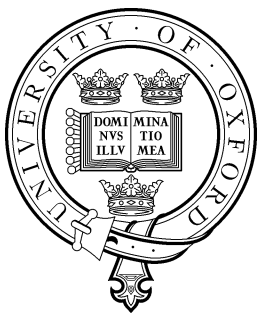
Programming Research Group

TOWARDS A DICHOTOMY THEOREM FOR THE COUNTING CONSTRAINT SATISFACTION PROBLEM

Andrei A. Bulatov
Computing Laboratory
University of Oxford, Oxford, UK
e-mail: andrei.Bulatov@comlab.ox.ac.uk
Víctor Dalmau
Department de Tecnologia
Universitat Pompeu Fabra, Barcelona, Spain
e-mail: victor.dalmau@tecn.upf.es

PRG-RR-03-13



Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

Abstract

The Counting Constraint Satisfaction Problem (#CSP) over a finite domain can be expressed as follows: given a first-order formula consisting of a conjunction of predicates, determine the number of satisfying assignments to the formula. It provides a general framework for numerous counting combinatorial problems including counting satisfying assignments to a propositional formula, graph homomorphisms, graph reliability and many others. #CSP can be parametrized by the set of allowed constraint predicates. In this paper we start a systematic study of subclasses of #CSP restricted in this way. The ultimate goal of this investigation is to distinguish those restricted subclasses of #CSP which are solvable in polynomial time from those which are not. We show that the complexity of any restricted #CSP class on a finite domain can be deduced from the properties of polymorphisms of the allowed constraints, similar to that for the decision constraint satisfaction problem. Then we prove that if a subclass of the #CSP is solvable in polynomial time, then constraints allowed by the class satisfy some very restrictive condition: it has to have a Mal'tsev polymorphism, that is a ternary operation $m(x, y, z)$ such that $m(x, y, y) = m(y, y, x) = x$. This condition uniformly explains all existing complexity results for particular cases of #CSP, and allows us to obtain new results and to conjecture a criterion distinguishing counting constraint satisfaction problems solvable in polynomial time. We also apply these results to obtain a dichotomy theorem for the complexity of #CSP with a 3-element domain and give a new simpler proofs of the dichotomy results for the problem of counting graph homomorphisms.

1 Introduction

In a counting combinatorial problem the objective is to find the number of feasible solutions to a certain search problem. Similar to its decision counterpart, the Counting Constraint Satisfaction Problem (#CSP) can be used to provide a generic framework for numerous combinatorial problems that arise frequently in a wide range of areas from logic, graph theory, and artificial intelligence [4, 12, 20, 25, 31, 38, 41, 46, 47, 50, 51], to statistical physics [3, 10, 36].

The prototypical counting problem, #SAT, i.e. the problem of counting the number of assignments that satisfy a CNF formula, constitutes an important particular case of #CSP. Since the pionnering papers of Valiant [50, 51] many efforts have been made to investigate the computational complexity of counting satisfying assignments to propositional formulas of various types [12, 38, 46, 47, 50, 51]. In particular, it has been found that #SAT is much more computationally demanding than its decision counterpart SAT, and is #P-complete even for Horn or monotone formulas, and even when the size of clauses and the number of occurrences of a variable in the formula are extremely limited. Finally, in [12], Creignou and Hermann have obtained a dichotomy theorem for #SAT, similar to that of Schaefer [48] for (decision) SAT.

The formalism of constraint networks, introduced by Montanari [40] provides a natural generalization of propositional formulas to domains with more than 2 elements. A constraint network is given by a collection of variables, a domain, and a family of constraints specifying some restrictions on the values of the domain that can be taken simultaneously by the variables. The problem of deciding whether there exists a solution to a constraint network, known as the constraint satisfaction problem (CSP) has recieved considerable attention in theoretical computer science and it also constitutes one of the major lines of research in artificial intelligence.

The class of counting constraint satisfaction problems is defined as the counting version on CSP, i.e. the problem of finding the number of solution to a constraint network. This problem can also be reformulated as (1) the problem of finding the number of models of a conjunctive formula (this is the form we use throughout the paper), as (2) the problem of counting the number of homomorphisms between two finite relational structures $\mathbf{A}$ and $\mathbf{B}$, and also as (3) the problem of computing the size (number of tuples) of the evaluation $Q(D)$ of a conjuntive query (without projection) Q on a database D; see [22, 35].

Further examples of combinatorial problems expressible in a natural way in #CSP terms include problems from propositional logic [12, 47], classical combinatorial problems such as #Clique, Graph Reliability, Antichain, Permanent [38, 46, 50, 51], enumerating of graph homomorphisms, and many others [4, 20, 25, 31].

A particular case of the counting graph homomorphisms problem, the class of $\#H$-Coloring problems, attracts a special attention. In a $\#H$-Coloring problem the goal is to count the number of homomorphisms from a graph G (the input) to a fixed graph H. Very recently, Dyer and Greenhill [20] have proved that, for every undirected graph H, its associated $\#H$-Coloring problem is either in FP or #P-complete and they have also provided a complete characterization of the tractable problems. This result has been

extended to the counting LIST $\#H$-COLORING problem [18], which allows additional restrictions on possible images of a node. Furthermore, some other variants of the $\#H$-COLORING problem for undirected graphs have been intensively studied during the last few years [16, 17].

Despite the large amount of results on its decision counterpart, numerous problems that can be stated in the form of the #CSP and massive work done for certain particular cases of #CSP, this problem, leaving apart the Boolean case (i.e. #SAT), has received little study. In this paper, we embark on a systematic study of the computational complexity of the subclasses of the #CSP parametrized by the set of allowed constraints. The ultimate goal of this study is to identify those restrictions which being imposed on the possible form of constraints lead to a problem solvable in polynomial time. To this end, we prove a somewhat surprising result claiming that the restricted classes of #CSP can be parametrized by sets of operations, polymorphisms, so that certain properties of the polymorphisms determine the complexity of a class. An analogous approach has proved to be very fruitful in the study of the decision constraint satisfaction problem [7, 8, 9, 32, 33, 34], and we expect that it also will be useful for the study of counting problems.

This algebraic approach allows us to identify a property common for all tractable restricted #CSP: we show that every #CSP-class having no polymorphism of a certain special type is #P-complete. Operations of this type are said to be Mal'tsev; these are ternary operations $m(x, y, z)$ such that $m(x, y, y) = m(y, y, x) = x$. Mal'tsev operations have been an object of intensive investigation in algebra for many years. We therefore get a new simple and powerful method of proving hardness results which has always been the most difficult part of any study of counting problems. Using this result we give a generic explanation to all existing complexity results on counting CSP and its particular cases including different types of $\#H$-COLORING [17, 16, 18, 20], and constraints considered in [14, 21, 22]. We also pose, for the first time, a plausible conjecture on a possible form of a criterion for #CSP, and prove this conjecture in the case of constraints on a 3-element domain. Another advantage of this approach is that it provides a new type of counting algorithms using deep algebraic results along with usual combinatorial methods.

2 The Counting Constraint Satisfaction Problem

2.1 Definitions and Examples

The counting constraint satisfaction problem can be formulated in several ways (see Section 1). We use the logical form of this problem. Throughout the paper we use the standard correspondence between predicates and relations: a relation consists of all tuples of values for which the corresponding predicate is true. We will use the same symbol for a predicate and its corresponding relation, since the meaning will always be clear from the context. By a *constraint language* on a set A we mean an arbitrary set of finitary predicates on A.

Definition 1 *Let A be a finite set, and Γ a constraint language on A. An* instance *of*

#CSP(Γ) *is a conjunctive formula* $F_1 \wedge \ldots \wedge F_n$ *where each* F_i *is an atomic formula interpreted as a predicate from* Γ. *The* objective *is to find the number of satisfying assignments (models) to the formula.*

For technical purposes, e.g. describing algorithms, we shall also use the definition of #CSP as a problem of counting solutions to a constraint network.

Definition 2 *The* counting constraint satisfaction problem (#CSP) *is the combinatorial function problem with*

INSTANCE: *a quadruple* $(V, A, \mathcal{C})$ *where* V *is a set of* variables, A *is a set of values [*domain*], *$\mathcal{C}$ *is a finite set of* constraints. *Each constraint* $C \in \mathcal{C}$ *is a pair* $\langle s, \varrho \rangle$, *where*

- $s = (v_1, \ldots, v_{m_C})$ *is a tuple of variables of length* m_C, *called the* constraint scope*;*
- ϱ *is an* m_C*-ary relation on* A, *called the* constraint relation.

OBJECTIVE: *compute the number of* solutions, *i.e. functions* φ, *from* V *to* A, *such that, for each constraint* $\langle s, \varrho \rangle \in \mathcal{C}$, *with* $s = (v_1, \ldots, v_m)$, *the tuple* $(\varphi(v_1), \ldots, \varphi(v_m))$ *belongs to* ϱ.

Example 1 (#SAT, [12, 13, 50, 51]) An instance of the #SAT problem is specified by giving a propositional logic formula in CNF, and asking how many assignments satisfy it. Therefore, #SAT is equivalent to #CSP(Γ) where Γ is the set of predicates defined on a two-element set and expressible by clauses. □

Example 2 (ANTICHAIN, [46]) In the problem ANTICHAIN we are given a finite poset $(P; \leq)$, and we aim to compute the number of antichains in P. This problem can be expressed in the #CSP-form as follows. Let $\varrho_{\prec}$ be the predicate of the natural order on $A = \{0, 1\}$. We assign a variable x_a to each element $a \in P$. Then the #CSP($\{\varrho_{\prec}\}$) instance $\Phi = \bigwedge_{a \leq b} \varrho_{\prec}(x_a, x_b)$ is equivalent to the original ANTICHAIN instance.

To show this, notice that every model φ to Φ satisfies the following condition: if $\varphi(x_a) = 1$ and $a \leq b$ then $\varphi(x_b) = 1$. This means that the set $F_\varphi = \{a \in P \mid \varphi(x_a) = 1\}$ is a *filter* of P, that is a set such that if $a \in F_\varphi$ and $a \leq b$ then $b \in F_\varphi$. Finally, notice that there is a one-to-one correspondence between antichains of the poset P and its filters. Indeed, for any antichain $H \subseteq P$, the set $\{q \in P \mid \text{there is } p \in H \text{ such that } p \leq q\}$ is a filter. Conversely, for a filter $F \subseteq P$, the set $\{q \in F \mid \text{if } p \in F \text{ and } p \leq q \text{ then } p = q\}$ of *minimal* elements from F forms an antichain. Thus the models of Φ one-to-one correspond to the filters of P, and consequently, to the antichains of P.

On the other hand, any #CSP($\{\varrho_{\prec}\}$) instance is reducible to an ANTICHAIN instance, though not so straightforwardly. Thus ANTICHAIN is equivalent to #CSP($\{\varrho_{\prec}\}$). □

Example 3 (#H-COLORING, [20, 28, 37]) Let H be a (directed) graph. In the H-COLORING problem we are asked, given a graph G, whether there is a homomorphism

from G to H. Correspondingly in its counting version, $\#H$-COLORING, the objective is to find the number of such homomorphisms.

Every (directed) graph $H = (V(H), E(H))$ corresponds to a binary predicate ϱ_H: $\varrho_H(a, b)$ holds iff (a, b) is an edge of H. Then every instance $G = (V(G); E(G))$ of the $\#H$-COLORING corresponds to the instance $\bigwedge_{(a,b)\in E(G)} \varrho_H(x_a, x_b)$ of #CSP where x_a is a variable assigned to a vertex $a \in V(G)$. Therefore $\#H$-COLORING is equivalent to $\#\mathrm{CSP}(\{\varrho_H\})$. □

Note: The notion of reduction used in the paper is one by oracles, as in [38, 46, 50, 51]. In fact, this notion of reduction was used also in [13], as its #P-completeness results relies upon the results of [38, 46, 50, 51]. Thus a problem L is #P-complete if $L \in \#\mathrm{P}$ and $\#P \subseteq \mathrm{FP}^L$. The general #CSP is known to be #P-complete, as follows from [51] and the examples above. However, some restricted problems have been shown to be computable in polynomial time.

Definition 3 *A constraint language* Γ *is called* #-tractable *if for any finite* $\Gamma' \subseteq \Gamma$ *the problem* $\#\mathrm{CSP}(\Gamma')$ *is solvable in polynomial time.*

A constraint language Γ *is called* #P-complete *if* $\#\mathrm{CSP}(\Gamma')$ *is #P-complete for a certain finite* $\Gamma' \subseteq \Gamma$.

Notice that this 'local' notion of tractability perfectly suits our aims, since, on the one hand, it deals with potentially infinite sets of predicates, that makes it possible to obtain general theoretical results; on the other hand, it is applicable to practical problems, because those mostly use finite sets of allowed predicates.

We are in a position to pose the main problem we tackle in this paper.

Problem 1 (classification problem) *Characterize #-tractable and #P-complete constraint languages on finite sets.*

This problem is completely solved for #SAT [12, 13].

Theorem 1 ([12, 13]) *A Boolean (that is on a 2-element set) constraint language* Γ *is #-tractable if and only if every relation from* Γ *is the solution space of a system of linear equations over the 2-element field. Otherwise,* Γ *is #P-complete.*

3 Invariance properties of #CSP

3.1 Relational clones

The main idea in tackling Problem 1 is to reduce the number of constraint languages to be considered by determining what predicates can be added to a #-tractable constraint language so that the obtained language remains #-tractable. This idea has been used in [12, 13] for #SAT, where new predicates are derived by the construction of *faithful implementation*. The following construction is equivalent to faithful implementation when $|A| = 2$, and, in fact, is prompted by the form of #CSP instances.

Proposition 1 *Let Γ be a constraint language on a finite set A, and $=_A$ be the predicate of equality on A. If ϱ is expressible by a conjunctive formula using predicates from Γ, then $\#\mathrm{CSP}(\Gamma \cup \{\varrho\})$ and $\#\mathrm{CSP}(\Gamma \cup \{=_A\})$ are polynomial time reducible to $\#\mathrm{CSP}(\Gamma)$.*

Proof: Take a problem instance $\Phi = F_1 \wedge \ldots \wedge F_s \in \#\mathrm{CSP}(\Gamma \cup \{\varrho\})$. Every predicate $F_i = \varrho(v_1, \ldots, v_m)$ can be represented in the form

$$\varrho(v_1, \ldots, v_m) = \varrho_1(v_{11}, \ldots, v_{1n_1}) \wedge \ldots \wedge \varrho_k(v_{k1}, \ldots, v_{kn_k})$$

where $\varrho_1, \ldots, \varrho_k \in \Gamma$ depend only on the predicate ϱ, and $v_{11}, \ldots, v_{1n_1}, v_{21}, \ldots, v_{kn_k} \in \{v_1, \ldots, v_m\}$. Replace F_i with the conjunction $\varrho_1(v_{11}, \ldots, v_{1n_1}) \wedge \ldots \wedge \varrho_k(v_{k1}, \ldots, v_{kn_k})$. We obtain a problem instance with the identical set of solutions. If we repeat the same replacement for every predicate in Φ equal to ϱ then we obtain an instance Φ' from $\#\mathrm{CSP}(\Gamma)$ with the same set of solutions, and hence, with the same number of solutions.

Now let $\Phi = F_1 \wedge \ldots \wedge F_s \in \#\mathrm{CSP}(\Gamma \cup \{=_A\})$. In order to get rid of the predicate of equality, $=_A$, we use the procedure suggested in [32]. For every F_i of the form $=_A (u, v)$ in Φ, we remove F_i from Φ and replace every occurence of v with u.

The obtained problem instance Φ^* belongs to $\#\mathrm{CSP}(\Gamma)$. Obviously, the reduction is fulfilled in polynomial time. Furthermore, although the set of solutions of Φ^* differs from the set of solutions to Φ (since some variables are removed), both have the same cardinality. □

Further constructions preserving #-tractability are much less easy and obvious.

