SIAM J. COMPUT.
Vol. 45, No. 6, pp. 2177–2198

NONNEGATIVE WEIGHTED #CSP: AN EFFECTIVE COMPLEXITY DICHOTOMY*

JIN-YI CAI†, XI CHEN‡, AND PINYAN LU§

Abstract. We prove a complexity dichotomy theorem for counting constraint satisfaction problems (#CSPs) with nonnegative and algebraic weights. This caps a long series of important results on counting problems including counting unweighted and weighted graph homomorphisms and the celebrated dichotomy theorem for unweighted #CSPs. Our dichotomy theorem gives a succinct criterion for tractability. If a set $\mathcal{F}$ of constraint functions satisfies this criterion, then the problem #CSP($\mathcal{F}$) defined by $\mathcal{F}$ is solvable in polynomial time; if $\mathcal{F}$ does not satisfy this criterion, then the problem is #P-hard. Furthermore, we show that the question of whether a given $\mathcal{F}$ satisfies the criterion or not is decidable in NP. Surprisingly, our tractability criterion is simpler than the previous criteria for the more restricted classes of counting problems, although when specialized to those classes, they are logically equivalent. Our proof mainly uses linear algebra and represents a departure from universal algebra, the dominant methodology in recent years for the study of #CSPs on large domains.

Key words. constraint satisfaction problem, counting problems, complexity dichotomy

AMS subject classifications. 68Q25, 68Q20

DOI. 10.1137/15M1032314

1. Introduction.

The investigation of constraint satisfaction problems (CSPs) has been one of the most active research areas in which significant progress has been made in recent years. The study of CSPs includes at least the following major branches: decision problems—determining whether a solution exists [42, 32, 6, 37]; optimization problems—finding a solution that satisfies the most constraints (or in the weighted case, achieving the highest total weight) [31, 36, 1, 24, 40, 44, 41]; and counting problems—computing the number of solutions (or the partition function in the weighted case) [8, 7, 10, 3, 27]. The decision CSP dichotomy conjecture of Feder and Vardi [28], that every decision CSP defined by a constraint language is either in P or NP-complete, remains open. Much work has been devoted to the optimization version of CSP, constituting a significant fraction of ongoing activities in approximation algorithms.

The subject of this paper is on counting CSPs (#CSPs), more precisely, on *weighted* #CSPs. For *unweighted* #CSPs, the problem is stated as follows: D is a fixed finite set called the domain set; $\Gamma = \{\Theta_1, \ldots, \Theta_h\}$ is a fixed finite set of constraint predicates, where each Θ_i is a relation on D^{r_i} of some finite arity $r_i \geq 1$. An instance of #CSP(Γ) consists of a finite set of n variables, each ranging over D, and a finite sequence of constraints from Γ, each applied to a sequence of these variables.

*Received by the editors July 24, 2015; accepted for publication (in revised form) August 23, 2016; published electronically December 20, 2016. A preliminary version of the paper appeared in *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011 [14].
http://www.siam.org/journals/sicomp/45-6/M103231.html
Funding: The first author was supported by NSF grants CCF-0914969 and CCF-1217549. The second author was supported by NSF grants CCF-1149257 and CCF-1423100.
†Department of Computer Sciences, University of Wisconsin, Madison, WI 53706 (jyc@cs.wisc.edu).
‡Computer Science Department, Columbia University, New York, NY 10027 (xichen@cs.columbia.edu).
§Institute for Theoretical Computer Science, Shanghai University of Finance and Economics, Shanghai 200433, China (lu.pinyan@mail.shufe.edu.cn).

It defines a new n-ary relation $R \subseteq D^n$, where an assignment $(x_1, \ldots, x_n) \in D^n$ of the variables is in R if and only if all the constraints are satisfied. The #CSP asks for the size of R.

In a (nonnegatively) weighted #CSP, the set Γ is replaced by a fixed finite set of constraint functions $\mathcal{F} = \{f_1, \ldots, f_h\}$ in which each f_i maps D^{r_i} to *nonnegative* and *algebraic* reals $\mathbb{R}_+$.[1] An instance of #CSP($\mathcal{F}$) similarly consists of n variables, ranging over D, as well as a finite sequence of constraint functions from $\mathcal{F}$, each applied over a sequence of these variables. It then defines an n-ary function F: for each assignment $(x_1, \ldots, x_n) \in D^n$ of the variables, $F(x_1, \ldots, x_n)$ is the product of the constraint function evaluations. The output is the so-called *partition function*, that is, the sum of F over all $(x_1, \ldots, x_n) \in D^n$. The unweighted #CSP is the special case where all functions in $\mathcal{F}$ are $\{0, 1\}$-valued. (Formal definitions can be found in section 2.)

Regarding unweighted #CSPs, Bulatov [7] proved a sweeping dichotomy theorem. He introduced a criterion called *congruence singularity* and showed that for any finite set Γ of predicates over any finite domain D, if Γ satisfies this criterion, then #CSP(Γ) is solvable in P; otherwise it is #P-complete. His proof uses deep structural theorems from universal algebra [11, 33, 29]. Indeed this approach using universal algebra has been one of the most exciting developments in the study of CSPs in recent years—first used in decision CSPs [34, 35, 6, 5]—and has been called the *algebraic approach*.

However, this is not the *only* approach. Later in [27] Dyer and Richerby obtained an alternative proof of the dichotomy theorem for unweighted #CSPs. Their proof is considerably more direct and uses no universal algebra other than the notion of a Mal'tsev polymorphism. Furthermore, they showed that the dichotomy is decidable in NP [27]. By treating rational weights as integral multiples of a common denominator, the dichotomy can also be extended to include *nonnegative rational* weights [3].

In this paper we present a complexity dichotomy theorem for all weighted #CSPs with *nonnegative* and *algebraic* weights. To describe our approach, we briefly review the proofs by Bulatov and by Dyer and Richerby for unweighted #CSP(Γ). Bulatov's proof is deeply embedded in a structural theory of universal algebra called *tame congruence theory* [33]. A congruence here is an equivalence relation expressible in a given universal algebra. The starting point of this *algebraic approach* is the realization of a close connection between unweighted #CSP(Γ) and the *relational clone* $\langle\Gamma\rangle$ generated by Γ. $\langle\Gamma\rangle$ is the closure set of all relations expressible from Γ by the Boolean conjunction $\wedge$ and the existential quantifier $\exists$. A basic property called *congruence permutability* is shown to be a necessary condition for the tractability of #CSP(Γ) [9, 8, 10]. It is also known from universal algebra that congruence permutability is equivalent to the existence of *Mal'tsev polymorphisms*, which is then equivalent to the more combinatorial condition of *strong rectangularity* of Dyer and Richerby [27]. Recall that Γ is *strongly rectangular* if for every n-ary relation R defined by an instance of #CSP(Γ) and every pair of positive integers k and ℓ with $k + \ell \le n$, the following $|D|^k \times |D|^\ell$ $\{0, 1\}$-matrix $\mathbf{M}$ is *block-diagonal* after separately permuting its rows and columns: The rows of $\mathbf{M}$ are indexed by tuples $\mathbf{u} \in D^k$, the columns are indexed by $\mathbf{v} \in D^\ell$, and $M(\mathbf{u}, \mathbf{v}) = 1$ if there exists a tuple $\mathbf{w} \in D^{n-k-\ell}$ such that their concatenation $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ is in R; otherwise, we have $M(\mathbf{u}, \mathbf{v}) = 0$. (See the formal definition in section 2.)

[1] For convenience, we use $\mathbb{R}$ to denote the set of algebraic real numbers and $\mathbb{R}_+$ to denote the set of nonnegative and algebraic numbers (though most of our definitions and lemmas apply to general real numbers when computation is not involved).

Assuming Γ satisfies this necessary condition (otherwise #CSP(Γ) is already #P-hard), Bulatov's proof delves much more deeply than Mal'tsev polymorphisms and uses many more results and techniques from universal algebra. The Dyer–Richerby proof, on the other hand, manages to avoid much of universal algebra. They went on to give a more combinatorial criterion, called *strong balance*: For every n-ary relation R defined by an instance of #CSP(Γ) and every triple of integers $k, \ell \geq 1$ and $t \geq 0$, with $k+\ell+t \leq n$, the following $|D|^k \times |D|^\ell$ *integer* matrix $\mathbf{M}$ must be block-diagonal, and *all of its blocks are of rank* 1 (which we will refer to as a *block-rank*-1 matrix):

$$M(\mathbf{u},\mathbf{v}) = \Big|\big\{\mathbf{w} \in D^t : \exists\, \mathbf{z} \in D^{n-k-\ell-t} \text{ such that } (\mathbf{u},\mathbf{v},\mathbf{w},\mathbf{z}) \in R\big\}\Big|. \tag{1.1}$$

(See the formal definition in section 9.) Dyer and Richerby showed in [27] that strong balance (which implies strong rectangularity) is a criterion that leads to a dichotomy theorem for the complexity of #CSP(Γ). They indeed proved that it is equivalent to Bulatov's congruence singularity which is stated in the language of universal algebra.

The first difficulty we encountered when trying to extend the unweighted dichotomy to weighted #CSP($\mathcal{F}$) with nonnegative and algebraic weights is that there is no direct extension of the notion of strong balance in the weighted setting. Although the number of $\mathbf{w}$ that satisfies R on the right side of (1.1) can be naturally replaced by the sum of F (the function defined by a #CSP($\mathcal{F}$) instance) over $\mathbf{w}$, there seems to be no easy way to introduce existential quantifiers to the more general weighted setting. Moreover, the use of existential quantifiers in the notion of strong balance is crucial to the proof of Dyer and Richerby [27]: Their polynomial-time counting algorithm for #CSP(Γ) with a strongly balanced Γ heavily relies on them.

