subset \#\Pi_1 \subset \#\Sigma_2 \subset \#\Pi_2 = \#\mathrm{FO}$$

Dyer et al. [14] introduced the fragment $\mathrm{RH}\Pi_1 \subseteq \Pi_1$ in their study of the complexity class of problems AP-interreducible with #BIS. A first-order formula $\varphi(\mathbf{z})$ with signature $\tau \cup \sigma$ is in $\mathrm{RH}\Pi_1$ if it is of the form $\forall \mathbf{y} \psi(\mathbf{y}, \mathbf{z})$ where ψ is a quantifier-free CNF in which every clause has at most one occurrence of an unnegated relation symbol from σ and at most one occurrence of a negated symbol from σ. The part Π_1 in notation $\#\mathrm{RH}\Pi_1$ indicates that the formula involves only universal quantification, and RH indicates that ψ is in 'restricted Horn' form.

It is shown in [14] that all problems in $\#\mathrm{RH}\Pi_1$ are AP-reducible to #BIS. Also, many problems AP-interreducible with #BIS (for example all problems listed in §2.3.1) are known to be in $\#\mathrm{RH}\Pi_1$.

▶ **Example 5.** Consider the problem #DOWNSET. By encoding a partial order $(P, \leq)$ as a structure with a binary relation we can define #DOWNSET with the $\mathrm{RH}\Pi_1$-sentence

$$\forall x, y\ (I(x) \vee \neg(x \leq y) \vee \neg I(y))$$

Our ultimate goal is to characterize under which circumstances $\#\mathrm{CSP}(\mathbf{H})$ belongs to $\#\mathrm{RH}\Pi_1$. Towards this end, in this paper we consider the fragment of $\mathrm{RH}\Pi_1$ obtained by requiring that, in addition, the predicates from τ occur only negatively. The resulting logic is called *monotone* $\mathrm{RH}\Pi_1$.

It makes sense to restrict to monotone formulas in the context of problems of the form $\#\mathrm{CSP}(\mathbf{H})$ as the addition of more facts to an input structure cannot increase the number of homomorphisms. Indeed, all problems listed in §2.3.1 are also definable by a formula in monotone $\mathrm{RH}\Pi_1$. Furthermore, it follows from the results of [20] that, if our goal is merely to decide if the number of homomorphisms is greater than zero, monotone $\mathrm{RH}\Pi_1$ is as expressive as $\mathrm{RH}\Pi_1$. Additionally we will deal exclusively with unordered structures as the analysis for ordered structures becomes much more complicated.

We say that a τ-structure $\mathbf{H}$ *contains equality* if τ contains a binary relational symbol eq that is interpreted as the equality on the set H (that is, $\mathbf{H}^{\mathrm{eq}} = \{(b, b) \mid b \in H\}$). Note that we do not require that eq is interpreted as the equality in the instances of $\#\mathrm{CSP}(\mathbf{H})$.

We are now in a position to state the main result of the paper.

▶ **Theorem 6.** *For every structure* $\mathbf{H}$ *the following holds:*
1. *If* $\mathbf{H}$ *has polymorphisms* $x \sqcap y$ *and* $x \sqcup y$ *for some distributive lattice* $(H; \sqcap, \sqcup)$ *then there exists a monotone* $\mathrm{RH}\Pi_1$*-formula defining* $\#\mathrm{CSP}(\mathbf{H})$*.*
2. *Furthermore, if* $\mathbf{H}$ *contains equality then the converse also holds.*

Observe that since the set of polymorphisms of a structure $\mathbf{H}$ does not change if one adds the equality relation to it, it follows that the sufficient condition of Theorem 6 is the best it can be achieved by considering only the algebraic invariants of $\mathbf{H}$. Also, note that it follows from Theorem 6 that the problem of deciding whether for a relational structure $\mathbf{H}$ containing equality the problem $\#\mathrm{CSP}(\mathbf{H})$ is definable in monotone $\mathrm{RH}\Pi_1$ belongs to NP. Indeed, after guessing lattice operation $\sqcap, \sqcup$ on the universe of $\mathbf{H}$, it is polynomial time to verify that these binary operations are polymorphisms of the structure.

3 Reduction to the monadic case

In this section, as a first step toward proving the necessary condition of Theorem 6, we prove the following proposition.

▶ **Proposition 7.** *For every structure* $\mathbf{H}$*, if* $\#\mathrm{CSP}(\mathbf{H})$ *is definable in monotone* $\mathrm{RH}\Pi_1$ *then it is also definable by a monadic monotone formula from* $\mathrm{RH}\Pi_1$ *without free variables.*

In what follows, τ and σ are finite vocabularies, $\mathbf{H}$ is a τ-structure, and $\varphi(\mathbf{z})$ is a monotone $\mathrm{RH}\Pi_1$-formula with signature $\tau \cup \sigma$ defining $\#\mathrm{CSP}(\mathbf{H})$.

For every $n \geq 1$, define $\mathbf{Isol}_n$ (from *isolated nodes*) to be the τ-structure with universe $\{1, \ldots, n\}$, where all relations are interpreted as the empty set.