Proposition 2 *Let Γ be a constraint language, and $\varrho(x_1, \ldots, x_n) \in \Gamma$. Then $\#\mathrm{CSP}(\Gamma \cup \{\exists x_n \varrho\})$ is polynomial time reducible to $\#\mathrm{CSP}(\Gamma)$.*

Proof: We use the interpolation technique first observed in [51]: given an $n \times n$ integer matrix with each entry bounded in magnitude by 2^m, the determinant and the inverse can be computed in time polynomial in n and m.

Let $\sigma = \exists x_n \varrho(x_1, \ldots, x_{n-1}, x_n)$, $\varrho \in \Gamma$. Basically, for each tuple $(a_1, \ldots, a_{n-1}) \in \sigma$ there are several b such that $\varrho(a_1, \ldots, a_{n-1}, b)$ holds. If $\sigma = \{\mathbf{a}_1, \ldots, \mathbf{a}_q\}$ where every $\mathbf{a}_j$ is an $n-1$-tuple over A, then let u_j be the number of extensions of $\mathbf{a}_j$.

Take a problem instance $\Phi = F_1 \wedge \ldots \wedge F_t \in \#\mathrm{CSP}(\Gamma \cup \{\sigma\})$. Without loss of generality we may assume that $F_1, \ldots, F_s$ are the predicates equal to σ. Let $l \geq 1$ and let $\Phi^{(l)}$ be the problem instance from $\#\mathrm{CSP}(\Gamma)$ defined as follows:

$$\Phi^{(l)} = \bigwedge_{i=1}^{s} (F_i^1 \wedge \ldots \wedge F_i^l) \wedge F_{s+1} \wedge \ldots \wedge F_t$$

where, for each $i = 1, \ldots, s$, if $F_i = \sigma(x_{i_1}, \ldots, x_{i_{n-1}})$ then $F_i^j = \varrho(x_{i_1}, \ldots, x_{i_{n-1}}, y_{ij})$, the variables y_{ij} are all different and do not occur in Φ.

For a solution φ to Φ, let $a_\varphi(j)$ denote the number of predicates $F_i(x_{i_1}, \ldots, x_{i_{n-1}})$, $i \leq s$, such that $(\varphi(x_{i_1}), \ldots, \varphi(x_{i_{n-1}})) = \mathbf{a}_j \in \sigma$. Clearly, $a_\varphi(1) + \ldots + a_\varphi(q) = s$. Furthermore, let $N(b_1, \ldots, b_q)$ denote the number of those solutions φ for which $a_\varphi(j) =$

b_j, $1 \leq j \leq q$. To solve the problem Φ we have to find the sum of all numbers of this form. Each $N(b_1, \ldots, b_q)$ corresponds to a partition of s in q nonnegative summands $b_1 + \ldots + b_q$. Hence the number p of Ns does not exceed $(s+1)^q$. Since q depends only on Γ and σ, this number is bounded by a polynomial in the size of Φ.

Every solution φ of Φ can be extended to a solution of $\Phi^{(l)}$. To this end, for each predicate $F_i(x_{i_1}, \ldots, x_{i_{n-1}})$, $i \leq s$, of Φ, let $(\varphi(x_{i_1}), \ldots, \varphi(x_{i_{n-1}})) = \mathbf{a}_j \in \sigma$. The tuple $\mathbf{a}_j$ can be extended to a tuple from ϱ in one of u_j ways. As F_i corresponds to l predicates in $\Phi^{(l)}$, the number of extensions for $\mathbf{a}_j$ is u_j^l. Since the same holds for each constraint we get

$$(u_1^l)^{a_\varphi(1)} \cdot (u_2^l)^{a_\varphi(2)} \cdot \ldots \cdot (u_q^l)^{a_\varphi(q)} = (u_1^{a_\varphi(1)} \cdot u_2^{a_\varphi(2)} \cdot \ldots \cdot u_q^{a_\varphi(q)})^l$$

extensions of φ to a solution of $\Phi^{(l)}$. The problem instance $\Phi^{(l)}$ belongs to $\#\mathrm{CSP}(\Gamma)$ and the number of its solutions is

$$N_l = \sum_{b_1+\ldots+b_q=s} N(b_1, \ldots, b_q)(u_1^{b_1} \cdot \ldots \cdot u_q^{b_q})^l.$$

To avoid cumbersome notation we assume that all the numbers $u_1^{b_1} \cdot \ldots \cdot u_q^{b_q}$ are different. We can do this without loss of generality because if $u_1^{b_1} \cdot \ldots \cdot u_q^{b_q} = u_1^{c_1} \cdot \ldots \cdot u_q^{c_q}$ then we may just replace $N(b_1, \ldots, b_q)$ and $N(c_1, \ldots, c_q)$ with their sum, and so shorten the sum above. The matrix of the linear system

$$\begin{aligned}
\sum_{b_1+\ldots+b_q=s} N(b_1, \ldots, b_q) u_1^{b_1} \cdot \ldots \cdot u_q^{b_q} &= N_1 \\
\sum_{b_1+\ldots+b_q=s} N(b_1, \ldots, b_q)(u_1^{b_1} \cdot \ldots \cdot u_q^{b_q})^2 &= N_2 \\
&\vdots \\
\sum_{b_1+\ldots+b_q=s} N(b_1, \ldots, b_q)(u_1^{b_1} \cdot \ldots \cdot u_q^{b_q})^p &= N_p
\end{aligned}$$

is Vandermonde, and therefore the system can be solved in polynomial time. □

Below are some examples of predicates that can be expressed by conjunction and existential quantification.

Example 4 (1) *Cartesian product* of relations (predicates). Let ϱ, σ be n- and m-ary relations respectively. Then

$$(\varrho \times \sigma)(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}) = \varrho(x_1, \ldots, x_n) \wedge \sigma(x_{n+1}, \ldots, x_{n+m}).$$

(2) The standard *composition* of binary relations (predicates). Let ϱ, σ be binary relations. Then

$$(\varrho \circ \sigma)(x, y) = \exists z \; \varrho(x, z) \wedge \sigma(z, y).$$

(3) Let ϱ be an n-ary relation (predicate), and $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$. The relation $\sigma = \mathrm{pr}_I \varrho$ called the *projection of ϱ onto I* is defined as follows

$$\sigma(x_{i_1}, \ldots, x_{i_k}) = \exists x_i (i \notin I)\ \varrho(x_1, \ldots, x_n).$$

Constraint languages containing the predicate of equality and closed with respect to taking conjuction and existential quantification have been intensively studied (see e.g. [42, 43, 44]) and have provided strong assistance in the study of the decision constraint satisfaction problem [8, 9, 32, 34].

Definition 4 *A constraint language Γ is said to be a* relational clone *if it contains the predicate of equality, and every predicate representable by a formula using predicates from Γ, conjunction, and existential quantification, belongs to Γ.*

For a constraint language Δ, the relational clone, consisting of all predicates expressible using predicates from Δ, conjunction, and existential quantification is denoted by $\langle\Delta\rangle$.

Propositions 1,2 imply the first main result of the paper.

Theorem 2 *Let Γ_1, Γ_2 be constraint languages on a finite set A such that Γ_2 is finite and $\Gamma_2 \subseteq \langle\Gamma_1\rangle$. Then $\#\mathrm{CSP}(\Gamma_2)$ is polynomial time reducible to $\#\mathrm{CSP}(\Gamma_1)$.*

Corollary 1 *A constraint language Γ is #-tractable (#P-complete) if and only if so is $\langle\Gamma\rangle$.*

3.2 Polymorphisms

The results of the previous section show that the class of constraint languages to be studied can be considerably reduced. To reduce it even further we use the invariance property of relations [32, 34, 44]. Recall that A^n denotes the set of all n-tuples of elements from A. Any operation on a set A can be extended in a standard way to an operation on tuples over A, as follows. For any (m-ary) operation f, and any collection of tuples $\mathbf{a}_1, \ldots, \mathbf{a}_m \in A^n$, where $\mathbf{a}_i = (\mathbf{a}_i[1], \ldots, \mathbf{a}_i[n])$ $(i = 1 \ldots m)$, define $f(\mathbf{a}_1, \ldots, \mathbf{a}_m)$ to be $(f(\mathbf{a}_1[1], \ldots, \mathbf{a}_m[1]), \ldots, f(\mathbf{a}_1[n], \ldots, \mathbf{a}_m[n]))$. Then f *preserves* an n-ary relation ϱ (or ϱ is *invariant* under f, or f is a *polymorphism of* ϱ) if for any $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \varrho$ the tuple $f(\mathbf{a}_1, \ldots, \mathbf{a}_m)$ belongs to ϱ. For a given set of operations, C, the set of all relations invariant under every operation from C is denoted by $\mathsf{Inv}\ C$. Conversely, for a constraint language, Γ, the set of all operations preserving every relation from Γ is denoted by $\mathsf{Pol}\ \Gamma$.

Example 5 *Let ϱ be the solution space of a system of linear equations over a field F. Then the operation $m(x, y, z) = x - y + z$ is a polymorphism of ϱ. Indeed, let $A \cdot \mathbf{x} = \mathbf{b}$ be the system defining ϱ, and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \varrho$. Then*

$$A \cdot m(\mathbf{x}, \mathbf{y}, \mathbf{z}) = A \cdot (\mathbf{x} - \mathbf{y} + \mathbf{z}) = A \cdot \mathbf{x} - A \cdot \mathbf{y} + A \cdot \mathbf{z} = \mathbf{b} - \mathbf{b} + \mathbf{b} = \mathbf{b}.$$

In fact, the converse can also be shown: if ϱ is invariant under m then it is the solution space of a certain system of linear equations. □

All sets of the form $\mathsf{Inv}\ C$ are relational clones, and every relational clone can be represented in this form [44, 49]. For example, for a relational clone Γ, we have $\Gamma = \mathsf{Inv}\ \mathsf{Pol}\ \Gamma$.

Proposition 3 ([44, 49]) *If Γ is a constraint language then $\langle\Gamma\rangle = \mathsf{Inv}\ \mathsf{Pol}\ \Gamma$. Therefore, $\mathsf{Pol}\ \Gamma = \mathsf{Pol}\ \langle\Gamma\rangle$.*

As a consequence of Theorem 2 and Proposition 3 we deduce the following important result that constitutes the basis of the algebraic approach to the counting CSP.

Theorem 3 *Let Γ_1, Γ_2 be constraint languages on A such that Γ_2 is finite and $\mathsf{Pol}\ \Gamma_1 \subseteq \mathsf{Pol}\ \Gamma_2$. Then $\#\mathrm{CSP}(\Gamma_2)$ is polynomial time reducible to $\#\mathrm{CSP}(\Gamma_1)$. Therefore, if $\#\mathrm{CSP}(\Gamma_1)$ is tractable then so is $\#\mathrm{CSP}(\Gamma_2)$.*

Consequently, all the information about the complexity of $\#\mathrm{CSP}(\Gamma)$ can be extracted from the family of polymorphisms of Γ. Sets of polymorphisms often provide a more convenient and concise way of describing a class of constraint satisfaction problems. In particular, the dichotomy result for Boolean constraint languages can be reformulated as follows.

Theorem 4 ([13]) *A Boolean constraint language Γ is #-tractable if and only if every relation from Γ is invariant with respect to the operation $x-y+z$ where $+, -$ are addition and subtraction modulo 2. Otherwise Γ is #P-complete.*

The operation $x - y + z$ is one of the simplest examples of so called Mal'tsev operations: a ternary operation $m(x, y, z)$ on a set A is said to be *Mal'tsev* if it satisfies the condition $m(x, y, y) = m(y, y, x) = x$ for any $x, y \in A$. Another well known example of a Mal'tsev operation is the operation $xy^{-1}z$ of a group.

The following theorem, our second main result, shows that Mal'tsev operations play, possibly, the crucial role in the study of #CSP.

Theorem 5 *If Γ is a constraint language which is invariant under no Mal'tsev operation then Γ is #P-complete.*

In the next section we develop the algebraic machinary required to prove Theorem 5. Then, in Section 5, we apply Theorem 5 to some particular cases of #CSP, obtaining new algorithms, reproving and generalizing existing results. We show that in all cases considered the existence of a Mal'tsev polymorphism is sufficient for the #-tractability of a constraint language. Therefore the following conjecture seems to be plausible.

Conjecture 1 (Dichotomy Conjecture) *A constraint language Γ is #-tractable if and only if it is invariant under a Mal'tsev operation. Otherwise Γ is #P-complete.*

4 Algebraic Structure of #CSP

In this section we further develop algebraic machinery for the counting constraint satisfaction problem. Our aims are, first, to make it possible to use the most advanced and powerful results from universal algebra in the study of #CSP, and, second, to develop techniques sufficient to prove Theorem 5.

4.1 Algebraic constructions and #CSP

In this subsection we give basic algebraic definitions. We also introduce the notion of "#-tractable algebra" and show how it relates to the complexity of problem classes of the form #CSP(Γ). In our algebraic definitions we follow [11, 39]. For algebraic notions and results concerning the decision CSP the reader is referred to [9, 7].

A (*universal*) *algebra* is an ordered pair $\mathbb{A} = (A, F)$ where A is a non-empty set and F is a family of finitary operations on A. The set A is called the *universe* of $\mathbb{A}$, the operations from F are called *basic*. An algebra with a finite universe is referred to as a *finite algebra*.

Any constraint language Γ on a set A can be converted into an algebra $\mathbb{A}_\Gamma = (A; \mathsf{Pol}\ \Gamma)$, and vice versa, for any algebra $(A; F)$, there is a corresponding constraint language $\mathsf{Inv}\ F$. By Theorem 3, if $\mathsf{Pol}\ \Gamma_1 = \mathsf{Pol}\ \Gamma_2$ or, equivalently, $\mathbb{A}_{\Gamma_1} = \mathbb{A}_{\Gamma_2}$, then Γ_1, Γ_2 are #-tractable or #P-complete simultaneously. Therefore all the problem classes can be parametrized by finite algebras so that classes with the same parameter have the same complexity. We make the following definition.

Definition 5 *An algebra $\mathbb{A} = (A; F)$ is said to be* #-tractable (#P-complete) *if the constraint language $\mathsf{Inv}\ F$ is #-tractable (#P-complete).*

By #CSP($\mathbb{A}$) *we denote the problem class* #CSP($\mathsf{Inv}\ F$).

Making use of this definition we reformulate Problem 1.

Problem 2 (classification problem) *Characterise #-tractable and #P-complete finite algebras.*

Theorem 4 provides the first step towards a solution of this problem, because it yields a complete classification of two-element algebras with respect to #-tractability. An operation f is said to be a *term* operation of an algebra $\mathbb{A} = (A; F)$ if $f \in \mathsf{Pol}\ \mathsf{Inv}\ F$. It is straightforward that, if a relation ϱ is invariant under F then it is also invariant under every term operation of $\mathbb{A}$.

Theorem 6 ([12, 13]) *A two-element algebra $\mathbb{A} = (\{0, 1\}, F)$ is #-tractable if and only if $x - y + z$ is a term operation of $\mathbb{A}$. Otherwise $\mathbb{A}$ is #P-complete.*

The main idea of the algebraic approach is to use some properties of an algebra in order to determine the complexity of the associated #CSP. To identify these properties, some connections between the complexity of an algebra and standard algebraic constructions will be very helpful.

Definition 6 *Let $\mathbb{A} = (A, F)$ be an algebra. The* n-th direct power *of $\mathbb{A}$ is the algebra $\mathbb{A}^n = (A^n, F)$ where we treat each (m-ary) operation $f \in F$ as acting on A^n.*

Definition 7 *Let $\mathbb{A} = (A, F)$ be an algebra, and B a subset of A such that, for any (n-ary) $f \in F$, and for any $b_1, \ldots, b_n \in B$, we have $f(b_1, \ldots, b_n) \in B$. Then the algebra $\mathbb{B} = (B, F|_B)$, where $F|_B$ consists of restrictions of operations $f \in F$ to B, is called a*

subalgebra *of* $\mathbb{A}$.