A key observation that allows us to overcome this difficulty is that the notion of strong balance is equivalent to the notion of balance *without using any existential quantifiers* (that is, we only consider partitions of the variables into three parts with no $\mathbf{z}$). We include the proof of this equivalence in section 9, which inspires us to use the following seemingly weaker notion of *balance* for weighted #CSP($\mathcal{F}$), with no existential quantifiers at all: For any n-ary function F defined by a #CSP($\mathcal{F}$) instance and every pair of $k, \ell \geq 1$ with $k+\ell \leq n$, the following $|D|^k \times |D|^\ell$ matrix $\mathbf{M}$ must be *block-rank*-1:

$$M(\mathbf{u},\mathbf{v}) = \sum_{\mathbf{w}\in D^{n-k-\ell}} F(\mathbf{u},\mathbf{v},\mathbf{w}) \quad \text{for all } \mathbf{u} \in D^k \text{ and } \mathbf{v} \in D^\ell.$$

It is not very difficult to show that #CSP($\mathcal{F}$) is #P-hard when $\mathcal{F}$ is not balanced. But is the condition of $\mathcal{F}$ being balanced *sufficient for the tractability* of #CSP($\mathcal{F}$)? We show that this is the case by obtaining a polynomial-time algorithm for #CSP($\mathcal{F}$) when $\mathcal{F}$ is balanced. Our algorithm works differently from that of Dyer and Richerby, in that it avoids the use of existential quantifiers and is designed specially for weighted #CSP($\mathcal{F}$) with a balanced $\mathcal{F}$. As a result, we get the following dichotomy for weighted #CSP with a logically simpler tractability criterion.

THEOREM 1.1. *For any language $\mathcal{F}$ with nonnegative and algebraic weights, the problem #CSP($\mathcal{F}$) is in polynomial time if $\mathcal{F}$ is balanced, and is #P-hard otherwise.*

Theorem 1.1 follows as a corollary of an alternative dichotomy theorem for #CSP with nonnegative and algebraic weights. Given $\mathcal{F} = \{f_1, \ldots, f_h\}$, we define its corresponding unweighted constraint language as $\Gamma = \{\Theta_1, \ldots, \Theta_h\}$, in which $\mathbf{x} \in \Theta_i$ if and only if $f_i(\mathbf{x}) > 0$. We introduce in section 3 a criterion called *weak balance*, which by name is (similar to but) weaker than that of balance, and prove the following dichotomy theorem from which Theorem 1.1 follows (see section 3).

THEOREM 1.2. *For any language $\mathcal{F}$ with nonnegative and algebraic weights, the problem #CSP($\mathcal{F}$) is in polynomial time if its corresponding unweighted language Γ is strongly rectangular and $\mathcal{F}$ is weakly balanced, and is #P-hard otherwise.*

While both theorems are complexity dichotomies of #CSPs with nonnegative and algebraic weights, and thus, the tractability criteria must be equivalent, *assuming that* #P $\neq$ FP, at this moment we are not aware of an unconditional proof of their equivalence. The language $\mathcal{F}$ being balanced implies trivially both the strong rectangularity of Γ and the weak balance of $\mathcal{F}$, but the converse direction remains an open problem.

A new ingredient in the algorithmic part of Theorem 1.2 is the concept of *a vector representation* for a nonnegative function F. Given a nonnegative function F over n variables, we say $s_1, \ldots, s_n : D \to \mathbb{R}_+$ is a vector representation of F if

$$F(x_1, \ldots, x_n) = s_1(x_1) \times \cdots \times s_n(x_n)$$

for all $(x_1, \ldots, x_n) \in D^n$ with $F(x_1, \ldots, x_n) > 0$.

The first step of our algorithm is to show that, given any instance of #CSP($\mathcal{F}$), where $\mathcal{F}$ satisfies conditions of Theorem 1.2, we can compute a vector representation of the function F it defines in polynomial time. However, F in general may have a lot of "holes" $\mathbf{x}$ where $s_1(x_1), \ldots, s_n(x_n) > 0$ but $F(x_1, \ldots, x_n) = 0$, so it is still not clear how to do the sum of F over $x_1, \ldots, x_n \in D$.

The next step is quite a surprise. When $\mathcal{F}$ satisfies conditions of Theorem 1.2 we show how to compute efficiently a sequence of functions $t_2, \ldots, t_n : D \to \mathbb{R}_+$ in polynomial time such that for any $(u_1, \ldots, u_n) \in D^n$ with $F(u_1, \ldots, u_n) > 0$,

$$\sum_{x_2,\ldots,x_n \in D} F(u_1, x_2, \ldots, x_n) = s_1(u_1) \cdot \prod_{j=2}^{n} \frac{s_j(u_j)}{t_j(u_j)}. \tag{1.2}$$

The intriguing part of (1.2) is that its left side depends only on u_1, but (1.2) holds for any tuple $(u_1, \ldots, u_n) \in D^n$ as long as $F(u_1, \ldots, u_n) > 0$. A crucial ingredient we use in computing $t_2, \ldots, t_n$ and proving (1.2) is the *succinct* data structure called *a frame* introduced by Dyer and Richerby for unweighted #CSP [27] (which is similar to the "compact representation" of Bulatov and Dalmau [2]). Once we obtain $t_2, \ldots, t_n$ and (1.2), computing the partition function becomes trivial.

After obtaining Theorem 1.2 (and using it to prove Theorem 1.1), we also show in section 6 that the tractability criterion of Theorem 1.2, i.e.,

whether Γ is strongly rectangular and $\mathcal{F}$ is weakly balanced,

is decidable in NP. The proof follows the approach of Dyer and Richerby [27] for unweighted #CSP, with new ideas and constructions developed for the weighted setting. It is worth pointing out that the decidability proof takes great advantage of the weaker notion of weak balance, which is the reason we introduce it and include Theorem 1.2 (instead of proving directly the much cleaner Theorem 1.1, based on the notion of balance). Given the lack of an unconditional proof of their equivalence, it remains an open problem whether the criterion of balance is decidable in NP as well.

This advance, from unweighted to nonnegatively weighted #CSP, is akin to the leap from the Dyer–Greenhill result on counting 0-1 graph homomorphisms [26] to the Bulatov–Grohe result for the nonnegative case [4]. The Bulatov–Grohe result paved the way for all future developments. This is because their dichotomy theorem not only is intrinsically important and sweeping but also gives an elegant tractability criterion, which allows many of its easy applications. Almost all future results in this area use the Bulatov–Grohe criterion. Our dichotomy in this paper covers nonnegatively weighted #CSP, which achieves a similar leap from the 0-1 case of Bulatov and Dyer and Richerby and, in the meanwhile, simplifies their tractability criteria. Therefore, it is hoped that it will also be useful for future research.

In hindsight, perhaps one may re-evaluate the *algebraic approach*. We now know that there is another *algebraic approach*, based mainly on matrix algebra rather than (relational) universal algebra, which gives us a more direct and complete dichotomy theorem for #CSP. It is perhaps also a case where the proper generalization, namely weighted #CSP, leads to a simpler resolution of the problem than the original form of unweighted #CSP.

Several special cases of weighted #CSP have been studied intensively in the literature. In particular, counting graph homomorphisms can be viewed as a special case where $\mathcal{F}$ contains a single binary constraint function. There have been great advances made on graph homomorphisms [26, 4, 25, 12], and our dichotomy theorem generalizes all previous dichotomy theorems where the constraint function is nonnegative.

Looking beyond nonnegatively weighted counting problems, in graph homomorphisms, great progress [30, 15, 43] has already been made. The success has also been extended to #CSPs recently in [13], which gives a complexity dichotomy theorem for all #CSPs with complex and algebraic weights, after a preliminary version of the current paper appeared as [14] in 2011. Compared to [13], the dichotomy of this paper is weaker as it only covers #CSPs with nonnegative weights. However, we believe that the approach and techniques of our dichotomy in the current paper are still of interest because the tractability criterion is conceptually much simpler than that of [13]; for the latter, it remains an open problem whether its tractability criterion is decidable or not. Ideas behind the NP decidability proof of the current paper as well as that of [27] may help in this direction.

Going beyond #CSP-type problems, holographic algorithms and reductions are aimed precisely at counting problems where cancellation is the main feature [45]. Works on Holant problems as well as their dichotomy theorems are the beginning steps in that direction [20, 21, 19]. Holant problems can be thought of as sum-of-product computations on graphs where edges are variables and vertices are constraints. A representative example is counting Perfect Matchings. #CSPs are Holant problems where the set of Equality constraint functions (of all arities) is implicitly present, namely, #CSP($\mathcal{F}$) is the same as Holant($\mathcal{F} \cup \mathcal{EQ}$), where $\mathcal{EQ} = \{=_k \mid k \geq 1\}$. Consequently, a complexity dichotomy for Holant problems would imply a corresponding dichotomy for #CSPs. Some progress in classifying Holant problems has been made, but most of the results are for constraint functions over the Boolean domain [22, 17, 16]. For the general domain D, there are only a few dichotomies for Holant problems where the constraint functions are required to be quite restricted [23, 18]. While these Holant dichotomies apply to complex-valued constraint functions, they are incomparable with results of the current paper because the Holant problems are more restrictive than #CSPs.

2. Preliminaries.

For $m \geq 1$, we write $[m]$ to denote $\{1, \ldots, m\}$. We start with some definitions about nonnegative matrices. We say a nonnegative $m \times n$ matrix $\mathbf{M}$ is *rectangular* if one can permute its rows and columns separately so that it becomes a block-diagonal matrix. Equivalently, $\mathbf{M}$ is rectangular if there exist s pairwise disjoint and nonempty subsets $A_1, \ldots, A_s$ of $[m]$, and s pairwise disjoint and nonempty subsets $B_1, \ldots, B_s$ of $[n]$, for some $s \geq 0$, such that for all $i \in [m]$, $j \in [n]$,

$$M(i,j) > 0 \iff i \in A_k \text{ and } j \in B_k \text{ for some } k \in [s].$$

Given a nonnegative and rectangular matrix $\mathbf{M}$ with s blocks $A_1 \times B_1, \dots, A_s \times B_s$, we say it is *block-rank*-1 if, in addition, each $A_k \times B_k$ submatrix of $\mathbf{M}$ is of rank 1.