▶ **Lemma 8.** φ *is a sentence (i.e, has no free variables).*

Proof. Consider structure $\mathbf{Isol}_p$ where p is a prime number that does not divide $|H|$. Let k be the number of free variables of φ and for every $\mathbf{a} \in A^k$ let $n(\mathbf{a})$ be

$$|\{(\mathbf{T}, \mathbf{a}) \mid (\mathbf{Isol}_p, \mathbf{T}) \models \varphi(\mathbf{a})\}|$$

Clearly $\sum_{\mathbf{a}\in A^k} n(\mathbf{a}) = |H|^p$. Consider the following equivalence relation θ on A^k: two tuples $\mathbf{a}, \mathbf{a}' \in A^k$ are θ-related if $\mathbf{a}' = h(\mathbf{a})$ for some bijection $h : A \to A$. Clearly, if $\mathbf{a}$ and $\mathbf{a}'$ are θ-related then $n(\mathbf{a}) = n(\mathbf{a}')$.

For every $\mathbf{a} \in A^k$ we shall denote by $\mathbf{a}_\theta$ the θ-class containing $\mathbf{a}$. Hence $|H|^p = \sum_{\mathbf{a}\in A^k} n(\mathbf{a}) = \sum_{\mathbf{a}_\theta\in(A^k)_\theta} |\mathbf{a}_\theta| \cdot n(\mathbf{a})$, where $(A^k)_\theta$ denotes the set of all θ-classes. Note that $|\mathbf{a}_\theta| = p(p-1)\cdots(p-m+1)$, where m is the number of different elements in $\mathbf{a}$. We are in a position to show that $k = 0$. If $k > 0$ then p divides $|\mathbf{a}_\theta|$ for any $\mathbf{a}$ and hence p divides $\sum_{\mathbf{a}\in A^k} n(\mathbf{a}) = |H|^p$, but p does not divide $|H|$, a contradiction. ◀

Consequently, from now on we can assume that φ is a sentence. Let $\mathbf{A}$ be a τ-structure and let $\mathbf{T}$ be any $\mathbf{A}$-assignment. It will be convenient to regard, alternatively, $\mathbf{T}$ as the collection of all atomic formulas $I(\mathbf{a})$ that hold in $\mathbf{T}$.

The following lemma is a direct consequence of the fact that every clause of an RHΠ_1-formula has at most one occurrence of an unnegated relation symbol from σ and at most one occurrence of a negated symbol from σ.

▶ **Lemma 9.** *Let $\mathbf{A}$ be a τ-structure. The set of all $\mathbf{A}$-assignments (seen as a collection of facts) satisfying φ is closed under union and intersection.*

▶ **Lemma 10.** *Let I be any predicate in σ with arity $k \geq 2$. Then*

$$\varphi \models \forall x_1, \ldots, x_k, y_1, \ldots, y_k \quad \neg(x_i = y_i) \vee \neg I(x_1, \ldots, x_k) \vee I(y_1, \ldots, y_k)$$

for some $1 \leq i \leq k$

Proof. For every $L \subseteq \{1, \ldots, k\}$, we define μ_L to be the sentence

$$\forall x_1, \ldots, x_k, y_1, \ldots, y_k \quad (\bigvee_{i\in L} \neg(x_i = y_i)) \vee \neg I(x_1, \ldots, x_k) \vee I(y_1, \ldots, y_k).$$

Observe that μ_L expresses the fact that $I(z_1, \ldots, z_k)$ only depends on the variables $z_i, i \in L$.

It follows easily that $\mu_J \wedge \mu_K \models \mu_L$ for every $J, K, L \subseteq \{1, \ldots, k\}$ with $J \cap K \subseteq L$. Note that the sentences appearing in the statement of the lemma are precisely the class of all sentences of the form μ_L where L is a singleton. Hence, the lemma follows by a direct application of the above property provided we are able to show the following:

For every different $i, j \in \{1, \ldots, k\}$ there exists L with $\{i, j\} \not\subseteq L$ and such that $\varphi \models \mu_L$.

To simplify the notation we shall prove the claim only for $i = k-1$ and $j = k$. Consider the structure $\mathbf{Isol}_{2n+k-2}$ with n large enough. Let X be the set containing all atomic formulas of the form $I(1, \ldots, k-2, a, b)$ where $a \in \{k-1, \ldots, n+k-2\}$ and $b \in \{n+k-1, \ldots, 2n+k-2\}$.

We claim that there exists some atomic formula $I(1, \ldots, k-2, a, b) \in X$ such that every satisfying assignment containing $I(1, \ldots, k-2, a, b)$ contains also some other atomic formula in X. Indeed, otherwise, since the set of satisfying assignments is closed under union, we could construct for every non-empty subset $Y \subseteq X$ a satisfying assignment containing all atomic predicates from Y and none of the atomic predicates from $Y \setminus X$. This would lead to a contradiction, as the set of satisfying assignments would be at least 2^{n^2}, which grows asymptotically faster than $|H|^{2n+k-2}$ (the number of homomorphisms from $\mathbf{Isol}_{2n+k-2}$ to $\mathbf{H}$).