Definition 8 *Let* $\mathbb{A}_1 = (A_1, F_1)$ *and* $\mathbb{A}_2 = (A_2, F_2)$ *such that* $F_1 = \{f_i^1 \mid i \in I\}$, $F_2 = \{f_i^2 \mid i \in I\}$, *and* f_i^1, f_i^2 *are of the same arity,* $i \in I$. *A mapping* $\varphi : A_1 \to A_2$ *is called a* homomorphism *from* $\mathbb{A}_1$ *to* $\mathbb{A}_2$ *if* $\varphi f_i^1(a_1, \ldots, a_{n_i}) = f_i^2(\varphi(a_1), \ldots, \varphi(a_{n_i}))$ *holds for all* $i \in I$ *and all* $a_1, \ldots, a_{n_i} \in A_1$. *If the mapping* φ *is onto then* $\mathbb{A}_2$ *is said to be a* homomorphic image *of* $\mathbb{A}_1$.

Definition 9 *Let* $\mathbb{A} = (A, F)$ *be an algebra. An equivalence relation* $\alpha \in \mathsf{Inv}\ F$ *is said to be a* congruence *of* $\mathbb{A}$.

For example, if $\mathbb{A}$ is a group then α is a congruence if and only if the classes of α are the cosets of a normal subgroup. A congruence allows one to break up the algebra into blocks, costituting the partition corresponding to the congruence as an equivalence relation, that are compartible with operations of the algebra. So, when varying values of arguments of an operation inside congruence blocks, the result of the operation also stays inside a congruence block. Therefore, we may collapse each congruence block into a single element, and correctly extrapolate the operations of the algebra (basic and term) onto this new *factor-set*. Then the factor-set together with the extrapolated operations form the *factor-algebra*. For an algebra $\mathbb{A}$ and a congruence θ, the factor-algebra is denoted by $\mathbb{A}/_\theta$. In a sense the factor-algebra captures the rough structure of the algebra.

A property of algebras such that if an algebra enjoys the property then any subalgebra, homomorphic image, and direct power also enjoys it, is said to be *hereditary*. Universal algebra mostly deals with hereditary properties [30, 39]. Therefore, the next theorem allows us to apply the most powerful methods of modern algebra to the study of the complexity of the counting constraint satisfaction problem.

Theorem 7 *Let* $\mathbb{A} = (A; F)$ *be a finite algebra. Then*

(i) if $\mathbb{A}$ *is #-tractable then so is every subalgebra, homomorphic image, and direct power of* $\mathbb{A}$.

(ii) if $\mathbb{A}$ *has an #P-complete subalgebra, homomorphic image, or direct power, then* $\mathbb{A}$ *is #P-complete itself.*

Proof: We prove that if $\mathbb{B} = (B; F')$ is a subalgebra, homomorphic image or direct power of $\mathbb{A}$, then for aby finite $\Gamma \subseteq \mathsf{Inv}\ F'$, there is a finite $\Delta \subseteq \mathsf{Inv}\ F$ such that $\#\mathrm{CSP}(\Gamma)$ is polynomial time reducible to $\#\mathrm{CSP}(\Delta)$. The result then follows straightforwardly.

(1) Let $\mathbb{B} = (B; F\big|_B)$ be a subalgebra of $\mathbb{A}$. This means that any relation $\varrho \in \mathsf{Inv}\ F\big|_B$ is also invariant with respect to F. Therefore $\mathsf{Inv}\ F\big|_B \subseteq \mathsf{Inv}\ F$, and $\#\mathrm{CSP}(\mathbb{B})$ is trivially reducible to $\#\mathrm{CSP}(\mathbb{A})$.

(2) Let $\mathbb{B} = (A^k; F') = \mathbb{A}^k$ be a direct power of $\mathbb{A}$. Then any (n-ary) $\varrho \in \mathsf{Inv}\ F'$ can be represented in the form

$$\varrho' = \{(a_{11}, \ldots, a_{1k}, a_{21}, \ldots, a_{nk}) \mid (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in \varrho, \mathbf{a}_i = (a_{i1}, \ldots, a_{ik})\},$$

and as is well known $\varrho' \in \mathsf{Inv}\ F$.

Take a problem instance $\mathcal{P} = (V; A^k; \mathcal{C})$ from $\#\mathrm{CSP}(\mathbb{B})$, and transform it to $\mathcal{P}' = (V'; A; \mathcal{C}')$ where

- $V' = \{v^1, \dots, v^k \mid v \in V\}$;
- every $\langle s, \varrho\rangle \in \mathcal{C}$ where $s = (v_1, \dots, v_l)$ is replaced with $\langle s', \varrho'\rangle$, $s' = (v_1^1, \dots, v_1^k, v_2^1, \dots, v_l^k)$ and ϱ' is constructed as above.

The instance $\mathcal{P}'$ has the same number of solutions as $\mathcal{P}$ has, and $\mathcal{P}' \in \#\mathrm{CSP}(\mathbb{A})$.

(3) Let $\mathbb{B} = (B; F')$ be a homomorphic image of $\mathbb{A} = (A; F)$ under a homomorphism φ. Take a finite constraint language $\Gamma \subseteq \mathsf{Inv}\ F'$ such that $=_B \in \Gamma$ and denote by $\Theta = \varphi^{-1}(\Gamma)$ the set consisting of full preimages of relations from Γ. In particular, $\varphi^{-1}(=_B)$ is a congruence of $\mathbb{A}$, the kernel of φ. It is well known that $\Theta \subseteq \mathsf{Inv}\ F$. We solve the problem $\#\mathrm{CSP}(\Gamma)$ with the oracle $\#\mathrm{CSP}(\Theta)$ in polynomial time.

Let $\mathcal{P} = (V; B; \mathcal{C})$ be a problem instance from $\#\mathrm{CSP}(\Gamma)$; and, for any $1 \le k \le \binom{m-1}{|V|}$ where $m = |B|$ is the number of classes of $\varphi^{-1}(=_B)$, let $\mathcal{P}^{(k)} = (V^{(k)}; A; \mathcal{C}^{(k)}) \in \#\mathrm{CSP}(\Theta)$ be defined as follows

- $V^{(k)} = \{v^1, \dots, v^k \mid v \in V\}$ is a disjoint union of k copies of V;
- for any $\langle s, \varrho\rangle \in \mathcal{C}$, $s = (v_1, \dots, v_n)$, we include in $\mathcal{C}^{(k)}$ the constraints $\langle s^1, \varphi^{-1}(\varrho)\rangle, \dots, \langle s^k, \varphi^{-1}(\varrho)\rangle$ where $s^i = (v_1^i, \dots, v_n^i)$;
- for every variable $v \in V$ and every pair $1 \le i, j \le k$ we include in $\mathcal{C}^{(k)}$ the constraint $\langle (v^i, v^j), \varphi^{-1}(=_B)\rangle$.

Notice that for a solution ψ to $\mathcal{P}^{(k)}$ and $1 \le i \le k$ the mapping $\psi^i \colon V \to B$, $\psi^i(v) = \varphi(\psi(v^i))$, is a solution to $\mathcal{P}$, and moreover, $\psi^i = \psi^j$ for any $1 \le i, j \le k$. Let the sizes of the classes of $\varphi^{-1}(=_B)$ be $u_1, \dots, u_m$. Then the number of solutions to $\mathcal{P}^{(k)}$ is

$$N_k = \sum_{t_1+t_2+\dots+t_m=|V|} N(t_1, \dots, t_m)(u_1^k)^{t_1}(u_2^k)^{t_2}\dots(u_m^k)^{t_m}$$

where $N(t_1, \dots, t_m)$ denotes the number of solutions to $\mathcal{P}$ that take t_i values with the size of the preimage u_i.

Since each $\mathcal{P}^{(k)}$ belongs to $\#\mathrm{CSP}(\Theta)$, the numbers N_k can be found in polynomial time. We may assume all the numbers $u_1^{t_1}u_2^{t_2}\dots u_m^{t_m}$ to be different for different tuples $(t_1, \dots, t_m)$. It will not be loss of generality because if $u_1^{t_1}\dots u_m^{t_m} = u_1^{t'_1}\dots u_m^{t'_m}$ then we may just replace $N(t_1, \dots, t_m)$ and $N(t'_1, \dots, t'_m)$ with their sum, and so shorten the sums above. We therefore have a system of linear equations whose matrix is Vandermonde, and hence the system can be solved in polynomial time. Finally, the number of solutions to $\mathcal{P}$ is the sum of all numbers of the form $N(t_1, \dots, t_m)$. □

An operation f is said to be *idempotent* if the identity $f(x, \dots, x) = x$ holds. Algebras whose basic (and therefore term) operations are idempotent posess many useful properties

that will assist in our investigation. The *full idempotent reduct* of an algebra $\mathbb{A} = (A; F)$ is the algebra $\mathsf{Id}(\mathbb{A}) = (A; F_{\mathrm{id}})$ where F_{id} consists of all idempotent term operations of $\mathbb{A}$. There is another way to characterize F_{id}: $F_{\mathrm{id}} = \mathsf{Pol}\,(\mathsf{Inv}\,F \cup \{\{(a)\} \mid a \in A\})$ where $\{(a)\}$ means a unary relation containing only one tuple.

Theorem 8 *A finite algebra $\mathbb{A}$ is #-tractable [#P-complete] if and only if so is $\mathsf{Id}(\mathbb{A})$.*

Let $\mathbb{A} = (A; F)$ be a finite algebra. We show that for every finite constraint language Γ on A, if $\#\mathrm{CSP}(\Gamma)$ is tractable, then so is $\#\mathrm{CSP}(\Gamma \cup \{\{(a)\} \mid a \in A\})$. Consequently, if $\mathsf{Inv}\ F$ is #-tractable then so is $\mathsf{Inv}\ F \cup \{\{(a)\} \mid a \in A\})$. Finally, since $F_{\mathrm{id}} = \mathsf{Pol}\,(\mathsf{Inv}\ F \cup \{\{(a)\} \mid a \in A\})$, the result follows.

Let $A = \{a_1, \ldots, a_n\}$ (assuming $a_1, \ldots, a_n$ are different) be a finite set and let Γ be a finite constraint language on A. It is known [44, 49] that the n-ary relation $\varrho_1 = \{(h(a_1), \ldots, h(a_n)) \mid h : A \to A, h \in \mathsf{Pol}\ \Gamma\}$ is in $\langle\Gamma\rangle$.

Let $\mathcal{P} = (V, A, \mathcal{C})$ be a problem instance in $\#\mathrm{CSP}(\Gamma \cup \{\{(a)\} \mid a \in V\})$ and let $\mathcal{P}'$ be the problem instance $(V', A, \mathcal{C}')$ in $\#\mathrm{CSP}(\Gamma \cup \{=_A, \varrho_1\}))$, where $V' = V \cup \{v_a \mid a \in A\}$, ($v_a$ for $a \in A$ are variables not in V), and $\mathcal{C}'$ contains every constraint $C = \langle s, \varrho\rangle$ in $\mathcal{C}$, such that $\varrho \in \Gamma$. Furthermore, for every constraint $\langle v, \{a\}\rangle$ in $\mathcal{C}$, $\mathcal{C}'$ contains the constraint $\langle (v, v_a), =_A\rangle$, and also $\mathcal{C}'$ contains the constraint $\langle (v_{a_1}, \ldots, v_{a_n}), \varrho_1\rangle$.

The number of solutions to $\mathcal{P}$ equals the number of solutions φ to $\mathcal{P}'$ such that $\varphi(v_a) = a$ for all $a \in A$. Let $\mathcal{N}$ be the set of all such solutions. The cardinality of $\mathcal{N}$ can be computed in two stages. For every subset B of A we define $\mathcal{P}'_B$ as the problem $(V', A, \mathcal{C}' \cup \{\langle (v_a, v_{a'}), =_A\rangle \mid a, a' \in B\})$. Notice that any function φ is a solution of $\mathcal{P}'_B$ if φ is a solution of $\mathcal{P}'$ and furthermore for every $a, a' \in B$, $\varphi(v_a) = \varphi(v_{a'})$. Let us call N_B the number of satisfying assignments to $\mathcal{P}'_B$. The number N_B can be computed in polynomial time since Γ is #-tractable and $\{=_A, \varrho_1\} \subseteq \langle\Gamma\rangle$.

In the first stage we will compute how many solutions φ of $\mathcal{P}'$ assign v_a, $a \in A$, pairwise different values. Let us denote by $\mathcal{M}$ the set of all such solutions. Clearly, the cardinality of $\mathcal{M}$ can be obtained by Inclusion-Exclusion Principle:

$$|\mathcal{M}| = N_\emptyset - N_{\{a_1,a_2\}} - N_{\{a_1,a_3\}} - \ldots - N_{\{a_{n-1},a_n\}} + N_{\{a_1,a_2,a_3\}} + \ldots$$

In the second stage we express the cardinality of $\mathcal{N}$ via the cardinality of $\mathcal{M}$. Let G be the set of all permutations in $\mathsf{Pol}\ \Gamma$. It is well known [44, 49] that, since A is finite, G constitutes a permutation group.

We show that $\mathcal{M} = \{g\varphi : g \in G, \varphi \in \mathcal{N}\}$. For every solution φ in $\mathcal{N}$ and every $g \in G$, $g\varphi$ is also a solution of $\mathcal{P}'$ [34] and, since g is one-to-one, $g\varphi$ is in $\mathcal{M}$. Conversely, for every $\psi \in \mathcal{M}$, there exists some $g \in G$ such that $g(a) = \psi(v_a), a \in A$. Notice that $g^{-1} \in G$ implies $\varphi = g^{-1}\psi \in \mathcal{N}$, and therefore $\psi = g\varphi$.

Finally, for every $\varphi, \varphi' \in \mathcal{N}$ and every $g, g' \in G$, if $\varphi \neq \varphi'$ or $g \neq g'$ then $g\varphi \neq g'\varphi'$. In consequence, $|\mathcal{M}| = |G| \cdot |\mathcal{N}|$. □

4.2 Hard cases of #CSP

As we shall see, binary reflexive, but not symmetric relations play the central role in solving Problems 1,2.

Theorem 9 *If σ is a binary reflexive but not symmetric relation on a finite set then* #CSP($\{\sigma\}$) *is #P-complete.*

Proof: Let σ be reflexive but not symmetric binary relation on a set A. This means that there are $a, b \in A$ such that $(a, b) \in \sigma$ and $(b, a) \notin \sigma$. First of all we show that σ can be supposed to be of a very special form. Since, by Theorem 2 and Theorem 8, $\{\sigma\}$ is #-tractable [#P-complete] if and only if the relational clone R generated by σ and the unary relations $\{(a)\}$, $a \in A$, is #-tractable [#P-complete], it is enough to find an #P-complete relation in R.

Claim 1. σ can be supposed satisfying the condition: for any $c \in A$, $(a, c), (c, b) \in \sigma$, see Fig. 1(a).

Let us denote by A' the set of all $c \in A$ such that $(a, c), (c, b) \in \sigma$. Notice that $a, b \in A'$. The unary relation A' belongs to R, as the following formula shows

$$A'(x) = \exists y, z\ \sigma(y, x) \wedge P_a(y) \wedge \sigma(x, z) \wedge P_b(z)$$

where P_d is the predicate corresponding to the relation $\{(d)\}$, $d \in A$. If $A' \neq A$ then #CSP($\sigma \cap A'^2$) is reducible to #CSP(σ), $\sigma \cap A'^2$ satisfies the required conditions on the set A', and σ can be replaced with $\sigma \cap A'^2$.