The two lemmas below follow directly from the definitions in the previous paragraph.

LEMMA 2.1. *Let* $\mathbf{M}$ *be a block-rank-*1 *matrix with* $s \geq 1$ *blocks:* $A_1 \times B_1, \dots, A_s \times B_s$*. If* $i^* \in A_k$ *and* $j^* \in B_k$ *for some* $k \in [s]$*, then for any* $i \in A_k$ *we have*

$$\frac{\sum_{j \in B_k} M(i,j)}{\sum_{j \in B_k} M(i^*,j)} = \frac{M(i,j^*)}{M(i^*,j^*)}.$$

LEMMA 2.2. *Let* $\mathbf{M}$ *be an* $m \times n$ *nonnegative matrix. If* $\mathbf{M}$ *is not block-rank-*1*, then* $\mathbf{M}\mathbf{M}^T$ (*which is a symmetric, nonnegative* $m \times m$ *matrix*) *is not block-rank-*1*.*

Proof. As $\mathbf{M}$ is not block-rank-1, it must have two rows that are neither linearly dependent nor orthogonal. Let $\mathbf{M}(i,*)$ and $\mathbf{M}(j,*)$ be such rows, $i, j \in [m]$. Then

$$0 < \big\langle \mathbf{M}(i,*), \mathbf{M}(j,*) \big\rangle^2 < \big\langle \mathbf{M}(i,*), \mathbf{M}(i,*) \big\rangle \cdot \big\langle \mathbf{M}(j,*), \mathbf{M}(j,*) \big\rangle.$$

Let $\mathbf{A} = \mathbf{M}\mathbf{M}^T$. We have $A_{i,i}, A_{i,j} = A_{j,i}, A_{j,j} > 0$, but $A_{i,i} \cdot A_{j,j} > A_{i,j} \cdot A_{j,i}$. Therefore, $\mathbf{A}$ is not block-rank-1. ∎

2.1. Counting graph homomorphisms. Each symmetric, nonnegative $n \times n$ matrix $\mathbf{A}$ defines a graph homomorphism (or partition) function $Z_{\mathbf{A}}(\cdot)$ as follows: Given any undirected graph $G = (V, E)$, we have

$$Z_{\mathbf{A}}(G) \stackrel{\text{def}}{=} \sum_{\xi: V \to [n]} \prod_{uv \in E} A\big(\xi(u), \xi(v)\big).$$

We need the following important result of Bulatov and Grohe [4] in the hardness part of our dichotomy theorem.

THEOREM 2.3. *Let* $\mathbf{A}$ *be a symmetric, nonnegative matrix with algebraic entries. Then the problem of computing* $Z_{\mathbf{A}}(\cdot)$ *is in polynomial time if* $\mathbf{A}$ *is* block-rank-1*, and is #P-hard otherwise.*

2.2. Weighted #CSPs. Let $D = \{1, \dots, d\}$ be the domain set, where the size d will be considered as a constant. A *weighted* constraint language $\mathcal{F}$ over the domain D is a finite set of functions $\{f_1, \dots, f_h\}$ for some $h \geq 1$ in which each $f_i : D^{r_i} \to \mathbb{R}$ is an r_i-ary function over D for some $r_i \geq 1$. The arity r_i of f_i, $i \in [h]$, the number h of functions in $\mathcal{F}$, as well as the values of f_i will all be considered as constants (except in section 6, where we study the decidability of the dichotomy). In this paper we only consider *nonnegative* weighted constraint languages $\mathcal{F}$ in which every f_i maps D^{r_i} to *nonnegative* and *algebraic* numbers.

The pair $(D, \mathcal{F})$ defines the following problem, which we denote by #CSP($\mathcal{F}$):

1. An instance of #CSP($\mathcal{F}$) is a pair (n, I), where n is the number of variables (indexed by $[n]$) and I is a sequence (or multiset) of m tuples $(f, i_1, \dots, i_r)$. For each $(f, i_1, \dots, i_r)$ in I, f is an r-ary function from $\mathcal{F}$ and is applied on variables indexed by $i_1, \dots, i_r \in [n]$. We call $n + m$ the size of (n, I).
2. Given an input instance (n, I), we define the following n-variable function F_I over D^n: For each assignment $\mathbf{x} = (x_1, \dots, x_n) \in D^n$,

$$F_I(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{(f, i_1, \dots, i_r) \in I} f(x_{i_1}, \dots, x_{i_r}),$$

and the output of the problem is the sum

$$Z(I) \stackrel{\text{def}}{=} \sum_{\mathbf{x}\in D^n} F_I(\mathbf{x}).$$

2.3. Reduction from unweighted to weighted #CSPs. A special case of #CSP($\mathcal{F}$) is when every function in $\mathcal{F}$ is $\{0,1\}$-valued. In this case we can view each function as a relation. We use the following notation for this special case.

An *unweighted* constraint language Γ over the domain D is a finite set of relations $\{\Theta_1,\dots,\Theta_h\}$ for some $h \ge 1$ in which each Θ_i is an r_i-ary relation over D^{r_i} for some $r_i \ge 1$. The pair (D,Γ) defines the following problem denoted by #CSP(Γ):

1. An instance of #CSP(Γ) is a pair (n, I), where n is the number of variables (indexed by $[n]$) and I is a sequence (or multiset) of m tuples $(\Theta, i_1,\dots,i_r)$. For each tuple in I, Θ is an r-ary relation in Γ and applied on variables indexed by $i_1,\dots,i_r \in [n]$. We call $n+m$ the size of (n, I).
2. Given an input instance (n, I), we define the following n-ary relation R_I over D^n: $\mathbf{x} = (x_1,\dots,x_n) \in D^n$ is in R_I if and only if we have $(x_{i_1},\dots,x_{i_r}) \in \Theta$ for every tuple $(\Theta, i_1,\dots,i_r) \in I$. Further, the output is the number of $\mathbf{x} \in D^n$ in R_I.

Given a nonnegative weighted constraint language $\mathcal{F} = \{f_1,\dots,f_h\}$, it is natural to define its corresponding unweighted constraint language $\Gamma = \{\Theta_1,\dots,\Theta_h\}$, where $\mathbf{x} \in \Theta_i$ if and only if $f_i(\mathbf{x}) > 0$ for all $i \in [h]$ and $\mathbf{x} \in D^{r_i}$. In section 7, we present a polynomial-time reduction from #CSP(Γ) to #CSP($\mathcal{F}$).

LEMMA 2.4. *Problem* #CSP(Γ) *is polynomial-time reducible to* #CSP($\mathcal{F}$).

2.4. Strong rectangularity. In the dichotomy theorem for unweighted #CSPs [7, 27], the following condition, called *strong rectangularity*, played a crucial role.

DEFINITION 2.5 (strong rectangularity). *We say an unweighted constraint language* Γ *over the domain set* D *is* strongly rectangular *if, for any input instance* (n, I) *of* #CSP(Γ) *which defines an* n*-ary relation* R_I *over* D^n *and for any integers* a *and* b *such that* $1 \le a < b \le n$, *the following* $d^a \times d^{b-a}$ *matrix* $\mathbf{M}$ *is* rectangular: *The rows of* $\mathbf{M}$ *are indexed by* $\mathbf{u} \in D^a$, *the columns of* $\mathbf{M}$ *are indexed by* $\mathbf{v} \in D^{b-a}$, *and*

$$M(\mathbf{u},\mathbf{v}) = \Big|\big\{\mathbf{w} \in D^{n-b} : (\mathbf{u},\mathbf{v},\mathbf{w}) \in R_I\big\}\Big| \quad \text{for all } \mathbf{u} \in D^a, \mathbf{v} \in D^{b-a}.$$

For the special case when $b = n$, $M(\mathbf{u},\mathbf{v}) = 1$ *if* $(\mathbf{u},\mathbf{v}) \in R_I$, *and* 0 *otherwise.*

The following theorem is proved in [7] and [27].

THEOREM 2.6. *If* Γ *is not strongly rectangular, then* #CSP(Γ) *is* #P*-hard.*

Let $\mathcal{F}$ be a nonnegative constraint language, and let Γ be its corresponding unweighted language. By Lemma 2.4 and Theorem 2.6, #CSP($\mathcal{F}$) is #P-hard if Γ is not strongly rectangular. When Γ is strongly rectangular, we can use the following algorithmic results for #CSP(Γ) from [27] (using the succinct and efficiently computable data structure called a *frame*), in the quest of obtaining a polynomial-time algorithm for the original weighted problem #CSP($\mathcal{F}$). We start with some notation. Let (n, I) be an input instance of #CSP(Γ) which defines a relation R over D^n.

DEFINITION 2.7. *We use* $\mathsf{pr}_i R \subseteq D$, $i \in [n]$, *to denote the projection of* R *on the* i*th coordinate:* $a \in \mathsf{pr}_i R$ *if and only if there exist* $\mathbf{u} \in D^{i-1}$ *and* $\mathbf{v} \in D^{n-i}$ *such that* $(\mathbf{u}, a, \mathbf{v}) \in R$. *We define the following relation* $\sim_i$ *on* $\mathsf{pr}_i R$: $a \sim_i b$ *if there exist tuples* $\mathbf{u} \in D^{i-1}$ *and* $\mathbf{v}_a, \mathbf{v}_b \in D^{n-i}$ *such that* $(\mathbf{u}, a, \mathbf{v}_a) \in R$ *and* $(\mathbf{u}, b, \mathbf{v}_b) \in R$.