Thus there exists an atomic formula $I(1,\ldots,k-2,a,b) \in X$ such that every satisfying assignment containing $I(1,\ldots,k-2,a,b)$ contains also some other atomic formula in X. Consider first the case in which there exists at at least one satisfying assignment containing $I(1,\ldots,k-2,a,b)$. Then, the *smallest*, with respect to inclusion, satisfying assignment containing $I(1,\ldots,k-2,a,b)$ (by Lemma 9 such an assignment exists) also contains some other atomic predicate $I(1,\ldots,k-2,a',b')$ in X. By the monotonicity of φ it follows that φ implies

$$\forall v_1,\ldots,v_{2n+k-2} \quad \neg I(v_1,\ldots,v_{k-2},v_a,v_b) \vee I(v_1,\ldots,v_{k-2},v_{a'},v_{b'}),$$

which after renaming variables is equivalent to μ_L for $L = \{1,\ldots,k-2,k-1\}$ or $L = \{1,\ldots,k-2,k\}$. Secondly, assume that there is no satisfying assignment containing $I(1,\ldots,k-2,a,b)$. Then

$$\varphi \models \forall v_1,\ldots,v_{2n+k-2} \quad \neg I(v_1,\ldots,v_{k-2},v_a,v_b).$$

It follows easily that, in this case, φ implies any formula of the form μ_L. ◀

Proof of Proposition 7. Let $\mathbf{H}$ be a τ-structure and let φ be a monotone RHΠ_1-formula with signature $\tau \cup \sigma$ defining #CSP($\mathbf{H}$). By Lemma 8, φ has no free variables.

Pick any predicate I in σ with arity $k \geq 2$. By Lemma 10

$$\varphi \models \forall x_1,\ldots,x_k,y_1,\ldots,y_k \quad \neg(x_i = y_i) \vee \neg I(x_1,\ldots,x_k) \vee I(y_1,\ldots,y_k)$$

for some $1 \leq i \leq k$. Let σ' be obtained from σ by replacing I by a new unary predicate I', and let φ' be the sentence with signature $\tau \cup \sigma'$ obtained from φ by replacing every atomic formula of the form $I(z_1,\ldots,z_k)$ by $I'(z_i)$. It follows easily that φ' has the same number of satisfying assignments as φ and one non-monadic predicate less. Iterating we obtain a sentence that contains only monadic predicates. ◀

4 Necessary condition

We start with proving item (2) of Theorem 6. In what follows $\mathbf{H}$ is a τ-structure and φ is a monadic monotone RHΠ_1-sentence with signature $\tau \cup \sigma$ defining #CSP($\mathbf{H}$). It will be convenient to assume that σ does not contain 0-ary predicate symbols. This can be achieved by replacing every 0-ary relation symbol $I \in \sigma$ with a new unary relation symbol I', adding to φ the clause $\neg I'(x) \vee I'(y)$, and replacing every atomic formula of the form $R()$ in φ with $R'(x)$ (x and y are bound variables in φ).

Let $\mathbf{A}$ be a τ-structure. Since all the predicate symbols in σ are unary one can establish a bijection from the set of all $\mathbf{A}$-assignments to the set of mappings from A to 2^σ. In particular, we associate with every $\mathbf{A}$-assignment $\mathbf{T}$ a mapping $h : A \to 2^\sigma$ where for every $a \in A$

$$h(a) = \{I \in \sigma \mid I(a) \text{ holds in } \mathbf{T}\}.$$

We shall use $\mathbf{T}_h$ to denote the $\mathbf{A}$-assignment associated with a mapping h. Let also Sol($\mathbf{A},\varphi$) is given by

$$\mathrm{Sol}(\mathbf{A},\varphi) = \{h : A \to 2^\sigma \mid (\mathbf{A},\mathbf{T}_h) \models \varphi\}.$$

▶ **Lemma 11.** *Let* $\mathbf{A}$*,* $\mathbf{B}$ *be* τ*-structures and let* g *be a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$*. For every* $f : B \to 2^\sigma$ *the following holds:*

1. *If* $f \in \mathrm{Sol}(\mathbf{B}, \varphi)$ *then* $f \circ g \in \mathrm{Sol}(\mathbf{A}, \varphi)$*.*
2. *If* $g(A) = B$ *and* $|\,\mathrm{Sol}(\mathbf{B}, \varphi)| = |\,\mathrm{Sol}(\mathbf{A}, \varphi)|$*, then for any* $h \in \mathrm{Sol}(\mathbf{A}, \varphi)$ *there is* $f \in \mathrm{Sol}(\mathbf{B}, \varphi)$ *such that* $h = f \circ g$*.*

Proof. (1) Follows directly from the monotonicity of φ. (2) Since $g(A) = B$ the set $\{f \circ g \mid f \in \mathrm{Sol}(\mathbf{B}, \varphi)\}$ contains $|\,\mathrm{Sol}(\mathbf{B}, \varphi)|$ different mappings and, consequently, $\mathrm{Sol}(\mathbf{A}, \varphi)$ cannot contain any other one. ◀

▶ **Lemma 12.** *Let* $\mathbf{A}$ *be a* τ*-structure, let* $a, a' \in A$*, and let* $\mathbf{B}$ *be the* τ*-structure obtained by adding the fact* $\mathrm{eq}(a, a')$ *to* $\mathbf{A}$*. For every* $f : A \to 2^\sigma$ *the following holds:*

$$f \in \mathrm{Sol}(\mathbf{B}, \varphi) \text{ if and only if } f \in \mathrm{Sol}(\mathbf{A}, \varphi) \text{ and } f(a) = f(a')$$

Proof. We can assume wlog. that $\mathbf{A}$ (and hence $\mathbf{B}$) contains equalities $\mathrm{eq}(a, a)$ and $\mathrm{eq}(a', a')$ as the addition of $\mathrm{eq}(a, a)$ and $\mathrm{eq}(a', a')$ does not alter $\hom(\mathbf{A}, \mathbf{H})$ or $\hom(\mathbf{B}, \mathbf{H})$, and, consequently, it cannot alter $\mathrm{Sol}(\mathbf{A}, \varphi)$ or $\mathrm{Sol}(\mathbf{B}, \varphi)$ either.