Claim 2. σ can be supposed satisfying the condition: if $(c, a) \in \sigma$ then $(c, d) \in \sigma$ for any $d \in A$, see Fig 1(b).

If $(c, a) \in \sigma$ then set $A' = \{d \in A \mid (c, d) \in \sigma\}$. The relation A' belongs to R,

$$A'(x) = \exists y\ \sigma(y, x) \wedge P_c(y),$$

and $a, b \in A'$. Therefore, σ can be replaced with the relation $\sigma \cap A'^2$.

Claim 3. σ can be supposed satisfying the condition: if $(c, a) \notin \sigma$ then $(d, c) \in \sigma$ for any $d \in A$.

If $(c, a) \notin \sigma$ then set $A' = \{d \in A \mid (d, c) \in \sigma\}$. The relation A' belongs to R,

$$P_{A'}(x) = \exists y\ P_\sigma(x, y) \wedge P_c(y),$$

and $a, c \in A'$. The relation $\sigma' = \sigma \cap A'^2$ on the set A' satisfies the conditions of Theorem 9, because $(a, c) \in \sigma'$ and $(c, a) \notin \sigma'$.

We need more notation. Set $D = \{c \in A \mid (c, a) \in \sigma\}$, $B = A - D$, and $D_d = \{c \in D \mid (d, c) \in \sigma\}$ for $d \in B$. Then $a \in D$, $b \in B$, and $a \in D_c$ for no $c \in B$.

Claim 4. σ can be supposed satisfying the condition: for any $d \in B$, $D_d = D_b$, see Fig 1(c).

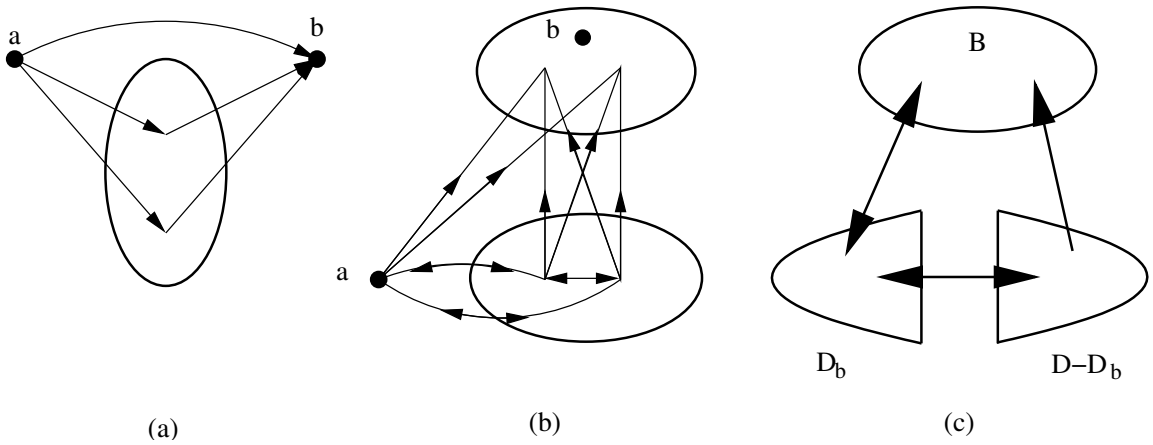


Figure 1: A scheme of the relation obtained in (a) Claim 1, (b) Claim 2, (c) Claim 4

Choose an element $d \in B$ such that D_d is maximal, and set $B' = \{c \in B \mid D_c = D_d\}$. The relation $B' \cup D$ belongs to R,

$$(B' \cup D)(x) = \exists y_1, \ldots, y_k \bigwedge_{i=1}^{k} (\sigma(x, y_i) \wedge P_{a_i}(y_i))$$

where $D_d = \{a_1, \ldots, a_k\}$. The relation $\sigma' = \sigma \cap (B' \cup D)^2$ on the set $B' \cup D$ satisfies the conditions of Theorem 9, because $a, d \in B' \cup D$, and $(a, d) \in \sigma'$, $(d, a) \notin \sigma'$.

We reduce Antichain to CSP(σ). Let $(P; \leq)$ be a problem instance of Antichain. As has been observed in Example 2, there is a one-to-one correspondence between antichains of the poset P and its filters.

Consider the problem instance $\mathcal{P}^{(l)} = (V^{(l)}; A; \mathcal{C}^{(l)})$, for a natural number l, defined as follows, see Fig. 2.

- $V^{(l)} = P_1 \cup \ldots \cup P_l \cup \{v_a, v_b\}$ where $P_1, \ldots, P_l$ are disjoint copies of P, and if $P = \{p_1, \ldots, p_n\}$ then $P_i = \{p_1^i, \ldots, p_n^i\}$.
- A constraint $C = \langle s, \varrho \rangle$ belongs to $\mathcal{C}^{(l)}$ if and only if C is one of the following forms
  - $s = (v_a)$, $\varrho = \{(a)\}$ or $s = (v_b)$, $\varrho = \{(b)\}$;
  - $s = (v_a, p_j^i)$, $\varrho = \sigma$ or $s = (p_j^i, v_b)$, $\varrho = \sigma$, for any $1 \leq i \leq l$, $1 \leq j \leq n$;
  - $s = (p_{j_1}^{i_1}, p_{j_2}^{i_2})$ where $p_{j_1} \leq p_{j_2}$, $\varrho = \sigma$.

We observe some properties of a solution φ to $\mathcal{P}^{(l)}$.

(1) If $p_{j_1} \leq p_{j_2}$, $1 \leq i_1, i_2 \leq l$ and $\varphi(p_{j_1}^{i_1}) \in B$ then $\varphi(p_{j_2}^{i_2}) \in B \cup D_b$.

(2) For any $p_j^{i_1}, p_j^{i_2}$, either both $\varphi(p_j^{i_1}), \varphi(p_j^{i_2})$ belong to D or both belong to $B \cup D_b$.

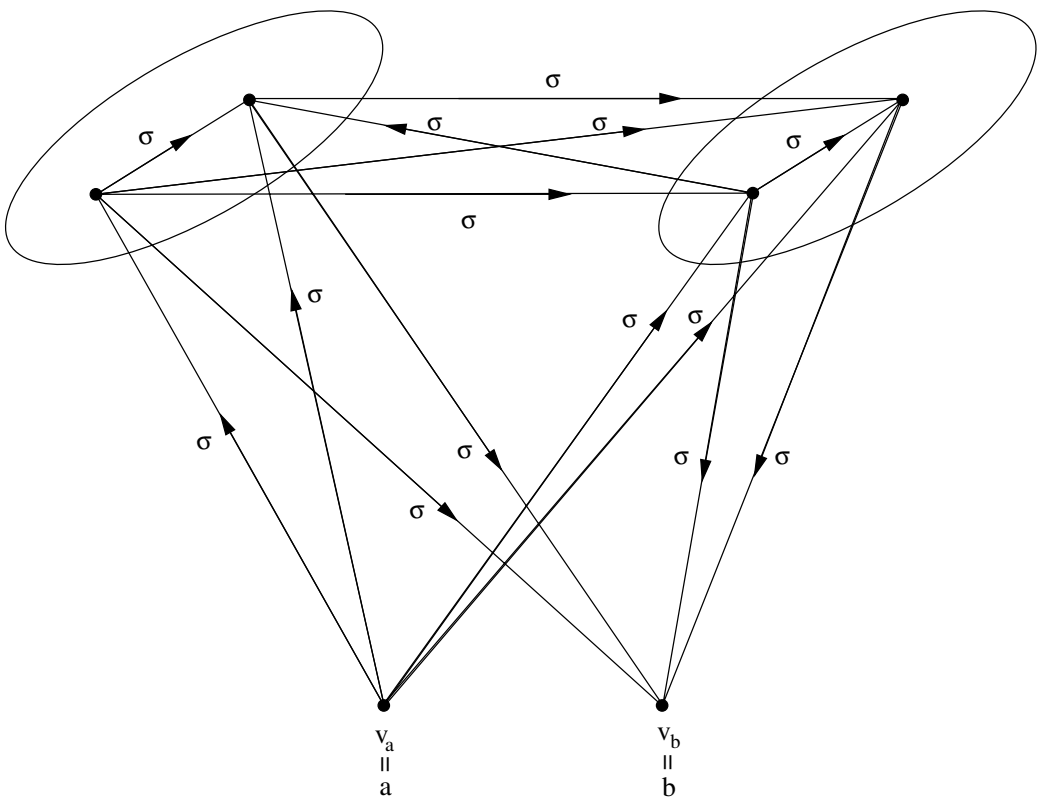


Figure 2:

Furthermore, let H_φ be the set of elements $p_j \in P$ minimal with the property that $\{\varphi(p_j^1), \ldots, \varphi(p_j^l)\}$ contains elements from B. Clearly H_φ is an antichain, so, let F_φ denote the corresponding filter. By properties (1),(2), for any p_j^i such that $p_j \in F\varphi$, we have $\varphi(p_j^i) \in B \cup D_b$, and for any p_j^i with $p_j \notin F\varphi$, we have $\varphi(p_j^i) \in D$.

Moreover, if $p_j \notin H_\varphi$ then there are no further restrictions on $\varphi(p_j^i)$; however, if $p_j \in H_\varphi$ then at least one of $\{\varphi(p_j^1), \ldots, \varphi(p_j^l)\}$ must belong to B. Therefore, for an antichain H and the corresponding filter F, the number of solutions φ to $\mathcal{P}^{(l)}$ such that $H_\varphi = H$, $F_\varphi = F$, equals

$$(|D|^{|P-F|})^l \cdot ((|B \cup D_b|^{|H|})^l - (|D_b|^{|H|})^l) \cdot (|B \cup D_b|^{|F-H|})^l.$$

Let $N(u, v, w)$ denote the number of filters F of P such that $|P - F| = u$, $|H| = v$, $|F - H| = w$, and $g = |B|$, $h = |D|$, $f = |D_b|$. The number N_l of solutions to $\mathcal{P}^{(l)}$ satisfies the identity

$$\sum_{u+v+w=|P|} N(u, v, w) h^{lu}((f+g)^{lv} - f^{lv})(f+g)^{lw} = N_l.$$

The number of all antichains is the sum of all numbers of the form $N(u, v, w)$. So, let p be the number of triples (u, v, w) such that $u + v + w = |P|$. The number of antichains can be found as the sum of solutions to the linear system

$$\begin{aligned}
\sum_{u+v+w=|P|} N(u, v, w) h^{u}((f+g)^{v} - f^{v})(f+g)^{w} &= N_1 \\
\sum_{u+v+w=|P|} N(u, v, w) h^{2u}((f+g)^{2v} - f^{2v})(f+g)^{2w} &= N_2 \\
&\vdots \\
\sum_{u+v+w=|P|} N(u, v, w) h^{pu}((f+g)^{pv} - f^{pv})(f+g)^{pw} &= N_p.
\end{aligned}$$

However, it is not obvious that the determinant of this system does not equal 0. To avoid calculating this determinant, we duplicate each variable and double the number of equations

$$\begin{aligned}
&\sum_{u+v+w=|P|} N'(u, v, w) h^{u}(f+g)^{v}(f+g)^{w} - \\
&\qquad \sum_{u+v+w=|P|} N''(u, v, w) h^{u} f^{v}(f+g)^{w} = N_1 \\
&\sum_{u+v+w=|P|} N'(u, v, w) h^{2u}(f+g)^{2v}(f+g)^{2w} - \\
&\qquad \sum_{u+v+w=|P|} N''(u, v, w) h^{2u} f^{2v}(f+g)^{2w} = N_2 \\
&\vdots
\end{aligned}$$

$$\sum_{u+v+w=|P|} N'(u,v,w)h^{2pu}(f+g)^{2pv}(f+g)^{2pw} - \sum_{u+v+w=|P|} N''(u,v,w)h^{2pu}f^{2pv}(f+g)^{2pw} = N_{2p}.$$

Similarly to the proof of Theorem 2, we may assume all the numbers $h^u(f+g)^v(f+g)^w$, $h^u f^v(f+g)^w$ to be different, and then the matrix of this system is Vandermonde. Finally, the sum of the solutions to the system is the double number of antichains. □

Theorems 7,9, and results from [30] provide a link between the complexity of #CSP and Mal'tsev operations. The next statement follows from Theorem 9.13 of [30].

Theorem 10 ([30]) *For a finite algebra $\mathbb{A}$ the following conditions are equivalent.*

1. $\mathbb{A}$ does not have a Mal'tsev term operation.

2. There is $\mathbb{B} = (B; F)$, a homomorphic image of a subalgebra of a direct power of $\mathbb{A}$, such that Inv *F contains a binary reflexive but not symmetric relation.*

An algebra with a Mal'tsev term operation is said to be *Mal'tsev*.

By Theorem 9, the algebra $\mathbb{B}$ from Theorem 10(2) is #P-complete. Furthermore, Theorem 7 implies that $\mathbb{A}$ is also #P-complete.

Corollary 2 *If $\mathbb{A}$ is a finite algebra that has no Mal'tsev term operation then $\mathbb{A}$ is #P-complete.*

Finally, Theorem 5 is just Corollary 2 expressed in terms of constraint languages.

5 Applications

5.1 Small domains

By making use of Theorem 5 we may obtain a very easy proof of the dichotomy theorem for #SAT [12]. On the one hand, by the results of [45], if a Boolean constraint language Γ is invariant with respect to a Mal'tsev operation it is also invariant with respect to $x - y + z$. Therefore, any #-tractable Boolean constraint language is invariant with respect to $x-y+z$. On the other hand, any relation from such a language is the solution space of a system of linear equation over a 2-element field. Hence it is possible to find its basis in polynomial time, and furthermore, the number of solutions equals 2^n where n is the number of vectors in the basis.

Mal'tsev operations on a 3-element set and the structure of relations invariant with respect to them are exhaustively studied in [5]. Making use of these results we are able to prove Conjecture 1 for constraint languages over a 3-element set.

Theorem 11 *Every 3-element Mal'tsev algebra is #-tractable.*

We set up the proof of Theorem 11 in several lemmas and claims.

It is well known and easy to see that a 3-element Mal'tsev algebra may have at most one non-trivial congruence. Therefore, when studying 3-element Mal'tsev algebras we have 2 cases to consider: simple Mal'tsev algebras, i.e. without proper congruences, and Mal'tsev algebras with a unique proper congruence. Every Mal'tsev algebra without a proper congruence is uniform hence, by Theorem 14, any simple Mal'tsev algebra is #-tractable. So consider a Mal'tsev algebra $\mathbb{A}$ with a proper congruence μ. Without loss of generality, the universe of $\mathbb{A}$ is $A = \{0, 1, 2\}$, and the blocks of μ are $\{0, 1\}, \{2\}$. In this case $\mathbb{A}$ has, as a term operation, the Mal'tsev operation d [5]:

$d(x,y,0)$	1	2	3
1	0	1	2
2	1	0	2
3	2	2	0

$d(x,y,1)$	1	2	3
1	1	0	2
2	0	1	2
3	2	2	1

$d(x,y,2)$	1	2	3
1	2	2	0
2	2	2	1
3	0	1	2

This means that it is enough to consider the algebra $\mathbb{A} = (\{0, 1, 2\}; d)$.

As is easily seen, $\mathbb{A}$ has 3 two-element subalgebras, that is every two-element subset of A gives rise to a subalgebra. We denote these subalgebras by $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3$. All three 2-element subalgebras of $\mathbb{A}$ and $\mathbb{A}/_\mu$ are isomorphic to the algebra $\mathbb{D} = (\{0, 1\}; x + y + z)$ where $+$ is the operation of the 2-element group. Let $\varphi_1, \varphi_2, \varphi_3$ be isomorphism of $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3$ to $\mathbb{D}$ respectively, and φ the homomorphism of $\mathbb{A}$ to $\mathbb{D}$ taking $0, 1$ to 0, and 2 to 1. Every relation from $\mathsf{Inv}\,\{x + y + z\}$ is the solution space of a system of linear over GF(2).