LEMMA 2.8 (see [27]). *If* Γ *is strongly rectangular, then for any input instance* (n, I) *of* $\#\mathrm{CSP}(\Gamma)$ *which defines an* n*-ary relation* R *over* D^n*, we have the following:*

(A) *For each* $i \in [n]$*, we can compute the set* $\mathsf{pr}_i R$ *in polynomial time in the size of* (n, I)*. Moreover, for every* $a \in \mathsf{pr}_i R$*, we can find a tuple* $\mathbf{u} \in R$ *such that* $u_i = a$ *in polynomial time.*

(B) *For each* $i \in [n]$*, the relation* $\sim_i$ *must be an* equivalence *relation and can be computed in polynomial time in the size of* (n, I)*. We will use* $\mathcal{E}_{i,k} \subseteq D$*,* $k = 1, \ldots,$ *to denote the equivalence classes of* $\sim_i$*.*

(C) *For each equivalence class* $\mathcal{E}_{i,k}$*, we can find in polynomial time a tuple* $\mathbf{u}^{[i,k]} \in D^{i-1}$ *and a tuple* $\mathbf{v}^{[i,k,a]} \in D^{n-i}$ *for each element* $a \in \mathcal{E}_{i,k}$ *such that* $(\mathbf{u}^{[i,k]}, a, \mathbf{v}^{[i,k,a]}) \in R$ *for every* $a \in \mathcal{E}_{i,k}$*.*

3. A dichotomy theorem for nonnegatively weighted #CSPs and its decidability. We state our dichotomy theorems in this section. The lemmas used in their proofs are proved in the rest of the paper.

In our dichotomy theorems the following two notions of *weak balance* and *balance* play a crucial role. They are similar to, and in some sense weaker than, the notion of *strong balance* used in [27] (no existential quantifier is used in the definitions).

DEFINITION 3.1 (weak balance). *We say a nonnegatively weighted language* $\mathcal{F}$ *over the domain set* D *is* weakly balanced *if for any input instance* (n, I) *of* $\#\mathrm{CSP}(\mathcal{F})$ *which defines a nonnegative and algebraic function* $F(x_1, \ldots, x_n)$ *over* D^n *and any integer* $a : 1 \le a < n$*, the following* $d^a \times d$ *matrix* $\mathbf{M}$ *is* block-rank-1: *The rows of* $\mathbf{M}$ *are indexed by tuples* $\mathbf{u} \in D^a$*, the columns of* $\mathbf{M}$ *are indexed by* $v \in D$*, and*

$$M(\mathbf{u}, v) = \sum_{\mathbf{w} \in D^{n-a-1}} F(\mathbf{u}, v, \mathbf{w}) \quad \text{for all } \mathbf{u} \in D^a \text{ and } v \in D.$$

For the special case when $a + 1 = n$*, we have* $M(\mathbf{u}, v) = F(\mathbf{u}, v)$ *is block-rank-*1*.*

DEFINITION 3.2 (balance). *We say* $\mathcal{F}$ *is* balanced *if for any input instance* (n, I) *of* $\#\mathrm{CSP}(\mathcal{F})$ *which defines a nonnegative and algebraic function* $F(x_1, \ldots, x_n)$ *over* D^n *and any* $a, b : 1 \le a < b \le n$*, the following* $d^a \times d^{b-a}$ *matrix* $\mathbf{M}$ *is* block-rank-1: *The rows of* $\mathbf{M}$ *are indexed by* $\mathbf{u} \in D^a$*, the columns are indexed by* $\mathbf{v} \in D^{b-a}$*, and*

$$M(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{w} \in D^{n-b}} F(\mathbf{u}, \mathbf{v}, \mathbf{w}) \quad \text{for all } \mathbf{u} \in D^a \text{ and } \mathbf{v} \in D^{b-a}.$$

For the special case when $b = n$*, we have* $M(\mathbf{u}, \mathbf{v}) = F(\mathbf{u}, \mathbf{v})$ *is block-rank-*1*.*

It is clear that *balance* implies *weak balance*. We prove the following complexity dichotomy theorem on nonnegatively weighted #CSPs.

THEOREM 3.3. *For any constraint language* $\mathcal{F}$ *with nonnegative and algebraic weights,* $\#\mathrm{CSP}(\mathcal{F})$ *is in polynomial time if its corresponding unweighted language* Γ *is strongly rectangular and* $\mathcal{F}$ *is weakly balanced, and it is* #P*-hard otherwise.*

Proof. Combining Lemma 2.4 and Theorem 2.6, $\#\mathrm{CSP}(\mathcal{F})$ is #P-hard if Γ is not strongly rectangular, where Γ is the unweighted constraint language that corresponds to $\mathcal{F}$. We prove the following hardness lemma in section 8, showing that $\#\mathrm{CSP}(\mathcal{F})$ is #P-hard if $\mathcal{F}$ is not balanced.

LEMMA 3.4. *If* $\mathcal{F}$ *is not balanced, then* $\#\mathrm{CSP}(\mathcal{F})$ *is* #P*-hard.*

As balance implies weak balance, $\#\mathrm{CSP}(\mathcal{F})$ must be #P-hard if $\mathcal{F}$ is not weakly balanced. This finishes the proof of the hardness part of the theorem. In the next two sections (sections 4 and 5) we focus on the proof of the following algorithmic lemma.

LEMMA 3.5. *If* Γ *is strongly rectangular and* $\mathcal{F}$ *is weakly balanced, then* #CSP($\mathcal{F}$) *can be solved in polynomial time.*

This finishes the proof of the dichotomy theorem. ∎

We can use Lemmas 3.4 and 3.5 to obtain the following dichotomy theorem that uses the notion of balance only in its dichotomy criterion.

THEOREM 3.6. #CSP($\mathcal{F}$) *is in polynomial time if* $\mathcal{F}$ *is balanced; otherwise, it is* #P*-hard.*

Proof. By Lemma 3.4, #CSP($\mathcal{F}$) is #P-hard if $\mathcal{F}$ is not balanced.

Next it follows from the definitions of strong rectangularity and balance (since a matrix that is block-rank-1 must first be rectangular) that $\mathcal{F}$ is strongly rectangular if it is balanced. It follows from Lemma 3.5 that #CSP($\mathcal{F}$) can be solved in polynomial time if $\mathcal{F}$ is balanced. This finishes the proof of the theorem. ∎

Finally, we show that the dichotomy criterion stated in Theorem 3.3 is decidable in NP. Given D and $\mathcal{F}$, we are interested in the decision problem of whether $\mathcal{F}$ satisfies the following two conditions: (1) Γ is strongly balanced, and (2) $\mathcal{F}$ is weakly balanced. (Note that here D and $\mathcal{F} = \{f_1, \dots, f_h\}$ are no longer considered as constants, but as the input of the decision problem. The input size is $|D|$ plus the number of bits needed to represent functions $f_1, \dots, f_h$ in $\mathcal{F}$, each as a table giving its value for each input tuple. We also follow the standard model of [38] for encoding algebraic numbers.) We prove the following theorem in section 6. The proof follows the approach of Dyer and Richerby [27], with some new ideas and constructions developed for the more general weighted case. It uses a method of Lovász [39], which was also used earlier in [25].

THEOREM 3.7. *Given* D *and* $\mathcal{F}$*, the problem of deciding whether* $\mathcal{F}$ *satisfies the criterion stated in Theorem* 3.3 *is in* NP.

4. Vector representation. Assume that $\mathcal{F}$ is weakly balanced, and let f be an r-ary function in $\mathcal{F}$. We use Θ to denote the corresponding r-ary relation of f in Γ. In this section, we show that there must exist r nonnegative, one-variable functions $s_1, \dots, s_r : D \to \mathbb{R}_+$, such that for all $\mathbf{x} \in D^r$, either $\mathbf{x} \notin \Theta$ and $f(\mathbf{x}) = 0$, or we have $f(\mathbf{x}) = s_1(x_1) \cdots s_r(x_r)$. We call an $\mathbf{s} = (s_1, \dots, s_r)$ that satisfies the above property a *vector representation* of f. We prove the following lemma.

LEMMA 4.1. *If* $\mathcal{F}$ *is weakly balanced, every* $f \in \mathcal{F}$ *has a vector representation.*

To this end, we need the following notation. Let f be any r-ary function over D. Then for any $\ell \in [r]$, we use $f^{[\ell]}$ to denote the following ℓ-ary function over D:

$$f^{[\ell]}(x_1, \dots, x_\ell) \stackrel{\text{def}}{=} \sum_{x_{\ell+1}, \dots, x_r \in D} f(x_1, \dots, x_\ell, x_{\ell+1}, \dots, x_r) \quad \text{for all } x_1, \dots, x_\ell \in D.$$

In particular, we have $f^{[r]} \equiv f$.

Let f be an r-ary, nonnegative function with $r \geq 1$. We say f is *block-rank-*1 if either $r = 1$, or the following $d^{r-1} \times d$ matrix $\mathbf{M}$ is block-rank-1: The rows of $\mathbf{M}$ are indexed by tuples $\mathbf{u} \in D^{r-1}$, the columns are indexed by $v \in D$, and $M(\mathbf{u}, v) = f(\mathbf{u}, v)$ for all $\mathbf{u} \in D^{r-1}$ and $v \in D$.

Using the definition of weak balance, Lemma 4.1 is a corollary of the following lemma.

LEMMA 4.2. *Let* $f(x_1, \dots, x_r)$ *be an* r*-ary nonnegative function. If* $f^{[\ell]}$ *is block-rank-*1 *for every* $\ell \in [r]$*, then* f *has a vector representation* $\mathbf{s}$.

Proof. We prove the lemma by induction on r, the arity of f.

The base case when $r = 1$ is trivial. Now we assume for induction that the claim is true for all $(r-1)$-ary nonnegative functions, for some $r \geq 2$, and let f be an r-ary nonnegative function such that $f^{[\ell]}$ is block-rank-1 for all $\ell \in [r]$. By definition,

$$\left(f^{[r-1]}\right)^{[\ell]} = f^{[\ell]} \qquad \text{for all } \ell \in [r-1].$$

As a result, if we denote $f^{[r-1]}$, an $(r-1)$-ary nonnegative function, by g, then $g^{[\ell]}$ is block-rank-1 for all $\ell \in [r-1]$. Therefore, by the inductive hypothesis, $g = f^{[r-1]}$ has a vector representation $(s_1, \ldots, s_{r-1})$.