($\Rightarrow$) Assume $f \in \mathrm{Sol}(\mathbf{B}, \varphi)$. It follows directly from Lemma 11(1) that $f \in \mathrm{Sol}(\mathbf{A}, \varphi)$ so it only remains to show that $f(a) = f(a')$. Again by Lemma 11(1) it is only necessary to prove the statement in the case when $\mathbf{A}$ does not contain any other fact besides the equalities $\mathrm{eq}(a, a)$ and $\mathrm{eq}(a', a')$. Indeed, if the claim is true for such structure $\mathbf{A}'$ then the identity homomorphisms from $\mathbf{A}'$ to $\mathbf{A}$ witnesses, with help of Lemma 11(1), that it is also true for $\mathbf{A}$. Let $g : A \to A \setminus \{a'\}$ be the mapping that sends a' to a and acts as the identify otherwise. Lemma 11(2) implies that $f = h \circ g$ for some $h \in \mathrm{Sol}(g(\mathbf{B}), \varphi)$ and hence $f(a) = f(a')$.

($\Leftarrow$) Let $\varphi = \forall \mathbf{y} \psi(\mathbf{y})$, let n be the number of variables in $\mathbf{y}$, and let $\mathbf{C}$ be the τ-structure obtained by adding to $\mathbf{A}$ the chain of equalities

$$\mathrm{eq}(a, a_1), \mathrm{eq}(a_1, a_2), \ldots, \mathrm{eq}(a_n, a_{n+1}), \mathrm{eq}(a_{n+1}, a'),$$

where $a_1, \ldots, a_{n+1}$ are new elements not occurring in $\mathbf{A}$. Assume that $f \in \mathrm{Sol}(\mathbf{A}, \varphi)$ and $f(a) = f(a')$, and let $h : C \to 2^\sigma$ be the extension of f that sets $h(a_i) = f(a)$ for every $i = 1, \ldots, n+1$.

We claim that $h \in \mathrm{Sol}(\mathbf{C}, \varphi)$. Let $\mathbf{c}$ be any instantiation of $\mathbf{y}$ over C. There exists some element a_i that does not appear in $\mathbf{c}$. Let $\mathbf{C}'$ be obtained by removing from $\mathbf{C}$ the equalities involving a_i. There is a retraction g from $\mathbf{C}'$ to $\mathbf{A}$ that maps a_j to a if $j \le i$ and to a' otherwise. By Lemma 11(1) $h = f \circ g$ belongs to $\mathrm{Sol}(\mathbf{C}', \varphi)$ and, hence, $(\mathbf{T}_h, \mathbf{C}') \models \psi(\mathbf{c})$. Since a_i does not appear in $\mathbf{c}$ we have $(\mathbf{T}_h, \mathbf{C}) \models \psi(\mathbf{c})$ as well. Since $(\mathbf{T}_h, \mathbf{C}) \models \psi(\mathbf{c})$ holds for every instantiation $\mathbf{c}$ of $\mathbf{y}$, the claim follows.

Let g be any retraction from $\mathbf{C}$ to $\mathbf{B}$ with $g(a_i) \in \{a, a'\}$ for every $i = 1, \ldots, n+1$. Since $h = f \circ g$ it follows from Lemma 11(2) that $f \in \mathrm{Sol}(\mathbf{B}, \varphi)$. ◀

For every $R \in \tau$ of arity, say, k, let $\mathbf{J}_R$ be the τ-structure with universe $\{1, \ldots, k\}$ containing only fact $R(1, \ldots, k)$. Recall the definition of $\mathbf{Isol}_n$ in the beginning of §3. We define $\mathbf{J}_\varphi$ to be the τ-structure with universe $\mathrm{J}_\varphi = \{h(1) \mid h \in \mathrm{Sol}(\mathbf{Isol}_1, \varphi)\}$ such that for every $R \in \tau$

$$R^{\mathbf{J}_\varphi} = \{(h(1), \ldots, h(k)) \mid h \in \mathrm{Sol}(\mathbf{J}_R, \varphi)\}$$

The next two lemmas follow directly from the definition of $\mathbf{J}_\varphi$.

▶ **Lemma 13.** *Let* $\mathbf{A}$ *be any* τ*-structure and let* $f \in \mathrm{Sol}(\mathbf{A}, \varphi)$*. Then* f *is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{J}_\varphi$*.*

Proof. First, let $a \in A$, and $g : \mathbf{Isol}_1 \to \mathbf{A}$ taking 1 to a. By Lemma 11, $f \circ g \in \mathrm{Sol}(\mathbf{Isol}_1, \varphi)$, implying $f(a)$ belongs to the universe of $\mathbf{J}_\varphi$. Similarly, let $R \in \tau$ and let $(a_1, \dots, a_k) \in R^{\mathbf{A}}$. The mapping $i \overset{g}{\mapsto} a_i$ defines a homomorphism from $\mathbf{J}_R$ to $\mathbf{A}$. By Lemma 11, $f \circ g \in \mathrm{Sol}(\mathbf{J}_R, \varphi)$, which is equivalent to say that $R(f(a_1), \dots, f(a_k))$ holds in $\mathbf{J}_\varphi$. ◀

▶ **Lemma 14.** *If* $\mathbf{H}$ *contains equality then* $\mathbf{H}$ *and* $\mathbf{J}_\varphi$ *are isomorphic.*

Proof. Let X and Y be sets, let F be a collection of mappings from X to Y, and let $\mathrm{equiv}(X)$ be the set of all equivalence relations in X. For every $\theta \in \mathrm{equiv}(X)$ we denote by F_θ the collection of all $f \in F$ such that $f(i) = f(j)$ whenever i and j are θ-related.