Let $(V; A; \mathcal{C})$, $V = \{v_1, \ldots, v_n\}$, be a problem instance from #CSP($\mathbb{A}$), $\mathcal{S}$ its solution space. As usual, we treat $\mathcal{S}$ as an n-ary relation. Generally, the size of $\mathcal{S}$ exponentially depends on the number of variables. However, it is shown in [6] that it is possible to find in polynomial time a *basis* B of $\mathcal{S}$ so that the size of B is bounded by a polynomial on the number of variables, and every member of $\mathcal{S}$ can be obtained from B in a certain uniform way.

By $\mathcal{S}_i$ we denote the set of partial solutions for the variable v_i, i.e. the intersection of the projections of the constraint relations onto v_i. Let also W be the set of those variables $v_i \in V$ for which $\mathcal{S}_i = \mathbb{A}$. We define the *factor-problem* $(V; D; \mathcal{C}')$ as follows:

- $D = \{0, 1\}$;
- for each $C = \langle s, \varrho \rangle \in \mathcal{C}$, $s = (v_{i_1}, \ldots, v_{i_m})$, there is $C' = \langle s, \varrho' \rangle \in \mathcal{C}'$ where

$$\begin{aligned} \varrho' &= \{(\mathbf{b}[1], \ldots, \mathbf{b}[m]) \in D^m \mid (\mathbf{a}[1], \ldots, \mathbf{a}[m]) \in \varrho\}, \\ \mathbf{b}[i] &= \begin{cases} \varphi_j(\mathbf{a}[l]), & \text{if } \mathcal{S}_{i_l} = \mathbb{A}_j,\ j = 1, 2, 3, \\ \varphi(\mathbf{a}[l]), & \text{if } \mathcal{S}_{i_l} = \mathbb{A}, \end{cases} \end{aligned}$$

and its solution space we denote by $\mathcal{S}'$. The factor problem is equivalent to a system of linear equations over GF(2) and its solution space posesses all properties of cosets of finite vector spaces. In particular, every partial solution $\mathbf{a} \in \mathrm{pr}_W \mathcal{S}'$ has the same number of extensions to a solution from $\mathcal{S}'$.

Lemma 1 *For any* $\mathbf{a} \in \mathcal{S}$*, any tuple* $\mathbf{b}$ *such that* $\mathbf{b}[l] = \mathbf{a}[l]$ *if* $v_l \notin W$ *and* $(\mathbf{b}[l], \mathbf{a}[l]) \in \mu$ *otherwise, is also a solution to* $\mathcal{P}$*.*

Lemma 1 follows straightforwardly from Lemma 3.7 of [5]. Let R be the number of extensions of partial solutions from $\mathrm{pr}_W \mathcal{S}'$, and, by Lemma 1, of partial solutions from $\mathrm{pr}_W \mathcal{S}$.

Corollary 3 $|\mathcal{S}| = R \cdot |\mathrm{pr}_W \mathcal{S}|$.

We, therefore, may restrict ourselves with the case $W = V$, that is $\mathcal{S}_l = \mathbb{A}$ for any $l \in \{1, \ldots, n\}$. In this case $\mathcal{S}$ is said to be a *subdirect power* of $\mathbb{A}$.

The rectangularity of $\mathcal{S}$ implies that, for any $i \in \{1, \ldots, n\}$ the relation $\theta_i = \{(a, b) \in \mathbb{A}^2 \mid$ there are $a[1], \ldots, a[i-1], a[i+1], \ldots, a[n]$ such that $(a[1], \ldots, a[i-1], a, a[i+1], \ldots, a[n])$, $(a[1], \ldots, a[i-1], b, a[i+1], \ldots, a[n]) \in \mathcal{S}\}$ is a congruence of $\mathbb{A}$. If θ_i is the equality relation, then $|\mathcal{S}| = |\mathrm{pr}_{\{1,\ldots,i-1,i+1,\ldots,n\}} \mathcal{S}|$, and, as we are interested in the number of tuples in relations, we may replace $\mathcal{S}$ with $\mathrm{pr}_{\{1,\ldots,i-1,i+1,\ldots,n\}} \mathcal{S}$. We show how identify such superfluous variables.

We make use of the digraph $\mathcal{G}(\mathcal{S}')$. The nodes of $\mathcal{G}(\mathcal{S}')$ are the variables from V. Nodes v_{i_1}, v_{i_2} are connected with an edge if and only if there is an mapping $\pi : D \to D$ such that, for any $\mathbf{a} \in \mathcal{S}'$, $\mathbf{a}[i_2] = \pi(\mathbf{a}[i_1])$. For any strongly connected component $U = \{v_{i_1}, \ldots, v_{i_m}\}$ of $\mathcal{G}(\mathcal{S}')$, there are $\pi_l \colon \mathbb{D} \to \mathbb{D}$, for $l \in \{1, \ldots, m\}$ such that

$$\mathrm{pr}_U \mathcal{S}' = \{(\pi_1(a), \pi_2(a), \ldots, \pi_m(a)) \mid a \in D\}.$$

Then U naturally splits into two parts: the first consists of those v_{i_l} for which π_l is the identity mapping, and the second of those for which it is not. Let $V_1, \ldots, V_k$ be the list of those parts of the strongly connected components of $\mathcal{G}(\mathcal{S}')$.

Lemma 2 ([5]) *If a tuple* $\mathbf{a}$ *belongs to* $\mathcal{S}$ *then any tuple* $\mathbf{b}$ *such that* $(\mathbf{a}[l], \mathbf{b}[l]) \in \mu$*,* $l \in \{1, \ldots, n\}$*, and* $\mathrm{pr}_{V_j} \mathbf{b} \in \mathrm{pr}_{V_j} \mathcal{S}$ *for* $j \in \{1, \ldots, k\}$*, also belongs to* $\mathcal{S}$*.*

The lemma shows that if θ_i is the equality relation for a certain i then the value of v_i is completely determined by the values of variables from the set V_j containing v_i. Notice that the strongly connected components of $\mathcal{G}(\mathcal{S})$ and the mappings π_l can be extracted from the basis of $\mathcal{S}$.

To study interaction of values of variables inside sets V_l, for every V_l we consider the restricted problem instance $\mathcal{P}_l = (V_l; \{0, 1\}; \mathcal{C}_l)$ where for each $C = \langle s, \varrho \rangle \in \mathcal{C}$ we include in $\mathcal{C}_l$ the constraint $C_l = \langle s_l, \varrho_l \rangle$ with

- $s_l = s \cap V_l$;
- $\varrho_l = (\mathrm{pr}_{s_l} \varrho) \cap \{0, 1\}^{|s_l|}$.

Since each ϱ_l is invariant under the operation $x + y + z$ over $\{0, 1\}$, the instance $\mathcal{P}_l$ is equivalent to a linear system over GF(2). Denote its solution space by $\mathcal{S}_l$.

The dimension of the underlying vector space for $\mathcal{S}_l$ is $|V_l|$. Let us denote the dimension of $\mathcal{S}_l$ by r_l; clearly $r_l \leq |V_l|$. This means that there are r_l 'free' variables in V_l such that

they may take arbitrary values from $\{0,1\}$ while values of the remaining variables are uniquely determined by the values of the free variables. These variables can be removed from the instance without changing the number of solutions. Denote the set of the free variables from V_l by V_l'. We get the following

Lemma 3 *For any tuples* $\mathbf{a}, \mathbf{b} \in \mathcal{S}$, $\mathbf{a} = \mathbf{b}$ *if and only if* $\mathbf{a}[i] = \mathbf{b}[i]$ *for all* $v_i \in V_1' \cup \ldots \cup V_k'$.

Restricting $\mathcal{S}$ onto the set $V_1' \cup \ldots \cup V_k'$ of variables, $\theta_i \geq \mu$ can be supposed for any i. This observation implies

Lemma 4 *For* $\mathbf{a} \in \mathcal{S}$, *any tuple* $\mathbf{b}$ *such that* $a[i] \stackrel{\mu}{\equiv} b[i]$, *for any* i, *belongs to* $\mathcal{S}$.

Furthermore, we have $\varphi(\mathcal{S}) \cong \mathcal{S}/_{\mu} \subseteq \mathbb{D}^n$. As for the number of tuples, we get

Lemma 5 *(1) For any* $\mathbf{b} \in \varphi(\mathcal{S})$, *there are* 2^k *tuples* $\mathbf{a}$ *in* $\mathcal{S}$, *such that* $\varphi(\mathbf{a}) = \mathbf{b}$, *where* k *is the number of zeroes in* $\mathbf{b}$.
(2) $|\mathcal{S}| = \sum_{i=1}^{n} N_i 2^i$ *where* N_i *is the number of tuples in* $\varphi(\mathcal{S})$ *with* i *zeroes.*

Thus, #CSP($\mathbb{A}$) is equivalent to evaluating the expression of the form $\sum_{i=1}^{n} N_i 2^i$ where N_i is the number of vectors from a linear manifold over GF(2) with i 0s. The following lemma completes the proof of Theorem 11.

Lemma 6 *Let* L *be a system of linear equations in* n *variables over* GF(2), *and* N_i *the number of solutions to* L *with* i *zeroes. Then the* N_i *are computable in polynomial time.*

Proof: Let us denote the solution space of L by $\mathcal{S}$, and the dimension of $\mathcal{S}$ by k. It is easy to distinguish if $(1,\ldots,1) \in \mathcal{S}$, and we restrict ourselves with the case $(1,\ldots,1) \notin \mathcal{S}$. The remaining case is quite similar. Clearly, $N_0 + \ldots + N_n$ is the total number of solutions to L; as $(1,\ldots,1) \notin \mathcal{S}$, $N_1 + \ldots + N_n = 2^k$.

Claim 1. If the solution space $\mathcal{S}$ of a linear system L in n variables has the dimension $k > 0$, then the total number of zeroes in tuples from $\mathcal{S}$ equals $n \cdot 2^{k-1}$.

The dimension k means that there are k 'free' variables, say $x_1, \ldots, x_k$, that can take arbitrary values, and the value of each of the remaining $n-k$ variables is computable via a linear function of $x_1, \ldots, x_k$. Clearly, the total number of zeroes among the values of $x_1, \ldots, x_k$ is $k2^{k-1}$. Furthermore, a linear function in GF(2) takes the value 0 on exactly half of the assignments to its arguments. Therefore, each of the variables $x_{k+1}, \ldots, x_n$ contributes 2^{k-1} zeroes. So, the number of zeroes in tuples from $\mathcal{S}$ equals $n2^{k-1}$.

For each q, $1 < q \leq n$, consider the linear system L^q obtained from L by adding, for each sequence $(i_1, \ldots, i_q) \in \{1, \ldots, n\}^q$, the equation

$$x_{i_1,\ldots,i_q} + x_{i_1} + \ldots + x_{i_q} = 0.$$

CLAIM 2. The rank of L^q is k, and each solution to L has a unique extension to a solution to L^q.

Indeed, the added equations allow one to express the new variables through the old ones, and impose no restrictions onto possible values of the old variables.

CLAIM 3. For every solution to L containing l zeroes, there exist

$$l^q + \binom{2}{q} l^{q-2}(n-l)^2 + \ldots + \binom{2i}{q} l^{q-2i}(n-l)^{2i} + \ldots$$

zeroes in its extension to a solution of L^q.

Take a solution $(a_1, \ldots, a_n)$ to L containing l zeroes. Without loss of generality, let $a_1 = \ldots = a_l = 0$, and $a_{i_1,\ldots,i_q}$, for $(i_1, \ldots, i_q) \in \{1, \ldots, n\}^q$, form the extension of this solution. Notice that $a_{i_1,\ldots,i_q} = 0$ if and only if the sequence $(a_{i_1}, \ldots, a_{i_q})$ contains an even number of units. Calculate, how many sequences of this form contain $2i$ units. The set of positions where the units stand can be chosen in $\binom{2i}{q}$ ways. These positions can be with by any elements (not necessarily different) from $\{l+1, \ldots, n\}$. The vacant positions can be filled by any elements from $\{1, \ldots, l\}$. Thus, we get $\binom{2i}{q} l^{q-2i}(n-l)^{2i}$ sequences. Summarizing for all i from 0 to $\lfloor q/2 \rfloor$, we get the required number.

Therefore, the number of zeroes in solutions to L^q, on the one hand, equals

$$n2^{k-1} + \sum_{l=1}^{n} \left(l^q + \binom{2}{q} l^{q-2}(n-l)^2 + \ldots + \binom{2i}{q} l^{q-2i}(n-l)^{2i} + \ldots \right) N_l;$$

and, on the other hand, by Claim 2, this number equals $(n^q + n)2^{k-1}$. So, for each q, we have an equation

$$\sum_{l=1}^{n} \left(l^q + \binom{2}{q} l^{q-2}(n-l)^2 + \ldots + \binom{2i}{q} l^{q-2i}(n-l)^{2i} + \ldots \right) N_l \qquad (1)$$
$$= \quad (n^q + n)2^{k-1} - n2^{k-1}.$$

Consider the system in unknowns $N_1, \ldots, N_n$ consisting of the equation

$$N_1 + \ldots + N_n = 2^{k-1},$$

the equation

$$1 \cdot N_1 + 2 \cdot N_2 + \ldots + n \cdot N_n = n2^{k-1}$$

(see Claim 1), and the equations of the form (1) for each $1 < q \leq n-1$. This system can be generated in polynomial tyme and contains n variables. Therefore, the proof will be completed if we show that the determinant of this system is non-zero.

Denote the determinant by D_n, we have

$$D_n = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & n \\ 1^2+(n-1)^2 & 2^2+(n-2)^2 & \cdots & n^2 \\ 1^3+\binom{2}{3}(n-1)^2 & 2^3+\binom{2}{3}2^1(n-2)^2 & \cdots & n^3 \\ 1^4+\binom{2}{4}(n-1)^2+(n-1)^4 & 2^4+\binom{2}{4}2^2(n-2)^2+(n-2)^4 & \cdots & n^4 \\ \vdots & \vdots & & \vdots \end{vmatrix}.$$

Notice that for each l, q, $1 \le l \le n$, $2 < q < n-1$, the entry in lth column and qth row of D_n is the sum of odd summands of the binomial formula $(l+(n-l))^q$, or, equivalently, of the positive summands of $(l-(n-l))^q$. The determinant

$$E_n = \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1-(n-1) & 2-(n-2) & 3-(n-3) & \cdots & n-(n-n) \\ (1-(n-1))^2 & (2-(n-2))^2 & (3-(n-3))^2 & \cdots & (n-(n-n))^2 \\ (1-(n-1))^3 & (2-(n-2))^3 & (3-(n-3))^3 & \cdots & (n-(n-n))^3 \\ (1-(n-1))^4 & (2-(n-2))^4 & (3-(n-3))^4 & \cdots & (n-(n-n))^4 \\ \vdots & \vdots & \vdots & & \vdots \end{vmatrix}.$$

is Vandermonde, and therefore non-zero. Finally, by routine calculation we obtain the formula $E_n = 2^{n-1}D_n$, and conclude that $D_n \neq 0$. □

Stating Theorem 11 for constraint languages we get the following statement.

Corollary 4 *A constraint language Γ over a 3-element set is #-tractable if and only if* Pol Γ *contains a Mal'tsev operation.*

5.2 Rectangularity and permutability

Relations invariant with respect to a Mal'tsev operation satisfy strong restrictions on their form. One of them is especially useful. An (n-ary) relation ϱ is said to be *rectangular* if, for any partition of $\{1,\dots,n\}$ into subsets I, J, and any $\mathbf{a},\mathbf{b} \in \mathrm{pr}_I\varrho$, $\mathbf{c},\mathbf{d} \in \mathrm{pr}_J\varrho$, if $(\mathbf{a},\mathbf{c}),(\mathbf{a},\mathbf{d}),(\mathbf{b},\mathbf{c}) \in \varrho$, then $(\mathbf{b},\mathbf{d}) \in \varrho$, see Fig. 3 ($(\mathbf{a},\mathbf{c})$ here denotes the tuple $\mathbf{e}$ such that $\mathbf{e}[i] = \mathbf{a}[i]$ if $i \in I$ and $\mathbf{e}[i] = \mathbf{c}[i]$ if $i \in J$).