Finally, we show how to construct s_r, so that $(s_1, \ldots, s_{r-1}, s_r)$ is a vector representation of f. To this end we let $\mathbf{M}$ denote the following $d^{r-1} \times d$ matrix: The rows are indexed by $\mathbf{u} \in D^{r-1}$, the columns are indexed by $v \in D$, and $M(\mathbf{u}, v) = f(\mathbf{u}, v)$ for every $\mathbf{u} \in D^{r-1}$ and $v \in D$. By the assumption, we know that $\mathbf{M}$ is block-rank-1. Therefore, by definition, there exist pairwise disjoint and nonempty subsets of D^{r-1}, denoted by $A_1, \ldots, A_s$, and pairwise disjoint and nonempty subsets of D, denoted by $B_1, \ldots, B_s$, for some $s \geq 0$, such that $M(\mathbf{u}, v) > 0$ if and only if $\mathbf{u} \in A_i$ and $v \in B_i$ for some $i \in [s]$, and for every $i \in [s]$, the $A_i \times B_i$ submatrix of $\mathbf{M}$ is of rank 1.

We now construct $s_r : D \to \mathbb{R}_+$ as follows. For each $i \in [s]$, we arbitrarily pick a vector from A_i and denote it $\mathbf{u}_i$. Then for each $v \in D$, we set $s_r(v)$ as follows:

1. If $v \notin B_i$ for any $i \in [s]$, then $s_r(v) = 0$.
2. Otherwise, assume that $v \in B_i$ (which must be unique). Then

$$s_r(v) = \frac{M(\mathbf{u}_i, v)}{\sum_{v' \in B_i} M(\mathbf{u}_i, v')}. \tag{4.1}$$

To prove that $(s_1, \ldots, s_r)$ is indeed a vector representation of f, we need only show that for every tuple $(\mathbf{u}, v)$ such that $\mathbf{u} \in A_i$ and $v \in B_i$ for some $i \in [s]$ (since otherwise we have $f(\mathbf{u}, v) = 0$), we have

$$f(\mathbf{u}, v) = M(\mathbf{u}, v) = s_r(v) \prod_{j \in [r-1]} s_j(u_j).$$

By using Lemma 2.1 and (4.1), we have

$$M(\mathbf{u}, v) = M(\mathbf{u}_i, v) \cdot \frac{\sum_{v' \in B_i} M(\mathbf{u}, v')}{\sum_{v' \in B_i} M(\mathbf{u}_i, v')} = s_r(v) \cdot f^{[r-1]}(\mathbf{u}) = s_r(v) \prod_{j \in [r-1]} s_j(u_j),$$

where the last equation follows from the inductive hypothesis that $(s_1, \ldots, s_{r-1})$ is a vector representation of $g = f^{[r-1]}$.

This finishes the induction, and the lemma is proved. ∎

5. Tractability: The counting algorithm. In this section we prove Lemma 3.5 by giving a polynomial-time algorithm for #CSP($\mathcal{F}$), assuming that Γ is strongly rectangular and $\mathcal{F}$ is weakly balanced. As mentioned earlier, as Γ is strongly rectangular, we can use the three polynomial-time algorithms summarized in Lemma 2.8 as subroutines. Also, because $\mathcal{F}$ is weakly balanced, we may assume, by Lemma 4.1, that every r-ary function f in $\mathcal{F}$ has a vector representation $\mathbf{s}_f = (s_{f,1}, \ldots, s_{f,r})$, where $s_{f,i} : D \to \mathbb{R}_+$ for all $i \in [r]$. (Note that since each f in $\mathcal{F}$ is considered as a constant, we may assume that $\mathbf{s}_f$ for f is given and is considered as a constant as well.)

Now let (n, I) be an instance of #CSP($\mathcal{F}$) and let F denote the function it defines over $\mathbf{x} = (x_1, \ldots, x_n) \in D^n$. For each tuple in I, one can replace the first component, that is, a function f in $\mathcal{F}$, by its corresponding relation Θ in Γ. We use I' to denote the new set, which is an instance of #CSP(Γ) and defines a relation R over $\mathbf{x} \in D^n$ (R is also referred to as the *feasibility relation* in literature on valued CSPs. We have $F(\mathbf{x}) > 0$ if and only if $\mathbf{x} \in R$ for all $\mathbf{x} \in D^n$.

Our algorithm starts by computing a vector representation $\mathbf{s} = (s_1, \ldots, s_n)$ of F using the known vector representations $\mathbf{s}_f$ of each $f \in \mathcal{F}$.

LEMMA 5.1. *Given I, one can compute $s_1(\cdot), \ldots, s_n(\cdot)$ in polynomial time such that for all $\mathbf{x} \in D^n$, either $\mathbf{x} \notin R$ and $F(\mathbf{x}) = 0$, or $F(\mathbf{x}) = s_1(x_1) \cdots s_n(x_n)$.*

Proof. We start with $s_1, \ldots, s_n$, where $s_i(a) = 1$ for all $i \in [n]$ and $a \in D$. Next we enumerate the constraints in I one by one. For each constraint $(f, i_1, \ldots, i_r) \in I$ and each $j \in [r]$, we update the function $s_{i_j}(\cdot)$ using $s_{f,j}(\cdot)$ as follows:

$$s_{i_j}(a) \stackrel{\text{set}}{=} s_{i_j}(a) \cdot s_{f,j}(a) \quad \text{for each } a \in D.$$

It is easy to check that the tuple $(s_1, \ldots, s_n)$ we get is a vector representation of F. This completes the proof. ∎

The second step of the algorithm computes a sequence of one-variable functions $t_n(\cdot), t_{n-1}(\cdot), \ldots, t_2(\cdot)$ with the following property: For any $i \in \{1, \ldots, n-1\}$ and for any $\mathbf{u} \in R$, we have

$$\begin{aligned} \sum_{x_{i+1},\ldots,x_n \in D} & F(u_1, \ldots, u_i, x_{i+1}, \ldots, x_n) \\ &= s_1(u_1) \cdots s_i(u_i) \cdot \frac{s_{i+1}(u_{i+1})}{t_{i+1}(u_{i+1})} \cdots \frac{s_n(u_n)}{t_n(u_n)}. \end{aligned} \tag{5.1}$$

Before describing the algorithm, we show that $Z(I)$ is easy to compute given the t_i's.

For this purpose we first compute $\mathsf{pr}_1 R$ in polynomial time using the algorithm in Lemma 2.8(A). In addition, we find a vector $\mathbf{u}_a = (u_{a,1}, u_{a,2}, \ldots, u_{a,n}) \in R$ for each $a \in \mathsf{pr}_1 R$ such that $u_{a,1} = a$ in polynomial time. Then we have

$$\begin{aligned} Z(I) = \sum_{\mathbf{x} \in D^n} F(\mathbf{x}) &= \sum_{a \in \mathsf{pr}_1 R} \; \sum_{x_2,\ldots,x_n \in D} F(a, x_2, \ldots, x_n) \\ &= \sum_{a \in \mathsf{pr}_1 R} s_1(a) \prod_{j \in [2:n]} \left(\frac{s_j(u_{a,j})}{t_j(u_{a,j})} \right), \end{aligned}$$

which can be evaluated in polynomial time using $s_1, \ldots, s_n$ and $t_2, \ldots, t_n$.

We now show how to compute $t_n, t_{n-1}, \ldots, t_2$ one by one in this order and prove (5.1) by induction. We start with $t_n(\cdot)$.

Because $\mathcal{F}$ is weakly balanced, the following $d^{n-1} \times d$ matrix $\mathbf{M}$ must be block-rank-1: The rows are indexed by $\mathbf{u} \in D^{n-1}$, the columns are indexed by $v \in D$, and $M(\mathbf{u}, v) = F(\mathbf{u}, v)$ for all $\mathbf{u} \in D^{n-1}$ and $v \in D$. By the definition of $\sim_n$, we have that $v_1 \sim_n v_2$ if and only if columns v_1 and v_2 are in the same block of $\mathbf{M}$, and thus, the equivalent classes $\{\mathcal{E}_{n,k}\}$ are exactly the column index sets of those blocks of $\mathbf{M}$.

We define $t_n(\cdot)$ as follows. For each $a \in D$, if $a \notin \mathsf{pr}_n R$, then $t_n(a) = 0$; otherwise, a belongs to one of the equivalence classes $\mathcal{E}_{n,k}$ of $\sim_n$ and

$$t_n(a) = \frac{s_n(a)}{\sum_{b \in \mathcal{E}_{n,k}} s_n(b)}. \tag{5.2}$$

Using the algorithm in Lemma 2.8(B), $t_n(\cdot)$ can be computed in polynomial time. We now prove (5.1) for $i = n-1$. Given any $\mathbf{u} \in R$, we have $u_n \in \mathsf{pr}_n R$ by definition and let $\mathcal{E}_{n,k}$ denote the equivalence class to which u_n belongs. Then

$$\sum_{b\in D} F(u_1,\ldots,u_{n-1},b) = \sum_{b\in\mathcal{E}_{n,k}} F(u_1,\ldots,u_{n-1},b)$$
$$= \prod_{j\in[n-1]} s_j(u_j) \sum_{b\in\mathcal{E}_{n,k}} s_n(b) = \prod_{j\in[n-1]} s_j(u_j)\cdot\frac{s_n(u_n)}{t_n(u_n)}.$$

The last equation follows from (5.2) of $t_n(\cdot)$ and the assumption that $u_n \in \mathcal{E}_{n,k}$.