Let $\mathbf{A}$ be any τ-structure. For every $\theta \in \mathrm{equiv}(A)$ we define $\mathbf{A}_\theta$ to be the structure that is obtained by adding to $\mathbf{A}$ all facts of the form $\mathrm{eq}(a, a')$ where a and a' are θ-related. For every $\theta \in \mathrm{equiv}(A)$ we have

$$|\,\mathrm{Sol}(\mathbf{A}, \varphi)_\theta| = |\,\mathrm{Sol}(\mathbf{A}_\theta, \varphi)| = |\,\mathrm{hom}(\mathbf{A}_\theta, \mathbf{H})| = |\,\mathrm{hom}(\mathbf{A}, \mathbf{H})_\theta|,$$

where the first equality follows from Lemma 12 and the other equalities follow directly from the definitions. Consequently, $\mathrm{Sol}(\mathbf{A}, \varphi)$ and $\mathrm{hom}(\mathbf{A}, \mathbf{H})$ contain the same number of injective mappings. This follows from the fact that the number of injective mappings in $\mathrm{Sol}(\mathbf{A}, \varphi)$ and the number of injective mappings in $\mathrm{hom}(\mathbf{A}, \mathbf{H})$ are completely determined by the values $|\,\mathrm{Sol}(\mathbf{A}, \varphi)_\theta|, \theta \in \mathrm{equiv}(A)$, and $|\,\mathrm{hom}(\mathbf{A}, \mathbf{H})_\theta|, \theta \in \mathrm{equiv}(A)$, respectively, according to the Möbius inversion formula. By setting $\mathbf{A} = \mathbf{H}$ we infer that $\mathrm{Sol}(\mathbf{H}, \varphi)$ contains an injective mapping h that, by Lemma 13, is an homomorphism from $\mathbf{H}$ to $\mathbf{J}_\varphi$. Since $|H| = |\,\mathrm{J}_\varphi\,|$, homomorphism h must be, in fact, a bijective homomorphism. For every relation symbol $R \in \tau$, we have $h(R^{\mathbf{H}}) \subseteq R^{\mathbf{J}_\varphi}$, because h is a homomorphism. We also have $|R^{\mathbf{J}_\varphi}| = |R^{\mathbf{H}}| = |h(R^{\mathbf{H}})|$ where the first equality follows from the definition of $\mathbf{J}_\varphi$ and the second one follows from the fact that h is a bijection. It follows that $h(R^{\mathbf{H}}) = R^{\mathbf{J}_\varphi}$. Consequently, h is an isomorphism. ◀

Proof of Theorem 6(2). Let $\mathbf{H}$ be a τ-structure that contains equality such that $\#\mathrm{CSP}(\mathbf{H})$ is definable in monotone $\mathrm{RH}\Pi_1$. By Lemma 14 $\mathbf{H}$ is isomorphic to $\mathbf{J}_\varphi$. Since by Lemma 9 $\mathbf{J}_\varphi$ has polymorphisms $x \cap y$ and $x \cup y$, the theorem follows. ◀

Lemma 14 fails if $\mathbf{H}$ does not contain equality as the following example shows. Let $\mathbf{H}$ be the digraph with universe $\{0, 1\}$ containing only edge $(0, 1)$. Consider the monotone $\mathrm{RH}\Pi_1$-sentence φ with $\sigma = \{I\}$

$$\forall x, y, z\ (\neg E(x, y) \vee I(x)) \wedge (\neg E(x, y) \vee I(y)) \wedge (\neg E(x, y) \vee \neg E(y, z))$$

It is not difficult to see that φ defines $\#\mathrm{CSP}(\mathbf{H})$ and that $\mathbf{H}$ is not isomorphic to $\mathbf{J}_\varphi$. Still, $\mathbf{H}$ is invariant under the meet and join of a distributive lattice, namely, $(\{0, 1\}, \vee, \wedge)$.

5 Sufficient condition

In this section we shall prove item (1) of Theorem 6. Throughout this section τ is a finite signature and $\mathbf{H}$ is a τ-structure with polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(H; \sqcap, \sqcup)$. Our goal is to show that there exists a monotone (monadic) $\mathrm{RH}\Pi_1$-sentence φ defining $\#\mathrm{CSP}(\mathbf{H})$.

It is well known (see, e.g., [22, Theorem 9, Corollary 11, Corollary 14, Ch. II.1]) that, since $(H; \sqcap, \sqcup)$ is distributive, there is an isomorphism g from $(H; \sqcap, \sqcup)$ to a sublattice of the lattice of subsets of some finite set S. It will be convenient to assume wlog. that $g(\top) = S$ and $g(\bot) \neq \emptyset$ where $\top$ and $\bot$ are the top and bottom elements, respectively, of $(H; \sqcap, \sqcup)$.

Let k be the maximum arity of a relation in τ, and let us define σ to have one monadic predicate for each symbol in S. To simplify notation we shall use the same symbol to represent a member of S and its associate predicate in σ. Sentence φ is defined to be the monotone monadic RHΠ_1-sentence $\forall \mathbf{x}\ \psi(\mathbf{x})$ with signature $\tau \cup \sigma$, where $\mathbf{x}$ has size k and $\psi(\mathbf{x})$ contains all clauses $\chi(\mathbf{x})$ with at most one occurrence of an unnegated symbol from σ and at most one occurrence of a negated symbol from σ such that $(\mathbf{H}, \mathbf{T}_g) \models \forall \mathbf{x}\ \chi(\mathbf{x})$ (recall the definition of $\mathbf{T}_g$ given in the beginning of §4). For every $I \in \sigma$ we shall denote by $\mathbf{T}_g^I$ the interpretation of I in $\mathbf{T}_g$, that is, the relation $\{a \in H \mid I \in g(a)\}$.