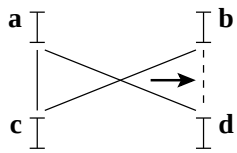


Figure 3: The property of rectangularity

Indeed,if $(\mathbf{a},\mathbf{c}),(\mathbf{a},\mathbf{d}),(\mathbf{b},\mathbf{c}) \in \varrho$ then

$$m\left(\begin{pmatrix}\mathbf{a}\\ \mathbf{d}\end{pmatrix},\begin{pmatrix}\mathbf{a}\\ \mathbf{c}\end{pmatrix},\begin{pmatrix}\mathbf{b}\\ \mathbf{c}\end{pmatrix}\right) = \begin{pmatrix}\mathbf{b}\\ \mathbf{d}\end{pmatrix} \in \varrho.$$

By Theorem 5, we get

Corollary 5 *If Γ is a #-tractable constraint language, then every relation from Γ is rectangular. If Γ contains a non-rectangular relation, then Γ is #P-complete.*

Unfortunately, the rectangularity of a relation or a set of relations does not guarantee existence of a Mal'tsev polymorphism. For instance, let Γ be a constraint language, and $\varrho_1, \varrho_2 \in \langle\Gamma\rangle$ equivalence relations. Clearly, ϱ_1, ϱ_2 are rectangular. However, if Γ has a Mal'tsev polymorphism, then they also must be *permutable*, that is $\varrho_1 \circ \varrho_2 = \varrho_2 \circ \varrho_1$. A proof of this fact is non-trivial and can be found e.g. in [11, 39].

5.3 #H-Coloring problem

The problem Graph homomorphism [24] and its counting counterpart are among the most well established combinatorial problems. H-Coloring and #H-Coloring problems constitute their subproblems when the target graph H is fixed. A massive work has been done in the study of the complexity of H-Coloring and #H-Coloring for different types of the graph H, and also for restrictions on the class of source graphs [1, 2, 17, 18, 19, 26, 27, 29, 52].

In particular, undirected graphs H for which H-Coloring is tractable have been characterized in [28]. An analogous result for #H-Coloring has been obtained by Dyer and Greenhill.

Theorem 12 ([20]) *If every connected component of an undirected graph H is either an isolated vertex, or a complete graph with all loops present, or a complete unlooped bipartite graph, then the #H-Coloring is tractable. Otherwise, it is #P-complete.*

Undirected graphs correspond to symmetric binary relations. Observe that graphs of rectangular symmetric binary relations are precisely those specified in Theorem 12. Indeed, a complete graph with all loops present correspond to the total binary relation which is rectangular, and if ϱ is a relation corresponding to a complete bipartite graph and $(a,c),(a,d),(b,c) \in \varrho$ then b,d are in different parts and therefore the edge (b,d) is present.

Conversely, let H be a connected graph such that ϱ_H is rectangular. Suppose first that H has a loop at vertex a and (a,b) is an edge. Then $(a,a),(a,b),(b,a) \in \varrho_H$ and, by the rectangularity, we get $(b,b) \in \varrho_H$. Therefore, all loops in H are present. Furthermore, if $(a,b),(b,c)$ is a path then $(b,a),(b,c),(a,a) \in \varrho_H$ implies $(a,c) \in \varrho_H$, and we conclude that H is complete. If H has no loops then notice that, for any path $(a,b),(b,c),(c,d)$, the edge (a,d) is also present, because $(a,b),(c,b),(c,d) \in \varrho_H$. Therefore if H contains a cycle of odd length then it also has a loop, a contradiction. Thus H is bipartite, and the completeness follows straightforwardly from the rectangularity.

Making use of Theorem 5, one may easily derive the #P-completeness part of Theorem 12. The tractability part of this theorem is fairly simple.

Specifying Conjecture 1 to the case of #H-Coloring we are able to conjecture a criterion for digraphs which give rise to tractable #H-Coloring problems.

Conjecture 2 *The #H-*Coloring *problem for a digraph H is tractable if and only if ϱ_H is invariant under a Mal'tsev operation.*

Unfortunately, it is not easy to express the criterion in graph-theoretic terms. However one necessary condition can be identified. A digraph is said to be *N-free* if it does not contain a subgraph of the form shown in Fig. 4.

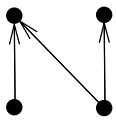


Figure 4:

Lemma 7 *If a digraph H is not N-free, then the #H-*Coloring *problem is #P-complete.*

In fact, the effect of the property of rectangularity is that it excludes the configuration of the form shown in Fig. 4.

Theorem 12 confirms Conjecture 2 for undirected graphs. For some classes of digraphs this conjecture can also be proved.

Proposition 4 *Let H be an oriented cycle. Then the #H-*Coloring *problem is tractable if and only if H is one of C_n, C_n^1, C_n^2 (see Fig. 5). Otherwise it is #P-complete.*

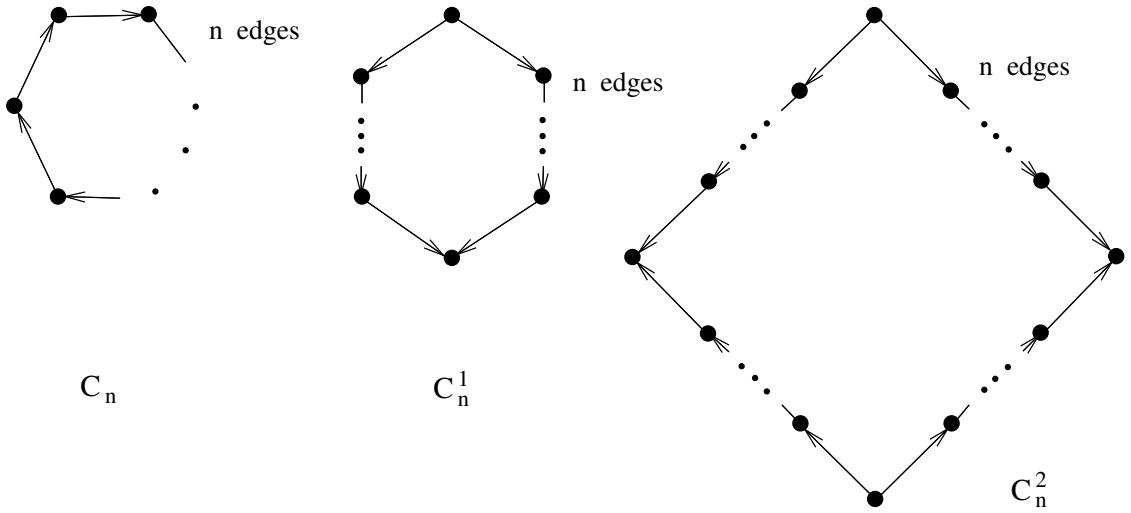


Figure 5:

Proof: Let $H = (V; E)$ be an oriented cycle. We prove that if there is a Mal'tsev polymorphism of ϱ_H then H is of the form specified in the theorem.

Clearly, if H is not a directed cycle then H contains no directed cycles. So, we shall assume that H has no directed cycles and N-free. By α_H we denote the relation $\varrho_H^{-1} \circ \varrho_H$,

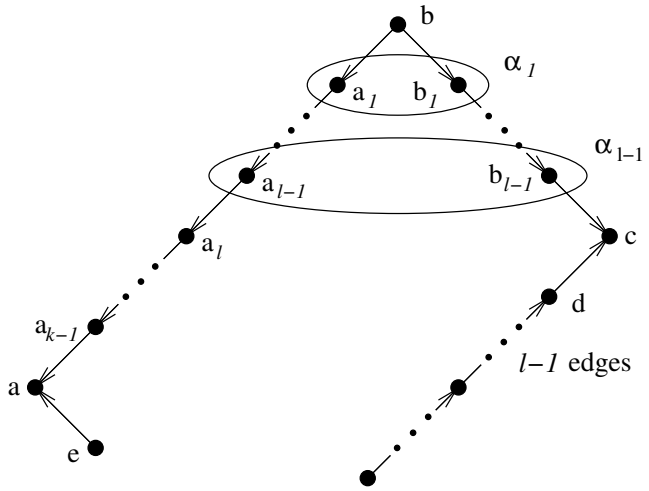


Figure 6:

an equivalence relation on the set V^+ of vertices of non-zero indegree. For any $v, w \in V$ with $(v, w) \in \alpha_H$, there is $u \in V$ such that $(u, v), (u, w) \in \varrho_H$. Moreover, since H is N-free, if $(u', v) \in \varrho_H$, then $(u', w) \in \varrho_H$. Let us denote by H_α the homomorphic image of the subgraph spanned by V^+ with kernel α_H.

If H contains no vertex of zero indegree then $H = C_{|V|}$. Otherwise, H may contain several directed paths. The maximal ones are the paths from a vertex of zero indegree to the nearest vertex of zero outdegree.

Take a vertex b of zero indegree and maximal directed paths $b = a_0, a_1, \ldots, a_k = a$ and $b = b_0, b_1, \ldots, b_l = c$ originated at b. Suppose that $l \leq k$, and $b_0, \ldots, b_l$ is the shortest maximal path. Then a, c are the vertices of zero outdegree. Let us denote $H_0 = H$, α_0 is $=_V$, and $H_i = (H_{i-1})_{\alpha_i}$, $\alpha_i = \alpha_{H_{i-1}}$. Clearly, we may treat the α_i as relations on V. Then $(u, v) \in \alpha_i$ if and only if there exist $w \in V$ and directed paths of the length i connecting w with u and v.

Therefore, $(a_0, b_0) \in \alpha_0, (a_1, b_1) \in \alpha_1, \ldots, (a_{l-1}, b_{l-1}) \in \alpha_{l-1}$ (see Fig. 6). Notice that, since l is the minimal length of a maximal path, the pair (d, d) is in α_{l-1}. Furthermore,

$$m\left(\begin{pmatrix} a_{l-1} \\ a_l \end{pmatrix}, \begin{pmatrix} b_{l-1} \\ c \end{pmatrix}, \begin{pmatrix} d \\ c \end{pmatrix}\right) \in \varrho_H,$$

which means $m(a_{l-1}, b_{l-1}, d) = g \in \{a_{l-1}, e\}$. Then, as m preserves α_{l-1}, we have

$$m\left(\begin{pmatrix} a_{l-1} \\ a_{l-1} \end{pmatrix}, \begin{pmatrix} b_{l-1} \\ a_{l-1} \end{pmatrix}, \begin{pmatrix} d \\ d \end{pmatrix}\right) = \begin{pmatrix} g \\ d \end{pmatrix} \in \alpha_{l-1}.$$

Therefore, there is a vertex v of zero indegree and paths of legth $l-1$ from v to g and d. If $g = a_{l-1}$ then $v = b$, $c = a$ and $H = C_l^1$. If $g = e \neq a_{l-1}$ then $H = C_l^2$.

We sketch counting algorithms for all three types of graphs. Suppose that $G = (V', E')$ is a connected graph such that there exists a homomorphism from G to H.

If $H = C_n$, then the algorithm is trivial: a homomorphism is completely determined by the image of any vertex of G.

Suppose that $H = C_n^1$, and let $a_0, a_1, \ldots, a_n$, $b_0, b_1, \ldots, b_n$ denote the maximal paths, $a_0 = b_0$, $a_n = b_n$. It is not hard to see that there exists a 'level' function $\nu: V' \to \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers, such that if $(a, b) \in E'$ then $\nu(b) = \nu(a) + 1$. Let us denote $M = \min_{v \in V'} \nu(v)$, $N = \max_{v \in V} \nu(v)$, and $K = N - M$. Clearly, $K \leq n$.

A homomorphism $\varphi: G \to H$ can be of one of the three types.

(1) For some $v, w \in V'$ such that $\nu(v) = M$, $\nu(w) = N$, we have $\varphi(v) \neq a_0$ and $\varphi(w) \neq a_n$.

(2) For some $v \in V$ with $\nu(v) = M$, $\varphi(v) \neq a_0$, but for any $w \in V$ with $\nu(w) = N$, $\varphi(w) = a_n$. [For some $w \in V$ with $\nu(w) = N$, $\varphi(w) \neq a_n$, but for any $v \in V$ with $\nu(v) = M$, $\varphi(v) = a_0$.]

(3) For any $v, w \in V$ with $\nu(v) = M$, $\nu(w) = a_n$, we have $\varphi(v) = a_0$, $\varphi(w) = a_n$.

Clearly, an algorithm of type (1) maps G into one the paths, and therefore is completely determined by the image of any vertex of G.

To count the homomorphisms of type (2), remove from G all vertices w with $\nu(w) = N$. As is easily seen any homomorphism φ of this type maps every connected component of the obtained graph into one of the paths. Moreover, φ is uniquely determine on each connected component by the image of any vertex from this component, and images of different connected components are independent.

For homomorphisms of type (3) we remove from G all vertices v with $\nu(v) = M$ or $\nu(v) = N$, and then do as in the previous case.

Finally, consider the case $H = C_n^2$. Let $a_0, a_1, \ldots, a_n$, $b_0, b_1, \ldots, b_n$, $c_0, c_1, \ldots, c_n$, $d_0, d_1, \ldots, d_n$ be the 4 maximal paths, and $a_0 = b_0$, $a_n = c_n$, $c_0 = d_0$, $b_n = d_n$. Similar to the previous case, if there is a homomorphism mapping G to H, then there exists a 'level' function $\nu: V' \to \mathbb{N}$, and every homomorphism φ is of one of the same 3 types, but we replace a_0 with $\{a_0, c_0\}$ and a_n with $\{a_n, b_n\}$.

The algorithms for homomorphisms of type (1) are exactly the same as in the previous case. Moreover, if φ is of type (2) then, since G is connected, either all vertices $w \in V'$ with $\nu(w) = N$ are mapped to a_n, or all vertices $w \in V'$ with $\nu(w) = N$ are mapped to b_n. Therefore, the number of homomorphisms of type (2) can be computed in a similar way.

Let φ be of type (3). Let us denote $A_0 = \{v \in V' \mid \nu(v) = M\}$, $A_n = \{v \in V' \mid \nu(v) = N\}$, and $C_1, \ldots, C_k$ the connected components of the graph obtained by removing from G the vertices from A_0 and A_n. As before, φ maps every C_i into one of the 4 paths. However, if, for C_i, C_j there are $v \in A_0$, $w \in A_n$, and edges from v to C_i, C_j and from C_i, C_j to W, then φ maps C_i, C_j into the same path. Let δ_0 denote the relation on $\{C_1, \ldots, C_k\}$ such that $(C_i, C_j) \in \delta_0$ iff there are v, w with the properties specified, and let δ_0^* denote the transitive closure of δ_0. Analogously, if for $v_1, v_2 \in A_0$ there are C_i, and edges from v_1, v_2 to C_i, then $\varphi(v_1) = \varphi(v_2)$. We therefore introduce relations η_0, θ_0 on A_0, A_n respectively, and their transitive closures η_0^*, θ_0^*. Let also $B_1, \ldots, B_l$ be the unions of the classes of δ_0^*.