Next, assume for induction that we have already computed $t_{i+1},\ldots,t_n$, for some $i \in [2 : n-1]$, and they together satisfy (5.1). To compute $t_i(\cdot)$, we first observe that the following $d^{i-1}\times d$ matrix $\mathbf{M}$ must be block-rank-1, because $\mathcal{F}$ is weakly balanced: The rows are indexed by $\mathbf{u} = (u_1,\ldots,u_{i-1}) \in D^{i-1}$, the columns are indexed by $v \in D$, and

$$M(\mathbf{u}, v) = \sum_{\mathbf{w}\in D^{n-i}} F(\mathbf{u}, v, \mathbf{w}) = \sum_{(\mathbf{u},v,\mathbf{w})\in R} F(\mathbf{u}, v, \mathbf{w}).$$

Similarly, by the definition of $\sim_i$, its equivalent classes $\{\mathcal{E}_{i,k}\}$ are precisely the column index sets of those blocks of $\mathbf{M}$. Using (5.1) and the inductive hypothesis, we have a concise form for $M(\mathbf{u}, v)$: For any $\mathbf{w} = (w_{i+1},\ldots,w_n) \in D^{n-i}$ such that $(\mathbf{u}, v, \mathbf{w}) \in R$,

$$M(\mathbf{u}, v) = \left(\prod_{j\in[i-1]} s_j(u_j)\right) s_i(v) \left(\prod_{j\in[i+1:n]} \frac{s_j(w_j)}{t_j(w_j)}\right). \tag{5.3}$$

Note that by (5.1), the choice of $\mathbf{w}$ can be arbitrary as long as $(\mathbf{u}, v, \mathbf{w}) \in R$.

We now define $t_i(\cdot)$. For every $a \in D$, the following hold:

1. If $a \notin \mathsf{pr}_i R$, then $t_i(a) = 0$.
2. Otherwise, let $\mathcal{E}_{i,k}$ denote the equivalence class of $\sim_i$ to which a belongs. Then by using the algorithm in Lemma 2.8(C), we find a tuple $\mathbf{u}^{[i,k]} \in D^{i-1}$ and a tuple $\mathbf{v}^{[i,k,b]} \in D^{n-i}$ for each $b \in \mathcal{E}_{i,k}$ such that

$$\left(\mathbf{u}^{[i,k]}, b, \mathbf{v}^{[i,k,b]}\right) \in R \quad \text{for all } b \in \mathcal{E}_{i,k}.$$

Then we set

$$t_i(a) = \frac{M(\mathbf{u}^{[i,k]}, a)}{\sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^{[i,k]}, b)}. \tag{5.4}$$

By (5.3), $t_i(a)$ can be computed efficiently using $\mathbf{u}^{[i,k]}, \mathbf{v}^{[i,k,b]}$ for $b \in \mathcal{E}_{i,k}$. This finishes the definition of $t_i(\cdot)$.

Finally we prove (5.1). Let $\mathbf{u}$ be any tuple in R, and let $\mathcal{E}_{i,k}$ be the equivalence class of $\sim_i$ to which u_i belongs. Then

$$\sum_{x_i,\ldots,x_n} F(u_1,\ldots,u_{i-1},x_i,\ldots,x_n) = \sum_{b\in\mathcal{E}_{i,k}} \sum_{x_{i+1},\ldots,x_n} F(u_1,\ldots,u_{i-1},b,x_{i+1},\ldots,x_n).$$

Let $\mathbf{u}^*$ denote the $(i-1)$-tuple $(u_1,\ldots,u_{i-1})$. Then by the definition of $\mathbf{M}$, we have

$$\sum_{x_i,\ldots,x_n\in D} F(u_1,\ldots,u_{i-1},x_i,\ldots,x_n) = \sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^*, b).$$

Recall $\mathbf{u}^{[i,k]}$ and $\mathbf{v}^{[i,k,b]}$, $b \in \mathcal{E}_{i,k}$, which we used earlier in the definition of $t_i(\cdot)$. Since $\mathbf{M}$ is block-rank-1 and $\mathbf{u}^*$ and $\mathbf{u}^{[i,k]}$ belong to the same block of $\mathbf{M}$, we have

$$\sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^*, b) = \sum_{b\in\mathcal{E}_{i,k}} \frac{M(\mathbf{u}^*, u_i)}{M(\mathbf{u}^{[i,k]}, u_i)} \cdot M(\mathbf{u}^{[i,k]}, b) = \frac{M(\mathbf{u}^*, u_i)}{M(\mathbf{u}^{[i,k]}, u_i)} \cdot \sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^{[i,k]}, b).$$

However, by the definition (5.4) of $t_i(\cdot)$, we have

$$\sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^{[i,k]}, b) = \frac{M(\mathbf{u}^{[i,k]}, u_i)}{t_i(u_i)},$$

since we assumed that $u_i \in \mathcal{E}_{i,k}$. As a result, we have

$$\begin{aligned}\sum_{x_i,\ldots,x_n\in D} F(u_1,\ldots,u_{i-1},x_i,\ldots,x_n) &= \sum_{b\in\mathcal{E}_{i,k}} M(\mathbf{u}^*, b) = \frac{M(\mathbf{u}^*, u_i)}{t_i(u_i)} \\ &= \left(\prod_{j\in[i-1]} s_j(u_j)\right)\left(\prod_{j\in[i:n]} \frac{s_j(u_j)}{t_j(u_j)}\right).\end{aligned}$$

The last equation follows from (5.3).

This finishes the description of our algorithm and the proof of Lemma 3.5.

6. Decidability of the dichotomy criterion. In this section we prove Theorem 3.7 by showing that the following decision problem is in NP: Given D and $\mathcal{F}$ (see the discussion on the representation of $\mathcal{F}$ at the end of section 3), decide whether Γ is strongly rectangular and $\mathcal{F}$ is weakly balanced. The strong rectangularity part can be done in NP [7, 27] by nondeterministically guessing a Mal'tsev polymorphism.

LEMMA 6.1 (see [7, 27]). *Given an unweighted constraint language Γ, the problem of deciding whether Γ is strongly rectangular is in* NP.

Thus in the rest of the section we focus on the decision problem of checking whether or not $\mathcal{F}$ is weakly balanced.

6.1. Primitive balance. First we show that the notion of weak balance is actually equivalent to the following seemingly *weaker* notion of *primitive balance*.

DEFINITION 6.2 (primitive balance). *We say $\mathcal{F}$ is* primitively balanced *if for any instance (n, I) of* #CSP($\mathcal{F}$) *which defines an n-ary function $F_I(x_1,\ldots,x_n)$ over D^n the following $d\times d$ matrix $\mathbf{M}_I$ is block-rank-1: The rows of $\mathbf{M}_I$ are indexed by $x_1 \in D$, the columns are indexed by $x_2 \in D$, and*

$$M_I(x_1,x_2) = \sum_{x_3,\ldots,x_n\in D} F_I(x_1,x_2,x_3,\ldots,x_n) \quad \text{for all } x_1,x_2\in D. \tag{6.1}$$

Clearly weak balance implies primitive balance. We prove the converse direction.

LEMMA 6.3. *If $\mathcal{F}$ is primitively balanced, then it is weakly balanced as well.*

Proof. Assume for a contradiction that $\mathcal{F}$ is primitively balanced but not weakly balanced. By definition, this means there exist an instance I over n variables and an integer $a : 1 \le a < n$ such that the following $d^a \times d$ matrix $\mathbf{M}$ is not block-rank-1: The rows of $\mathbf{M}$ are indexed by $\mathbf{u} \in D^a$, the columns are indexed by $v \in D$, and

$$M(\mathbf{u}, v) = \sum_{\mathbf{w}\in D^{n-a-1}} F_I(\mathbf{u}, v, \mathbf{w}) \quad \text{for all } \mathbf{u}\in D^a \text{ and } v\in D.$$

As a result, it follows from Lemma 2.2 that $\mathbf{A} = \mathbf{M}\mathbf{M}^T$ is not block-rank-1.

For a contradiction, we construct I' from I as follows: I' has $2n-a$ variables in the order $x_1, x_2, y_1, \ldots, y_a, z_1, \ldots, z_{n-a-1}, w_1, \ldots, w_{n-a-1}$. The instance I' consists of two parts: a copy of I over $(y_1, \ldots, y_a, x_1, z_1, \ldots, z_{n-a-1})$ and a copy of I over $(y_1, \ldots, y_a, x_2, w_1, \ldots, w_{n-a-1})$.

Let $F_{I'}$ be the function that I' defines. It gives us the following $d \times d$ matrix $\mathbf{M}_{I'}$:

$$\begin{aligned} M_{I'}(x_1, x_2) &= \sum_{\substack{\mathbf{y} \in D^a \\ \mathbf{z}, \mathbf{w} \in D^{n-a-1}}} F_I(\mathbf{y}, x_1, \mathbf{z}) \cdot F_I(\mathbf{y}, x_2, \mathbf{w}) \\ &= \sum_{\mathbf{y} \in D^a} M(\mathbf{y}, x_1) \cdot M(\mathbf{y}, x_2) = A(x_1, x_2), \end{aligned}$$

which is not block-rank-1 and contradicts the assumption that $\mathcal{F}$ is primitively balanced. This completes the proof of the lemma. ∎

Given Lemma 6.3, the decision problem reduces to the following:

PRIMITIVE BALANCE: Given D and $\mathcal{F}$ such that Γ is strongly rectangular (which by Lemma 6.1 can be verified in NP), decide whether $\mathcal{F}$ is primitively balanced or not.

As Γ is strongly rectangular, we know that for any instance I of #CSP($\mathcal{F}$), the $d \times d$ matrix $\mathbf{M}_I$ defined in (6.1) is rectangular. We need the following useful lemma from [27], which can be used to check whether a rectangular matrix is block-rank-1.

LEMMA 6.4 (see [27]). *A rectangular $d \times d$ matrix $\mathbf{M}$ is block-rank-1 if and only if*

$$M(\alpha,\kappa)^2 M(\beta,\lambda)^2 M(\alpha,\lambda) M(\beta,\kappa) = M(\alpha,\lambda)^2 M(\beta,\kappa)^2 M(\alpha,\kappa) M(\beta,\lambda) \tag{6.2}$$

for all $\alpha \neq \beta \in D$ and $\kappa \neq \lambda \in D$.

As a consequence, for PRIMITIVE BALANCE it suffices to check whether (6.2) holds for $\mathbf{M}_I$ over all instances I of #CSP($\mathcal{F}$) and for all $\alpha \neq \beta, \kappa \neq \lambda \in D$. In the rest of this section, we fix $\alpha \neq \beta \in D$ and $\kappa \neq \lambda \in D$ and prove that the decision problem (i.e., whether (6.2) holds for all I) is in NP. Theorem 3.7 follows directly since there are only polynomially many possible tuples $(\alpha, \beta, \kappa, \lambda)$ to check.