▶ **Lemma 15.** *Let $b, b' \in H$ and let $X = \mathbf{T}_g^I$ for some $I \in \sigma$. Then:*
1. $b \sqcup b' \in X \Leftrightarrow b \in X$ *or* $b' \in X$
2. $b \sqcap b' \in X \Leftrightarrow b \in X$ *and* $b' \in X$

Proof. Follows directly from the definitions. ◀

▶ **Lemma 16.** *Let $\mathbf{A}$ be a τ-structure and let $h \in \mathrm{Sol}(\mathbf{A}, \varphi)$. Then $g^{-1} \circ h$ is well defined and belongs to* $\hom(\mathbf{A}, \mathbf{H})$.

Proof. Let $a \in A$ and let $\mathcal{Y}$ be a nonempty collection of subsets of H. $\mathcal{Y}$ is said to be *consistent* with $h(a)$ if for every $I \in \sigma$ the following holds:

$$I \in h(a) \Leftrightarrow Y \subseteq \mathbf{T}_g^I \text{ for some } Y \in \mathcal{Y}.$$

We claim that if there is a set $\mathcal{Y}$ consistent with $h(a)$ then $g^{-1}(h(a))$ is well defined and is equal to $\sqcup_{Y \in \mathcal{Y}} \sqcap Y$, where $\sqcap Y$ denotes the meet of all elements from Y. To see this, let $b = \sqcup_{Y \in \mathcal{Y}} \sqcap Y$. It follows from the definition of consistency and Lemma 15 that for every $I \in \sigma$

$$I \in h(a) \Leftrightarrow b \in \mathbf{T}_g^I,$$

which is equivalent to saying that $g(b) = h(a)$.

For every $a \in A$, let $\mathcal{Y}_a$ be the set $\{\mathbf{T}_g^I \mid I \in h(a)\}$. We have $\emptyset \neq g(\bot) \subseteq h(a)$, and hence $\mathcal{Y}_a$ is non-empty. We claim that $\mathcal{Y}_a$ is consistent with $h(a)$. Let $I \in \sigma$ and consider the two cases:

- $I \in h(a)$. In this case $\mathcal{Y}_a$ contains $\mathbf{T}_g^I$ and we are done.
- $I \notin h(a)$. Assume, towards a contradiction that $\mathbf{T}_g^J \subseteq \mathbf{T}_g^I$ for some $J \in h(a)$. This implies, by the definition of φ, that φ contains the clause $\neg J(x) \vee I(x)$, in contradiction with the fact that $I \notin h(a)$ and $J \in h(a)$.

Since for every $a \in A$, $\mathcal{Y}_a$ is a nonempty collection of sets consistent with $h(a)$, it follows that $g^{-1} \circ h$ is well defined. Now, let us prove that $g^{-1} \circ h \in \hom(\mathbf{A}, \mathbf{H})$.

Let $R \in \tau$ and let $(a_1, \ldots, a_k) \in \mathbf{A}^R$. For every $i = 1, \ldots, k$ and every $I \in h(a_i)$, let $Y_{i,I}$ be the set of all tuples $(b_1, \ldots, b_k) \in R^{\mathbf{H}}$ where $b_i \in \mathbf{T}_g^I$. The set $Y_{i,I}$ satisfies the following two claims:

Claim 1: $Y_{i,I} \neq \emptyset$. Otherwise, $\forall \mathbf{x}\ (\neg R(x_1, \ldots, x_k) \vee \neg I(x_i))$ holds in $(\mathbf{H}, \mathbf{T}_g)$, which implies that φ contains the clause $\neg R(x_1, \ldots, x_k) \vee \neg I(x_i)$, in contradiction with the fact that $(a_1, \ldots, a_k) \in \mathbf{A}^R$ and $I \in h(a_i)$.

Claim 2: For every $j = 1, \ldots, k$ and every $J \notin h(a_j)$, set $Y_{i,I}$ contains a tuple with $b_j \notin \mathbf{T}_g^J$. Otherwise, $\forall \mathbf{x}\ (\neg R(x_1, \ldots, x_k) \vee \neg I(x_i) \vee I(x_j))$ holds in $(\mathbf{H}, \mathbf{T}_g)$, which implies that φ contains the clause $\neg R(x_1, \ldots, x_k) \vee \neg I(x_i) \vee J(x_j)$, in contradiction with the fact that $(a_1, \ldots, a_k) \in \mathbf{A}^R$, $I \in h(a_i)$, and $J \notin h(a_j)$.