Let G_1 denote the homomorphic image of G obtained by collapsing vertices from the δ_0^*-, η_0^*-, and θ_0^*-classes. Clearly, we may repeat the prosedure with the graph G_1 and obtain relations δ_1^*, η_1^*, θ_1^*, and so on. Suppose that m is such that δ_m^*, η_m^*, θ_m^* are the equality relations, and $G_m = (W_1 \cup W_2 \cup W_3, E'')$ where W_1 is the set of vertices of zero indegree and W_2 is the set of vertices of zero outdegree. Make a new graph $G' = (W_1 \cup W_2, E''')$ such that $(v, w) \in E'''$, $v \in W_1, w \in W_2$, if and only if there is a directed path from v to w in G_m.

It is straightforward that the number of homomorphisms of type (3) from G to H equals to that from G' to C_1^2. Moreover, the vertices from W_1 can be mapped arbitrarily to $\{a_0, c_0\}$ and the vertices from W_2 can be mapped arbitrarily to $\{a_1, b_1\}$ where a_0, c_0, a_1, b_1 are the vertices of C_1^2. □

5.4 List $\#H$-Coloring problem

In the List $\#H$-Coloring problem we are given an input (directed) graph G, and for every vertex g of G, a *list* $L(g)$, that is a subset of vertices of H. The objective is to find the number of homomorphisms $\varphi\colon G \to H$ such that $\varphi(g) \in L(g)$ for every vertex g of G. It is not hard to see that this problem is equivalent to $\#\mathrm{CSP}(\Gamma_H)$ where $\Gamma_H = \{\varrho_H\} \cup \{\sigma \mid \sigma \text{ is a unary predicate over the vertex set of } H\}$.

In the case of undirected graphs a dichotomy theorem was obtained independently in [18] and [27]. The criterion happened to be the same as that for $\#H$-Coloring.

Theorem 13 ([18, 27]) *If every connected component of an undirected graph H is either an isolated vertex, or a complete graph with all loops present, or a complete unlooped bipartite graph, then* List $\#H$-Coloring *is tractable. Otherwise, it is #P-complete.*

If List $\#H$-Coloring is tractable then a Mal'tsev polymorphism m of the constraint language Γ_H preserves every unary predicate. As is easily seen this is equivalent to the condition $m(x, y, z) \in \{x, y, z\}$ for any x, y, z. An operation satisfying this condition is called *conservative*. Therefore Conjecture 1 in the case of the List $\#H$-Coloring problem looks as follows.

Conjecture 3 *The* List $\#H$-Coloring *problem is tractable if and only if ϱ_H is invariant with respect to a conservative Mal'tsev operation. Otherwise it is #P-complete.*

Remarkably, in contrast to the case of undirected graphs, the classes of digraph specified in Conjectures 2 and 3 are not equal. For example, consider the digraph $H = C_2^2$ (Fig. 4). By Proposition 4, the $\#H$-Coloring problem can be solved in polynomial time, while ϱ_H has no conservative Mal'tsev polymorphism, which means that List $\#H$-Coloring is #P-complete. Indeed, let m be a conservative polymorphism of ϱ_H. The relations $\delta_1 = \varrho_H \circ \varrho_H^{-1}$ and $\delta_2 = \varrho_H^{-1} \circ \varrho_H$ are equivalence relations on the sets $\{a, b, c, d, g, h\}, \{a, b, c, d, e, f\}$, respectively, with classes $\{a, b\}, \{c, d\}, \{g\}, \{h\}$ and $\{a, d\}, \{b, c\}, \{e\}, \{f\}$, respectively. Since m preserves δ_1, δ_2, we have

$$m\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}, \begin{pmatrix} c \\ c \end{pmatrix}\right) = \begin{pmatrix} c \\ c \end{pmatrix} \qquad m\left(\begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}, \begin{pmatrix} c \\ b \end{pmatrix}\right) = \begin{pmatrix} a \\ a \end{pmatrix}.$$

This means $m(a,b,c) = c$ and $m(a,b,c) = a$, a contradiction.

5.5 Uniform constraint languages

In [6], it was shown that the decision constraint satisfaction problem arising from a constraint language invariant with respect to a Mal'tsev operation is tractable. Moreover, the algorithm presented in that paper allows one not only to determine whether there is a solution to a problem instance, and not only to find a solution, but also to find a basis of the solution space. In some cases we may find the number of solutions by making use of this basis.

A constraint language Γ is said to be *uniform* if the blocks of any equivalence relation from $\langle\Gamma\rangle$ are of the same size. Correspondingly, an algebra is said to be *uniform* if the blocks of any congruence of any its subalgebra are of the same size.

Clearly, every Boolean constraint language is uniform. Further examples of uniform constraint languages are provided by 'group constraints' [21, 22], and also by the constraints studied in [14].

Example 6 [Groups and Quasigroups] As is easily seen, for any group $\mathbb{G}$, the operation $m(x,y,z) = xy^{-1}z$ is Mal'tsev. This is also a standard exercise [39, 23] to show that a finite quasigroup $\mathbb{Q}$ has binary term operations $\backslash$ and $/$ of *left* and *right division* such that

$$\begin{array}{ll} x \cdot (x \setminus y) = y, & (x \,/\, y) \cdot y = x, \\ x \setminus (x \cdot y) = y, & (x \cdot y) \,/\, y = x. \end{array}$$

Moreover, the operations $p(x,y,z) = (x \,/\, (y \setminus y)) \cdot (y \setminus z)$, and $q(x,y,z) = ((x \cdot y) \,/\, x) \setminus (x \cdot z)$ are Mal'tsev.

Obviously, the less term operations an algebra has, the larger the corresponding constraint language is. To obtain as large as possible problem class, we therefore should try to restrict the set of term operations of an algebra as much as possible provided the algebra remains Mal'tsev. The poorest Mal'tsev algebras obtained from a group and a quasigroup are $(G; m)$, $(Q; p)$, $(Q; q)$ where G, Q are the base set of the group and the quasigroup respectively.

Proposition 5 *The algebras $(G; m)$, $(Q; p)$, $(Q; q)$ are uniform Mal'tsev algebras.*

Proof: Let $\mathbb{A}$ be one of the listed algebras, f one of the operations m, p, q, and α a congruence of a subalgebra $\mathbb{B}$ of $\mathbb{A}$. Notice first that if $(a,b) \in \alpha$ then

$$f\left(\begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} b \\ a \end{pmatrix}, \begin{pmatrix} c \\ c \end{pmatrix}\right) = \begin{pmatrix} f(a,b,c) \\ c \end{pmatrix} \in \alpha.$$

Therefore, for any classes A, C of α, and any $b \in A$, $c \in C$, the operation $f(x,b,c)$ defines a mapping from A into C. We prove that this mapping is one-to-one.

Indeed, if f is the group Mal'tsev operation $xy^{-1}z$, and $f(x,b,c) = f(y,b,c)$, then we have $xb^{-1}c = yb^{-1}c$ implies $x = y$. If f is the operation p of a quasigroup then

$$\begin{aligned}
(x \,/\, (b \setminus b)) \cdot (b \setminus c) &= (y \,/\, (b \setminus b)) \cdot (b \setminus c) \\
((x \,/\, (b \setminus b)) \cdot (b \setminus c)) \,/\, (b \setminus c) &= ((y \,/\, (b \setminus b)) \cdot (b \setminus c) \,/\, (b \setminus c) \\
x \,/\, (b \setminus b) &= y \,/\, (b \setminus b) \\
(x \,/\, (b \setminus b)) \cdot (b \setminus b) &= (y \,/\, (b \setminus b)) \cdot (b \setminus b) \\
x &= y.
\end{aligned}$$

For the operation q the argument is quite analogous.

Thus, $|A| \leq |C|$, and the converse inequality can be proved in the same way. □

As we shall prove later on in this section, every uniform Mal'tsev algebra is #-tractable. In particular, this holds for algebras obtained from groups and quasigroups.

Corollary 6 *The algebras $(G; m)$, $(Q; p)$, $(Q; q)$ are #-tractable, and therefore, any finite group and quasigroup is #-tractable.*

Notice that the decision constraint satisfaction problem arising from a finite group have been also studied in [22, 21].

Example 7 [Para-primal Algebras] See [49] for extensive results about para-primal algebras. In [15, 14] it is shown that the decision counterpart of #CSP($\mathbb{A}$) is solvable in polynomial time when $\mathbb{A}$ is a para-primal algebra. In this section we will show that para-primal algebras constitute an special case of uniform algebras, hence extending the results in [15, 14] to the counting problem.

Let A be a finite set and let ϱ be a n-ary relation on A, $(n \geq 1)$. Given a set $I = \{i_1, \ldots, i_k\}$, with $1 \leq i_1 < \cdots < i_k \leq n$, the *projection* $\mathrm{pr}_I \varrho$ of ϱ to I is defined to be the k-ary relation: $\{(\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_k}) : \mathbf{a} \in \varrho\}$. For every $1 \leq i \leq n$ we will denote $\mathrm{pr}_{\{i\}} \varrho$ by ϱ_i. A nonvoid subset I of $\{1, \ldots, k\}$ is said to be *ϱ-minimal* iff it is minimal in the sense that the projection $\varrho \to \mathrm{pr}_I \varrho$ is one-to-one.

Definition 10 *An algebra $\mathbb{A} = (A; F)$ is called* para-primal *iff it is finite and for every natural number n, every n-ary relation ϱ in* Inv F*, and every ϱ-minimal set I, we have* $\mathrm{pr}_I \varrho = \prod_{i \in I} \varrho_i$.

As a direct consequence of the definition of para-primal algebras we have that for every binary relaton $\varrho \subseteq A^2$ invariant under F, either: (1) $\varrho = \varrho_1 \times \varrho_2$ or (2) $\varrho = \{(a, f(a)) \mid a \in \varrho_1\}$ for some bijection $f : \varrho_1 \to \varrho_2$.

Let α be any congruence of a subalgebra on $\mathbb{A}$. Thus, α is a binary relation in Inv F and therefore it must be of one of the two types specified before. In consequence, the possibilities for α are very reduced. More precisely, α only can be a trivial congruence of a subalgebra of $\mathbb{A}$. In any case, all the classes of α have the same size, and hence para-primal algebras are uniform.

The existence of a Mal'tsev operation among the term operations of a para-primal algebra is a well-established fact (see for example [49],Theorem 4.15).

In consequence, we have

Corollary 7 *Para-primal algebras are uniform Mal'tsev algebras and hence, #-tractable.*

We conclude this section with a general result on the tractability of uniform algebras.

Theorem 14 *A uniform Mal'tsev algebra is #-tractable.*

Proof: In this proof we freely use some basic algebraic notions that can be found in [11, 39]. For algebraic notions and results concerning the CSP the reader is referred to [9, 7]. However, we introduce algebraic terminology that plays the key role in our proof with more details.

If η, θ are congruences of $\mathbb{A}$ such that $\eta \subseteq \theta$ then θ may join some of η-blocks, and induces a congruence on $\mathbb{A}/_{\eta}$. This congruence will be denoted by $\mathbb{A}/_{\theta}$.

In this proof we use, in fact, a generalized version of the constraint satisfaction problem, the multi-sorted constraint satisfaction problem, in which each variable has its own domain. As the multi-sorted CSP appears only as an auxiliary tool, we do not introduce it formally. Instead, we refer the reader to [8, 7] where all definitions are given accurately, and also some algebraic structure of the multi-sorted CSP is revealed. The only fact about the multi-sorted CSP used in this paper is that it involves multi-sorted relations, that is subsets of Cartesian products $A_1 \times \ldots \times A_n$ of domains.

Let $\mathbb{A} = (A; F)$ be a Mal'tsev algebra with a Mal'tsev operation d, $\mathcal{P} = (V; A; \mathcal{C}) \in \#\mathrm{CSP}(\mathbb{A})$, and $V = \{v_1, \ldots, v_n\}$. Let also $\mathcal{S}$ denote the set of all solutions to $\mathcal{P}$. We treat $\mathcal{S}$ as an n-ary relation over A. We make use of the result of [6] where a polynomial time algorithm has been exhibited finding a basis of $\mathcal{S}$. Theorem 14 will be proved by observing some properties of this basis, so we recall its detailed structure. Let $\mathbb{B}_i$ denote the set $\mathcal{S}_i = \{\mathbf{a}[v_i] \mid \mathbf{a} \in \mathcal{S}\}$. As is easily seen $\mathbb{B}_i$ is a subalgebra of $\mathbb{A}$. For congruences α, β we write $\alpha \prec \beta$ if $\alpha \subset \beta$ and, for no congruence γ, $\alpha \subset \gamma \subset \beta$. The identity relation and the total relation on an algebra $\mathbb{C}$ are denoted by $\Delta_{\mathbb{C}}$ and $\nabla_{\mathbb{C}}$ respectively.

Let ϱ be a multi-sorted relation, $\varrho \subseteq A_1 \times \ldots \times A_n$. The digraph $\mathcal{G}(\varrho) = (\underline{n}, E)$ is defined as follows:

$$(i,j) \in E \quad \text{iff} \quad \text{there is a mapping } \pi\colon A_i \to A_j \text{ such that for any } \mathbf{a} \in \varrho,\ \mathbf{a}[j] = \pi(\mathbf{a}[i]).$$

So, if i and j are connected by an edge, then the value of j-th entry of any tuple from ϱ is uniquely determined by the i-th entry.

For any i, choose a chain $\Delta_{\mathbb{B}_i} = \alpha_0^i \prec \alpha_1^i \prec \ldots \prec \alpha_{q_i}^i = \nabla_{\mathbb{B}_i}$ of congruences of $\mathbb{B}_i$, and consider the direct product $\mathbb{B} = \mathbb{B}_1 \times \ldots \times \mathbb{B}_n$. For any sequence $(\beta^1, \ldots, \beta^n)$ where $\beta^i \in \{\alpha_0^i, \ldots, \alpha_{q_i}^i\}$, the congruence $\overline{\beta}$ of $\mathbb{B}$ is defined as follows: $\langle(\mathbf{a}[1], \ldots, \mathbf{a}[n]), (\mathbf{b}[1], \ldots, \mathbf{b}[n])\rangle \in \overline{\beta}$ if and only if $(\mathbf{a}[i], \mathbf{b}[i]) \in \beta^i$, for all $i \in \underline{n}$. Notice that $\mathbb{B}/\overline{\beta}$ can be viewed as a subset of the direct product $\mathbb{B}_1/_{\beta^1} \times \ldots \times \mathbb{B}_n/_{\beta^n}$.

There is a chain $\overline{\beta}_0 \subset \overline{\beta}_1 \subset \ldots \subset \overline{\beta}_k$ of congruences of $\mathbb{B}$ such that $\overline{\beta}_l$ corresponds to a sequence $(\beta_{l1}, \ldots, \beta_{ln})$, $\beta_{li} \in \{\alpha_0^i, \ldots, \alpha_{q_i}^i\}^n$, and

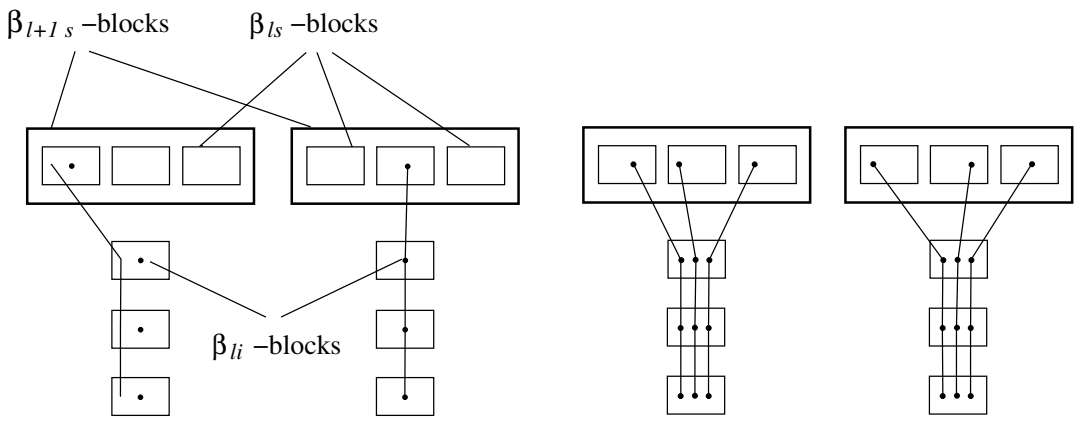


Figure 7: The structure of the part B_l of the basis B. Boxes represent congruence classes, lines represents tuples from the basis, and dots entries of the tuples.