6.2. Reformulation of the decision problem. Fixing $\alpha \neq \beta \in D$ and $\kappa \neq \lambda \in D$, we follow [27] and reformulate the decision problem PRIMITIVE BALANCE using a new pair $(\mathfrak{D}, \mathfrak{F})$, that is, the 6*th power* of #CSP($\mathcal{F}$):

1. The new domain $\mathfrak{D} = D^6$, and we use $\mathfrak{s} = (s_1, \ldots, s_6)$ to denote an element in $\mathfrak{D}$, where $s_i \in D$ for each $i \in [6]$.
2. $\mathfrak{F} = \{g_1, \ldots, g_h\}$ has the same number of functions as $\mathcal{F}$, and each g_i, $i \in [h]$, has the same arity r_i as f_i. Function $g_i : \mathfrak{D}^{r_i} \to \mathbb{R}_+$ is defined explicitly from f_i as follows:

$$g_i(\mathfrak{s}_1, \ldots, \mathfrak{s}_{r_i}) = \prod_{j \in [6]} f_i(s_{1,j}, \ldots, s_{r_i,j}) \quad \text{for all } \mathfrak{s}_1, \ldots, \mathfrak{s}_{r_i} \in \mathfrak{D} = D^6.$$

In the rest of the section, we will always use x_i to denote variables over D, and y_i, z_i to denote variables over $\mathfrak{D}$.

An input instance I of #CSP($\mathcal{F}$) over n variables $(x_1, \ldots, x_n)$ naturally corresponds to an input instance $\mathfrak{I}$ of $(\mathfrak{D}, \mathfrak{F})$ over n variables $(y_1, \ldots, y_n)$ as follows: For each tuple $(f, i_1, \ldots, i_r) \in I$, add a tuple $(g, i_1, \ldots, i_r)$ to $\mathfrak{I}$, where $g \in \mathfrak{F}$ corresponds to $f \in \mathcal{F}$. Moreover, this is clearly a bijection between the set of all I and the set of all $\mathfrak{I}$. Similarly, we let $G : \mathfrak{D}^n \to \mathbb{R}_+$ denote the n-ary function that $\mathfrak{I}$ defines:

$$G(y_1, \ldots, y_n) = \prod_{(g,i_1,\ldots,i_r)\in\mathfrak{I}} g(y_{i_1}, \ldots, y_{i_r}) \quad \text{for all } y_1, \ldots, y_n \in \mathfrak{D}.$$

We introduce the new pair $(\mathfrak{D}, \mathfrak{F})$ because it gives us a new and much simpler formulation of the decision problem in which we are interested.

To see this, we let $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ denote the following three specific elements from $\mathfrak{D}$:

$$\mathfrak{a} = (\alpha, \alpha, \alpha, \beta, \beta, \beta), \quad \mathfrak{b} = (\kappa, \kappa, \lambda, \lambda, \lambda, \kappa), \quad \mathfrak{c} = (\lambda, \lambda, \kappa, \kappa, \kappa, \lambda).$$

Since $\alpha \neq \beta$ and $\kappa \neq \lambda$, $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ are three distinct elements in $\mathfrak{D}$. We adopt the notation of [27]. For each $\mathfrak{s} \in \mathfrak{D}$, let

$$\text{hom}_{\mathfrak{s}}(\mathfrak{I}) \stackrel{\text{def}}{=} \sum_{y_3,\ldots,y_n\in\mathfrak{D}} G(\mathfrak{a}, \mathfrak{s}, y_3, \ldots, y_n) \quad \text{for every instance } \mathfrak{I} \text{ of } (\mathfrak{D}, \mathfrak{F}).$$

It is easy to prove the following two equations. Let $\mathfrak{I}$ be the instance of $(\mathfrak{D}, \mathfrak{F})$ that corresponds to I, and let $\mathbf{M}_I$ be the $d \times d$ matrix as defined in (6.1). Then

$$\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = M_I(\alpha, \kappa)^2 M_I(\beta, \lambda)^2 M_I(\alpha, \lambda) M_I(\beta, \kappa),$$
$$\text{hom}_{\mathfrak{c}}(\mathfrak{I}) = M_I(\alpha, \lambda)^2 M_I(\beta, \kappa)^2 M_I(\alpha, \kappa) M_I(\beta, \lambda).$$

As a result, we have the following reformulation of the decision problem:

$$\mathbf{M}_I \text{ satisfies (6.2) for all } I \iff \text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I}) \text{ for all } \mathfrak{I}.$$

The next reformulation considers sums over *injective* tuples only. We say a tuple $(y_1, \ldots, y_n) \in \mathfrak{D}^n$ is an injective tuple if $y_i \neq y_j$ for all $i \neq j \in [n]$ (or, equivalently, if we view $(y_1, \ldots, y_n)$ as a map from $[n]$ to $\mathfrak{D}$, it is injective). We use Y_n to denote the set of injective n-tuples. (Clearly this definition is only useful when $n \leq |\mathfrak{D}|$; otherwise Y_n is empty.) We now define functions $\text{mon}_{\mathfrak{s}}(\mathfrak{I})$, which are sums over injective tuples: For each $\mathfrak{s} \in \mathfrak{D}$, let

$$\text{mon}_{\mathfrak{s}}(\mathfrak{I}) \stackrel{\text{def}}{=} \sum_{(\mathfrak{a},\mathfrak{s},y_3,\ldots,y_n)\in Y_n} G(\mathfrak{a}, \mathfrak{s}, y_3, \ldots, y_n) \quad \text{for every instance } \mathfrak{I} \text{ of } (\mathfrak{D}, \mathfrak{F}).$$

The following lemma shows that $\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I})$ for all $\mathfrak{I}$ if and only if the same equation holds for the sums over injective tuples. The proof is exactly the same as that of Lemma 41 in [27], using the Möbius inversion. So we skip it here.

LEMMA 6.5 (see [27, Lemma 41]). $\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$ *if and only if we have* $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$.

Finally, the following reformulation gives a condition that can be checked in NP.

LEMMA 6.6. $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$ *if and only if there exists a bijection* π *from the domain* $\mathfrak{D}$ *to itself* (*which we will refer to as an* automorphism *of* $(\mathfrak{D}, \mathfrak{F})$) *such that* $\pi(\mathfrak{a}) = \pi(\mathfrak{a})$, $\pi(\mathfrak{b}) = \pi(\mathfrak{c})$, *and for every* r*-ary function* $g \in \mathfrak{F}$,

$$g(y_1, \ldots, y_r) = g\Big(\pi(y_1), \ldots, \pi(y_r)\Big) \quad \text{for all } y_1, \ldots, y_r \in \mathfrak{D}. \tag{6.3}$$

Proof. We start with the easier direction: If π exists, then $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ for all $\mathfrak{I}$. This is because for any injective n-tuple $(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) \in Y_n$, we can apply π to obtain a new injective n-tuple $(\mathfrak{a}, \mathfrak{c}, \pi(y_3), \ldots, \pi(y_n)) \in Y_n$, and this is a bijection mapping $(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) \in Y_n$ to $(\mathfrak{a}, \mathfrak{c}, z_3, \ldots, z_n) \in Y_n$. Moreover, by (6.3) we have

$$G(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) = G\big(\mathfrak{a}, \mathfrak{c}, \pi(y_3), \ldots, \pi(y_n)\big).$$

As a result, the two sums $\text{mon}_{\mathfrak{b}}(\mathfrak{I})$ and $\text{mon}_{\mathfrak{c}}(\mathfrak{I})$ over injective tuples must be equal.

The other direction is more difficult. First, we prove that if $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ for all $\mathfrak{I}$, then for any $\mathfrak{I}$ and any tuple $(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) \in Y_n$ with $G(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) > 0$, there exists a tuple $(\mathfrak{a}, \mathfrak{c}, z_3, \ldots, z_n) \in Y_n$ such that

$$G(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) = G(\mathfrak{a}, \mathfrak{c}, z_3, \ldots, z_n). \tag{6.4}$$

For this purpose we look at the sequence of instances $\mathfrak{J}_1 = \mathfrak{I}, \mathfrak{J}_2, \ldots$ defined from $\mathfrak{I}$, where $\mathfrak{J}_j$ consists of exactly j copies of $\mathfrak{I}$ over the same set of variables. We use G_j to denote the n-ary function that $\mathfrak{J}_j$ defines; then

$$G_j(y_1, \ldots, y_n) = \big(G(y_1, \ldots, y_n)\big)^j \quad \text{for all } y_1, \ldots, y_n \in \mathfrak{D}.$$

Let $Q = \{q_1, \ldots, q_{|Q|}\}$ denote the set of all possible positive values of G over Y_n; let $k_i \geq 0$ denote the number of tuples $(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) \in Y_n$ with $G(\mathfrak{a}, \mathfrak{b}, y_3, \ldots, y_n) = q_i$, $i \in [|Q|]$; and let $\ell_i \geq 0$ denote the number of tuples $(\mathfrak{a}, \mathfrak{c}, y_3, \ldots, y_n) \in Y_n$ such that $G(\mathfrak{a}, \mathfrak{c}, y_3, \ldots, y_n) = q_i$, $i \in [|Q|]$. Then by $\text{mon}_{\mathfrak{b}}(\mathfrak{J}_j) = \text{mon}_{\mathfrak{c}}(\mathfrak{J}_j)$, we have

$$\sum_{i \in [|Q|]} k_i \cdot (q_i)^j = \sum_{i \in [|Q|]} \ell_i \cdot (q_i)^j \quad \text{for all } j \geq 1.$$

Viewing $k_i - \ell_i$ as variables, the equations form a linear system with a Vandermonde matrix if we let j go from 1 to $|Q|$. As a result, we must have $k_i = \ell_i$ for all $i \in [|Q|]$, and (6.4) follows.

To finish the proof, we need the following technical lemma.