Let

$$\mathbf{c} = (c_1, \ldots, c_k) = \bigsqcup_{1 \le i \le k, I \in h(a_i)} \sqcap Y_{i,I}$$

Since $g(\perp) \subseteq h(a_i)$, Claim 1 above guarantees that the right term is not void. It follows from Claims 1 and 2 above that for every $j = 1, \ldots, k$ the set $\{\mathrm{proj}_j Y_{i,I} \mid 1 \le i \le k, I \in h(a_i)\}$ (where $\mathrm{proj}_j Y_{i,I}$ denotes the *projection* of $Y_{i,I}$ to its jth coordinate) is consistent with $h(a_j)$. This implies that $c_j = (g^{-1} \circ h)(a_j)$ for every $j = 1, \ldots, k$. Since $\mathbf{c}$ is obtained by iterative application of $\sqcup$ and $\sqcap$ to tuples in $R^{\mathbf{H}}$ we conclude that $\mathbf{c} \in R^{\mathbf{H}}$ and we are done. ◀

Proof of Theorem 6(1). Let $\mathbf{A}$ be any τ-structure. It follows from the definition of φ that $(\mathbf{H}, \mathbf{T}_g) \models \varphi$ or, equivalently, that $g \in \mathrm{Sol}(\mathbf{H}, \varphi)$. Then, by Lemma 11(1) $f \mapsto g \circ f$ defines a mapping from $\hom(\mathbf{A}, \mathbf{H})$ and $\mathrm{Sol}(\mathbf{A}, \varphi)$. This mapping is injective (because so is g) and exhaustive (by Lemma 16). ◀

6 Counting problems and linear Datalog

Datalog has a long and successful history as a tool in database theory and the study of the decision CSP (see [9] and the references therein). In this section we show how Datalog is also related to counting CSPs, and, more generally, to counting problems. As a byproduct of the proof of Theorem 6 we obtain a characterization of counting CSPs that can be represented through certain Datalog programs.

Datalog is a language of logic programs primarily developed in database theory. Let τ and σ be finite signatures. Symbols from τ are called *EDBs* (for Extensional DataBase symbols), and symbols from σ are called *IDBs* (for Intensional DataBase symbols). Every Datalog program is a collection of rules of the form

$$\psi_1 : -\psi_2, \ldots, \psi_m,$$

where $\psi_1, \ldots, \psi_m$ are atomic formulas using predicates from $\tau \cup \sigma$. The left side of the rule is called the *head* of the rule, and the predicate symbol occurring in it must be an IDB. The right hand side is called the *body* of the rule and might contain both IDBs and EDBs. A rule is called *linear* if its body contains at most one occurrence of an IDB. A Datalog program is said to be linear if all its rules are linear. The linear fragment of Datalog has been used for decision CSPs (see [2, 11, 12, 13]). In particular, the decision CSPs expressible through linear Datalog belong to the class NL. Moreover, it is conjectured that the converse is also true.

A Datalog program P is applied to a τ-structure $\mathbf{A}$ using the semantics of fixed points. Let $\mathbf{T}$ be an interpretation of the IDBs on the universe A of $\mathbf{A}$. Then $\mathbf{T}$ is said to be a *fixed point* of P on $\mathbf{A}$ if for every rule the following holds: for every interpretation of the variables of the rule that makes the body of the rule true given the intepretation of the IDBs (in $\mathbf{T}$) and of the EDBs (in $\mathbf{A}$), the head of the rule must hold in $\mathbf{T}$. Since the intersection of two fixed points is again a fixed point, there is always a least fixed point of a Datalog program. To express a decision CSP in terms of Datalog one should consider programs that contain a distinguished 'goal' IDB (usually null-ary). The corresponding CSP has no solution on input $\mathbf{A}$ if and only if the least fixed point of the program on $\mathbf{A}$ contains the goal IDB. The link to counting CSPs works in a different way. Here we consider not only the least fixed point, but all fixed points of a Datalog program. For a τ-structure $\mathbf{H}$ we say that a Datalog program P with EDBs from τ defines the problem $\#\mathrm{CSP}(\mathbf{H})$ if for any τ-structure $\mathbf{A}$ the number of homomorphisms from $\mathbf{A}$ to $\mathbf{H}$ equals the number of fixed points of P on $\mathbf{A}$.

For the purpose of this paper, however, we do not have to define the semantics of Datalog as above. Instead, we view linear Datalog as a fragment of monotone $\mathrm{RH}\Pi_1$ and fixed points as satisfying assignments of the corresponding formulas.

▶ **Lemma 17.** *Every linear Datalog program is equivalent to a monotone* $\mathrm{RH}\Pi_1$ *sentence in which each clause contains an unnegated predicate symbol from* σ *(equivalently, an IDB). Conversely, every monotone* $\mathrm{RH}\Pi_1$ *formula of this kind is equivalent to a Datalog program.*

Proof. The lemma easily follows from the observation that every rule $\psi_1 : -\psi_2, \dots, \psi_m$ of Datalog is equivalent to the clause $\psi_1 \vee \neg\psi_2 \vee \dots \vee \neg\psi_m$ of a monotone $\mathrm{RH}\Pi_1$ sentence. Note that the clause contains a positive occurrence (namely the symbol in ψ_1) of an IDB. Conversely, every clause like that with exactly one unnegated predicate from σ can be translated into a Datalog rule. ◀

Since Datalog is a proper subset of monotone $\mathrm{RH}\Pi_1$, it is likely to be less expressive.

▶ **Theorem 18.** *For every structure* $\mathbf{B}$ *the following holds:*

1. *If* $\mathbf{H}$ *has polymorphisms* $x \sqcap y$ *and* $x \sqcup y$ *for some distributive lattice* $(H; \sqcap, \sqcup)$ *and the nullary operation* $\top$ *(returning the greatest element of the lattice) then there is a linear Datalog program that defines* $\#\mathrm{CSP}(\mathbf{H})$
2. *Furthermore, if* $\mathbf{H}$ *contains equality then the converse also holds.*

Proof. (1) It has been show in §5 that, under the hypothesis of item (1), $\#\mathrm{CSP}(\mathbf{H})$ is definable by a monadic monotone $\mathrm{RH}\Pi_1$-sentence φ. Recall the definitions of φ, g, S, σ, and $\mathbf{T}_g$ from §5.