- $\beta_{k1} = \alpha^1_{q_1} = \nabla_{\mathbb{B}_1} \dots \beta_{kn} = \alpha^n_{q_n} = \nabla_{\mathbb{B}_n}$, $\beta_{01} = \alpha^1_0 = \Delta_{\mathbb{B}_1} \dots \beta_{0n} = \alpha^n_0 = \Delta_{\mathbb{B}_n}$;
- for any $l \in \{0, \dots, k-1\}$ and $i \in \{1, \dots, n\}$, either $\beta_{li} = \beta_{l+1i}$ or $\beta_{li} \prec \beta_{l+1i}$;
- the set $I_l = \{i \in \underline{n} : \beta_{li} \prec \beta_{l+1i}\}$ is a strongly connected component of $\mathcal{G}(\mathcal{S}/\overline{\beta_l})$.

Notice that (v_i, v_j) is an edge of $\mathcal{G}(\mathcal{S}/\overline{\beta_l})$ if and only if for any $\mathbf{a}_1, \mathbf{a}_2 \in \mathcal{S}$, if $(\mathbf{a}_1(v_i), \mathbf{a}_2(v_i)) \in \beta_{li}$ then $(\mathbf{a}_1(v_j), \mathbf{a}_2(v_j)) \in \beta_{lj}$

Then the basis consists of k parts, $B = B^k \cup \ldots \cup B^1$ where

- $B^k = \{\mathbf{a}\}$ where $\mathbf{a}$ is an arbitrary tuple from $\mathcal{S}$;
- $B^l = B^l_1 \cup \ldots \cup B^l_{r_l}$ where r_l is the number of classes of the congruence β_{l+1s}/β_{ls}, for certain $s \in I_l$.
- Let $C^l_1, \dots, C^l_{r_l}$ be β_{l+1s}/β_{ls}-classes. Then one of the following two possibilities holds:
  (a) for any $1 \le j \le r_l$, $B^l_j = \{\mathbf{a}\}$ with $\mathbf{a}[s]/\beta_{ls} \in C_j$;
  (b) for any $1 \le j \le r_l$,
  $$B^l_j = \{\mathbf{a}_1, \dots, \mathbf{a}_t\}$$
  where $t = |C^l_j|$, $\mathbf{a}_1[i], \dots, \mathbf{a}_t[i]$ lie in the same β_{li}-class for $i \notin I_l$, and for any $b \in C^l_j$ there is $u \in \underline{t}$ with $\mathbf{a}_u[s]/\beta_{ls} = b$, see Fig. 7.

In [6], it is shown that an arbitrary tuple $\mathbf{a} \in \mathcal{S}$ can be represented in the form

$$d(d(\dots d(d(\dots d(\mathbf{a}_k, \mathbf{a}_{k-1}, \mathbf{b}_{k-1}), \dots), \mathbf{a}_l, \mathbf{b}_l), \dots), \mathbf{a}_1, \mathbf{b}_1)$$

where, $\{\mathbf{a}_k\} = B^k$, for each $1 \leq l < k$, if $\mathbf{c}_{l+1}$ denotes the tuple

$$d(d(\ldots d(\mathbf{a}_k, \mathbf{a}_{k-1}, \mathbf{b}_{k-1}), \ldots), \mathbf{a}_{l+1}, \mathbf{b}_{l+1})$$

then $(\mathbf{c}_{l+1}, \mathbf{a}) \in \overline{\beta}_{l+1}$, and $\mathbf{a}_l, \mathbf{b}_l \in B^l$ are such that $(\mathbf{a}_l[s], \mathbf{c}_{l+1}[s]) \in \beta_{ls}$, $(\mathbf{b}_l[s], \mathbf{a}[s]) \in \beta_{ls}$ for $s \in I_l$. Moreover, such a representation is unique.

Example 8 Let S_3 denote the full symmetric group on the 3-element set $\{1, 2, 3\}$. The elements of S_3 are

$$\varepsilon = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \quad \beta_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \quad \beta_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix},$$
$$\gamma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \quad \gamma_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \quad \gamma_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

The group S_3 is a Mal'tsev algebra with a Mal'tsev operation $m(x, y, z) = xy^{-1}z$. We consider the algebra $\mathbb{A} = (S_3; \{m\})$. As is easily seen, the only proper congruence α of $\mathbb{A}$ and S_3 has the classes $\{\varepsilon, \beta_1, \beta_2\}, \{\gamma_1, \gamma_2, \gamma_3\}$, and the corresponding normal subgroup of S_3 is $\{\varepsilon, \beta_1, \beta_2\}$.

A basis of a subalgebra

$$\mathcal{S} = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \beta_1 & \beta_1 & \beta_1 & \beta_2 & \beta_2 & \beta_2 & \gamma_1 & \gamma_1 & \gamma_1 & \gamma_2 & \gamma_2 & \gamma_2 & \gamma_3 & \gamma_3 & \gamma_3 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_1 & \gamma_2 & \gamma_3 & \varepsilon & \beta_1 & \beta_2 & \varepsilon & \beta_1 & \beta_2 & \varepsilon & \beta_1 & \beta_2 \\ \beta_2 & \beta_2 & \beta_2 & \varepsilon & \varepsilon & \varepsilon & \beta_1 & \beta_1 & \beta_1 & \gamma_2 & \gamma_2 & \gamma_2 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_1 & \gamma_1 & \gamma_1 \end{pmatrix}$$

of $\mathbb{A}^3$ can be chosen as follows

$$\underbrace{\begin{matrix} \varepsilon \\ \gamma_3 \\ \beta_2 \end{matrix}}_{B^5} \underbrace{\left|\begin{matrix} \beta_1 & \gamma_1 \\ \gamma_1 & \varepsilon \\ \varepsilon & \gamma_2 \end{matrix}\right.}_{B^4} \underbrace{\left|\begin{matrix} \beta_1 & \beta_2 & \varepsilon \\ \gamma_1 & \gamma_1 & \gamma_2 \\ \varepsilon & \beta_1 & \beta_2 \end{matrix}\right.}_{B_1^3} \underbrace{\begin{matrix} \gamma_3 & \gamma_1 & \gamma_2 \\ \beta_1 & \beta_2 & \beta_2 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{matrix}}_{B_2^3} \underbrace{\left|\begin{matrix} \beta_1 & \beta_1 & \beta_1 \\ \gamma_1 & \gamma_2 & \gamma_3 \\ \varepsilon & \varepsilon & \varepsilon \end{matrix}\right.}_{B_1^2} \underbrace{\begin{matrix} \gamma_3 & \gamma_3 & \gamma_3 \\ \varepsilon & \beta_1 & \beta_2 \\ \gamma_1 & \gamma_1 & \gamma_1 \end{matrix}}_{B_2^2} \underbrace{\left|\begin{matrix} \beta_1 \\ \gamma_1 \\ \varepsilon \end{matrix}\right.}_{B_1^1} \underbrace{\begin{matrix} \gamma_3 \\ \varepsilon \\ \gamma_1 \end{matrix}}_{B_2^1}$$

The strongly connected components are $I_1 = \{1\}, I_2 = \{2\}, I_3 = \{3\}, I_4 = \{1, 2, 3\}$, and the chain of congruences is $(\Delta, \Delta, \Delta) \subset (\alpha, \Delta, \Delta) \subset (\alpha, \alpha, \Delta) \subset (\alpha, \alpha, \alpha) \subset (\bigtriangledown, \bigtriangledown, \bigtriangledown)$.

Theorem 15 ([6]) *Let $\mathbb{A}$ be a finite Mal'tsev algebra. Then a basis of the solution space to any problem instance $\mathcal{P} \in \#\mathrm{CSP}(\mathbb{A})$ can be found in polynomial time, and each solution to $\mathcal{P}$ is uniquely decomposable.*

If the Mal'tsev algebra $\mathbb{A}$ is uniform then Theorem 15 may assists in finding the number of solutions. We now in a position to prove Theorem 14.

Let $B = B^k \cup \ldots \cup B^1$ be a basis of the set of solutions to $\mathcal{P}$; and let N_l denote the number of different $(B^k \cup \ldots \cup B^l)$-decompositions. By Theorem 15, a B-decomposition of a solution is unique, therefore, N_1 equals the number of solutions.

Obviously, $N_k = 1$. Then, if N_{l+1} is already known, notice that any $(B^k \cup \ldots \cup B^{l+1})$-decomposition has a unique extension if each B_j^l is a singleton, and has $|C_j^l|$ extensions

otherwise. Since all the classes C_j^l are of the same size, we have $N_l = N_{l+1}$ in the first case, and $N_l = N_{l+1} \cdot |C_j^l|$ in the second case. □

For constraint languages we have

Corollary 8 *A uniform constraint language invariant with respect to a Mal'tsev operation is #-tractable.*

6 Conclusion

We have shown that the algebraic approach developed originally to deal with the decision constraint satisfaction problem is applicable, in some aspects even more efficiently, to the counting constraint satisfaction problem. On the one hand, this has allowed us, by making use of advanced and deep algebraic results (Theorem 10), to obtain a strong necessary condition on tractable cases of the counting constraint satisfaction problem (Theorem 5), and to pose a plausible conjecture about a possible form of a criterion for such cases. On the other hand, earlier results on algorithmic and structural aspects of Mal'tsev algebras make it possible to design polynomial time algorithms solving many of the counting constraint satisfaction problems satisfying the criterion. In particular, the criterion is true for problems over small domains.

The algebraic approach also appears to be very helpful in systematization of existing complexity results on #CSP by identifying a common property of classes solvable in polynomial time, and providing a strong guidance for further research.

The applicability of the algorithm solving Mal'tsev decision constraint satisfaction problems to #CSP is limited. However, as Theorem 11 shows, those limitations can be overcome by using structural properties of relations invariant under a Mal'tsev operation. We strongly believe that future developments in algebraic theory related to Mal'tsev operations will lead to a complete solution of Problem 1.

References

[1] J. Bang-Jensen and P. Hell. The effect of two cycles on the complexity of colourings by directed graphs. *Discrete Applied Math.*, 26(1):1–23, 1990.

[2] J. Bang-Jensen, P. Hell, and G. MacGillivray. Hereditarily hard H-coloring problems. *Discrete Math.*, 138:75–92, 1995.

[3] G.R. Brightwell and P. Winkler. Graph homomorphisms and phase transitions. *Journal of Combinatorial Theory, Ser. B*, 77:221–262, 1999.

[4] R. Bubley, M. Dyer, C. Greenhill, and M. Jerrum. On approximately counting colourings of small degree graphs. *SIAM Journal of Computing*, 29:387–400, 1999.

[5] A. Bulatov. Three-element Mal'tsev algebras. *Acta Sci. Math (Szeged).* To appear.

[6] A.A. Bulatov. Mal'tsev constraints are tractable. Technical Report PRG-RR-02-05, Computing Laboratory, University of Oxford, Oxford, UK, 2002.

[7] A.A. Bulatov and P.G. Jeavons. Algebraic approach to multi-sorted constraints. Technical Report PRG-RR-01-18, Computing Laboratory, University of Oxford, Oxford, UK, 2001.

[8] A.A. Bulatov and P.G. Jeavons. Algebraic structures in combinatorial problems. Technical Report MATH-AL-4-2001, Technische universität Dresden, Dresden, Germany, 2001.

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

[10] R. Burton and J. Steif. Nonuniqueness of measures of maximal entropy for subshifts of finite type. *Ergodic Theory and Dynamical Systems*, 14:213–236, 1994.

[11] P.M. Cohn. *Universal Algebra*. Harper & Row, 1965.

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

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

[14] V. Dalmau. A new tractable class of constraint satisfaction problems. In *Proceedings 6th International Symposium on Artificial Intelligence and Mathematics*, 2000.

[15] V. Dalmau. *Computational Complexity of Problems over Generalised Formulas*. PhD thesis, Department LSI of the Universitat Politecnica de Catalunya (UPC), Barcelona., March, 2000.

[16] J. Diaz, M. Serna, and D.M. Thilikos. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, DIMACS/DIMATIA Workshop on Graphs, Morphism and Statistical Physics. American Mathematical Society*, chapter Recent results on parameterized H-coloring. To appear.

[17] J. Diaz, M. Serna, and D.M. Thilikos. The complexity of restrictive H-coloring. *To appear in Discrete Applied Mathematics*.

[18] J. Diaz, M. Serna, and D.M. Thilikos. Counting list H-colorings and variants. Technical Report LSI-01-27-R, Departament LSI, Universitat Politècnica de Catalunya, 2001.

[19] J. Diaz, M. Serna, and D.M. Thilikos. Counting h-colorings of partial k-trees. *Theoretical Computer Science*, 281:291–309, 2002.

[20] M. Dyer and C. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17:260–289, 2000.

[21] T. Feder. Constraint satisfaction on finite groups with near subgroups. Submitted to Journal of Algebra.

[22] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal of Computing*, 28:57–104, 1998.

[23] R. Freese and R. McKenzie. *Commutator theory for congruence modular varieties*, volume 125 of *London Math. Soc. Lecture Notes*. London, 1987.

[24] M. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA., 1979.

[25] C. Greenhill. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. *Computational Complexity*, 9:52–73, 2000.

[26] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the 33rd Annual ACM Simposium on Theory of Computing*, pages 657–666, Hersonissos, Crete, Greece, July 2001. ACM Press.

[27] P. Hell and J. Nešetřil. Counting list homomorphisms for graphs with bounded degrees. *Discrete Mathematics*. to appear.

[28] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Ser.B*, 48:92–110, 1990.

[29] P. Hell, J. Nešetřil, and X. Zhu. Duality and polynomial testing of tree homomorphisms. *Trans. of the AMS*, 348(4):1281–1297, 1996.

[30] D. Hobby and R.N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.

[31] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, and R.E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27:1142–1167, 1998.

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

[33] P.G. Jeavons, D.A. Cohen, and M.C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

[34] P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

[35] Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61:302–332, 2000.

[36] J.L. Lebowitz and G. Gallavotti. Phase transitions in binary lattice gases. *Journal of Math. Physics*, 12:1129–1133, 1971.

[37] L.A. Levin. Universal enumeration problems. *Problems on Information Transmission*, 9:265–266, 1973.

[38] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):331–335, 1986.

[39] R.N. McKenzie, G.F. McNulty, and W.F. Taylor. *Algebras, Lattices and Varieties*, volume I. Wadsworth and Brooks, California, 1987.

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

[41] P. Orponen. Dempster's rule of combination is #-complete. *Artificial Intelligence*, 44:245–253, 1990.

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

[43] R. Pöschel. Galois connection for operations and relations. Technical Report MATH-AL-8-2001, Technische Universität Dresden, Germany, 2001.

[44] R. Pöschel and L.A. Kalužnin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.

[45] E.L. Post. *The two-valued iterative systems of mathematical logic*, volume 5 of *Annals Mathematical Studies*. Princeton University Press, 1941.

[46] J.S. Provan and M.O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[47] D. Roth. On the hardness of approximate reasonning. *Artificial Intelligence*, 82:273–302, 1996.

[48] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226, 1978.

[49] A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathematiques Superieures*. Université de Móntreal, 1986.

[50] L. Valiant. The complexity of computing the permanent. *Theoretical Computing Science*, 8:189–201, 1979.

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

[52] X. Zhu. A polynomial algorithm for homomorphisms to oriented cycles. *Journal of Algorithms*, 19:333–345, 1995.