Assume now, additionally, that $\mathbf{H}$ is invariant under the nullary operation $\top$ returning the top element of the lattice. Let $\psi(\mathbf{x})$ be any clause in φ. We just need to show that some relation symbol from σ occurs unnegated in $\psi(\mathbf{x})$. We have $g(\top) = S$ by assumption (here and during the rest of the proof we shall slightly abuse the notation by using $\top$ to denote also the element in H that $\top$ returns). It follows that $\top \in \mathbf{T}_g^I$ for every $I \in \sigma$. Also, since $\mathbf{H}$ is invariant under $\top$ it follows that $(\top, \dots, \top) \in \mathbf{H}^R$ for every $R \in \tau$. By definition $(\mathbf{H}, \mathbf{T}_g) \models \forall \mathbf{x}\ \chi(\mathbf{x})$. Hence, if one instantiates all variables in $\mathbf{x}$ to $\top$ then one obtains an assigment that falsifies all negated atomic formulas in χ. Consequently, $\chi(\mathbf{x})$ must contain one unnegated atomic formula. Since $\chi(\mathbf{x})$ is monotone the predicate symbol of this unnegated atomic formula must be from σ.

(2) Recall the definition of J_φ from §4. It has been show in §4 that $\mathbf{J}_\varphi$ is invariant under set-theoretic union and intersection and that, under the hypothesis of item (2), $\mathbf{H}$ is isomorphic to $\mathbf{J}_\varphi$. Assume now that φ is a linear Datalog program. It is only necessary to show that, additionally, $\mathbf{J}_\varphi$ is invariant under the nullary operation returning σ (the top element in the lattice $(\mathrm{J}_\varphi, \cap, \cup)$). Let R be any predicate symbol in τ. Since every clause of φ has an occurrence of an unnegated predicate symbol from σ it follows that the mapping $\{1, \dots, k\} \mapsto \sigma$ belongs to $\mathrm{Sol}(\mathbf{J}_R, \varphi)$. Hence, by definition $(\sigma, \dots, \sigma) \in R^{\mathbf{J}_\varphi}$. ◀

References

1 Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *FOCS*, pages 595–603, 2009.

2 Libor Barto, Marcin Kozik, and Ross Willard. Near unanimity constraints have bounded pathwidth duality. In *LICS*, pages 125–134, 2012.

3 Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.

4 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. In *ICALP (1)*, pages 646–661, 2008.

5 Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24, 2011.

6 Andrei A. Bulatov and Víctor Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Information and Computation*, 205(5):651–678, 2007.

7 Andrei A. Bulatov and Martin Grohe. The complexity of partition functions. *Theor. Comput. Sci.*, 348(2-3):148–186, 2005.

8 Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

9 Andrei A. Bulatov, Andrei A. Krokhin, and Benoit Larose. Dualities for constraint satisfaction problems. In *Complexity of Constraints*, pages 93–124, 2008.

10 Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. In *STOC*, pages 909–920, 2012.

11 Catarina Carvalho, Víctor Dalmau, and Andrei A. Krokhin. CSP duality and trees of bounded pathwidth. *Theor. Comput. Sci.*, 411(34-36):3188–3208, 2010.

12 Víctor Dalmau. Linear datalog and bounded path duality of relational structures. *Logical Methods in Computer Science*, 1(1), 2005.

13 Víctor Dalmau and Andrei A. Krokhin. Majority constraints have bounded pathwidth duality. *Eur. J. Comb.*, 29(4):821–837, 2008.

14 Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2003.

15 Martin E. Dyer, Leslie Ann Goldberg, and Mark Jerrum. An approximation trichotomy for Boolean #CSP. *J. Comput. Syst. Sci.*, 76(3-4):267–277, 2010.

16 Martin E. Dyer, Leslie Ann Goldberg, and Mike Paterson. On counting homomorphisms to directed acyclic graphs. *J. ACM*, 54(6), 2007.

17 Martin E. Dyer and David Richerby. On the complexity of #CSP. In *STOC*, pages 725–734, 2010.

18 Martin E. Dyer and David Richerby. The #CSP dichotomy is decidable. In *STACS*, pages 261–272, 2011.

19 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

20 Tomás Feder and Moshe Y. Vardi. Homomorphism closed vs existential positive. In *LICS*, pages 311–320, 2003.

21 Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. In *STACS*, pages 493–504, 2009.

22 G. Grätzer. *General Lattice Theory.* Birkhäuser Verlag, Basel, 2003.

23 Pawel M. Idziak, Petar Markovic, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

24 Tomer Kotek and Johann Makowsky. Connection matrices and the definability of graph parameters. In *CSL*, pages 411–425, 2009.

25 Benoit Larose, Cynthia Loten, and Claude Tardif. A characterisation of first-order constraint satisfaction problems. *Logical Methods in Computer Science*, 3(4), 2007.

26 Benoit Larose and Pascal Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.*, 410(18):1629–1647, 2009.

27 Sanjeev Saluja, K. V. Subrahmanyam, and Madhukar N. Thakur. Descriptive complexity of #P functions. *J. Comput. Syst. Sci.*, 50(3):493–505, 1995.

28 L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.