Lemma 6.7. *Let Q be a finite and nonempty set of positive numbers. Then for any $k \geq 1$, there exists a sequence of positive integers $N_1, \ldots, N_k$ such that*

$$q_1^{N_1} q_2^{N_2} \cdots q_k^{N_k} = (q_1')^{N_1}(q_2')^{N_2} \cdots (q_k')^{N_k}, \quad \text{where } q_1, \ldots, q_k, q_1' \ldots, q_k' \in Q, \tag{6.5}$$

if and only if $q_i = q_i'$ for every $i \in [k]$.

Proof. The lemma is trivial if $|Q| = 1$, so we assume $|Q| \geq 2$. We use induction on k. The basis is trivial: We just set $N_1 = 1$. Now we assume the lemma holds for some $k \geq 1$, and $N_1, \ldots, N_k$ is the sequence for k. We show how to find N_{k+1} so that $N_1, \ldots, N_{k+1}$ satisfies the lemma for $k + 1$. To this end, we let

$$c_{\min} = \min_{q > q' \in Q} q/q' > 1 \quad \text{and} \quad c_{\max} = \max_{q > q' \in Q} q/q'.$$

Then we let N_{k+1} be a large enough integer such that

$$\big(c_{\min}\big)^{N_{k+1}} > \big(c_{\max}\big)^{\sum_{i \in [k]} N_i}.$$

To prove the correctness we assume (6.5) holds. First we must have $q_{k+1} = q_{k+1}'$. Otherwise, assume without generality that $q_{k+1} > q_{k+1}'$; then by (6.5)

$$\big(c_{\min}\big)^{N_{k+1}} \leq \big(q_{k+1}/q_{k+1}'\big)^{N_{k+1}} = \big(q_1'/q_1\big)^{N_1} \cdots \big(q_k'/q_k\big)^{N_k} \leq \big(c_{\max}\big)^{\sum_{i \in [k]} N_k},$$

which contradicts the definition of N_{k+1}. Once we have $q_{k+1} = q'_{k+1}$, they can be removed from (6.5), and by the inductive hypothesis, we have $q_i = q'_i$ for all $i \in [k]$. This finishes the induction, and the lemma is proved. ∎

To find π, we define the following $\mathfrak{I}$: Let $\mathfrak{s}_1, \ldots, \mathfrak{s}_{|\mathfrak{D}|}$ be an arbitrary enumeration of the domain set $\mathfrak{D}$, with $\mathfrak{s}_1 = \mathfrak{a}$ and $\mathfrak{s}_2 = \mathfrak{b}$. There are $|\mathfrak{D}|$ variables in $\mathfrak{I}$ indexed by $[|\mathfrak{D}|]$. Let L be the set of all tuples $(g, i_1, \ldots, i_r)$ such that g is an r-ary function in $\mathfrak{F}$ and $g(\mathfrak{s}_{i_1}, \ldots, \mathfrak{s}_{i_r}) > 0$. We let $N_1, \ldots, N_{|L|}$ be the sequence of positive integers that satisfies Lemma 6.7 with $k = |L|$ and

$$Q = \Big\{ g(\mathfrak{s}_{i_1}, \ldots, \mathfrak{s}_{i_r}) : (g, i_1, \ldots, i_r) \in L \Big\}.$$

Then we enumerate all tuples in L in any order. For the ith tuple $(g, i_1, \ldots, i_r) \in L$, $i \in [|L|]$, we add N_i copies of the same tuple $(g, i_1, \ldots, i_r)$ to $\mathfrak{I}$, and this finishes the definition of $\mathfrak{I}$.

By the definition of $\mathfrak{I}$, we have $G(\mathfrak{a}, \mathfrak{b}, \mathfrak{s}_3, \ldots, \mathfrak{s}_{|\mathfrak{D}|}) = G(\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3, \ldots, \mathfrak{s}_{|\mathfrak{D}|}) > 0$. So by (6.4) we know there exists a tuple $(\mathfrak{t}_i : i \in [|\mathfrak{D}|]) \in Y_n$ such that $\mathfrak{t}_1 = \mathfrak{a}$, $\mathfrak{t}_2 = \mathfrak{c}$, and

$$G\big(\mathfrak{a}, \mathfrak{b}, \mathfrak{s}_3, \ldots, \mathfrak{s}_{|\mathfrak{D}|}\big) = G\big(\mathfrak{a}, \mathfrak{c}, \mathfrak{t}_3, \ldots, \mathfrak{t}_{|\mathfrak{D}|}\big) > 0.$$

Let π be the map from $\mathfrak{D}$ to $\mathfrak{D}$ with $\pi(\mathfrak{s}_i) = \mathfrak{t}_i$ for each $i \in |\mathfrak{D}|$. We show below that π is the bijection that we are looking for. (Note that π is a bijection because $(\mathfrak{t}_i) \in Y_n$, and thus $(\mathfrak{t}_i)$ must be a permutation of the domain set $\mathfrak{D}$.)

First, using Lemma 6.7, it follows from the definition of $\mathfrak{I}$ that

$$g\big(\mathfrak{s}_{i_1}, \ldots, \mathfrak{s}_{i_r}\big) = g\big(\pi(\mathfrak{s}_{i_1}), \ldots, \pi(\mathfrak{s}_{i_r})\big) \quad \text{for every tuple } (g, i_1, \ldots, i_r) \in L.$$

Hence it suffices to show that $g(\pi(\mathfrak{s}_{i_1}), \ldots, \pi(\mathfrak{s}_{i_r})) = 0$ when $g(\mathfrak{s}_{i_1}, \ldots, \mathfrak{s}_{i_r}) = 0$. This follows from the fact that π is a bijection, and thus $(\mathfrak{s}_{i_1}, \ldots, \mathfrak{s}_{i_r}) \to (\pi(\mathfrak{s}_{i_1}), \ldots, \pi(\mathfrak{s}_{i_r}))$ is also a bijection. ∎

Given Lemmas 6.5 and 6.6, it suffices to check whether there is an automorphism π of $(\mathfrak{D}, \mathfrak{F})$ such that $\pi(\mathfrak{a}) = \mathfrak{a}$ and $\pi(\mathfrak{b}) = \mathfrak{c}$. We can nondeterministically check all possible bijections from $\mathfrak{D}$ to itself, which gives us the membership in NP.

7. Proof of Lemma 2.4. We restate Lemma 2.4 and prove it in this section.

LEMMA 2.4. *Problem* #CSP(Γ) *is polynomial-time reducible to* #CSP($\mathcal{F}$).

Recall that $\mathcal{F} = \{f_1, \ldots, f_h\}$ and $\Gamma = \{\Theta_1, \ldots, \Theta_h\}$, with r_i being the arity of f_i. Let I be an instance of #CSP(Γ) with n variables indexed by $[n]$ and m constraints, and let R be the relation it defines.

For each $k \geq 1$, we use I_k to denote the following instance of #CSP($\mathcal{F}$): I_k has n variables indexed by $[n]$ and for each constraint $(\Theta, i_1, \ldots, i_r) \in I$, we add k copies of $(f, i_1, \ldots, i_r)$ to I_k, where $f \in \mathcal{F}$ is the r-ary function that corresponds to $\Theta \in \Gamma$. We use $F_k(\mathbf{x})$ to denote the n-ary nonnegative function that I_k defines. Then we have

$$F_k(\mathbf{x}) = \Big(F_1(\mathbf{x})\Big)^k \quad \text{for all } \mathbf{x} \in D^n. \tag{7.1}$$

We show below that to compute $|R|$ it suffices to evaluate $Z(I_k)$ for k from 1 to some polynomial of m. This gives the desired reduction from #CSP(Γ) to #CSP($\mathcal{F}$).

To this end, we let Q_m denote the set of all integer tuples

$$\mathbf{q} = \Big(q_{i,\mathbf{t}} \geq 0 : i \in [h] \text{ and } \mathbf{t} \in D^{r_i} \text{ such that } f_i(\mathbf{t}) > 0\Big)$$

that sum to m. Then we let $\textsc{Value}_m$ denote the following set of positive numbers:

$$\textsc{Value}_m = \left\{ \prod_{i\in[h],\,\mathbf{t}\in D^{r_i}} \Big(f_i(\mathbf{t})\Big)^{q_{i,\mathbf{t}}} : \mathbf{q} \in Q_m \right\}.$$

It is easy to show that both $|Q_m|$ and $|\textsc{Value}_m|$ are polynomial in m (as $d = |D|$, h, and r_i, $i \in [h]$, are all constants) and can be computed in polynomial time in m. Moreover, by the definition of $\textsc{Value}_m$ we have, for every $\mathbf{x} \in D^n$,

$$F_1(\mathbf{x}) > 0 \implies F_1(\mathbf{x}) \in \textsc{Value}_m.$$

For each $c \in \textsc{Value}_m$, let N_c be the number of $\mathbf{x} \in D^n$ such that $F_1(\mathbf{x}) = c$. Then

$$Z(I_1) = \sum_{c\in\textsc{Value}_m} N_c \cdot c. \tag{7.2}$$

We also have

$$\big|R\big| = \sum_{c\in\textsc{Value}_m} N_c, \tag{7.3}$$

and by (7.1)

$$Z(I_k) = \sum_{c\in\textsc{Value}_m} N_c \cdot c^k \quad \text{for every } k \ge 1. \tag{7.4}$$

By viewing $\{N_c : c \in \textsc{Value}_m\}$ as variables and taking k from 1 to $|\textsc{Value}_m|$, (7.4) gives us a Vandermonde system from which we can compute N_c for each $c \in \textsc{Value}_m$ in polynomial time. We can then use (7.3) to compute $|R|$.

This finishes the proof of Lemma 2.4. ∎

8. Proof of Lemma 3.4. We restate Lemma 3.4 and prove it in this section.

LEMMA 3.4. *If $\mathcal{F}$ is not balanced, then* #CSP($\mathcal{F}$) *is* #P-*hard.*

Assume that $\mathcal{F}$ is not balanced. Then there must exist an instance I of #CSP($\mathcal{F}$) which defines an n-ary function $F(x_1, \ldots, x_n)$ and integers $a, b : 1 \le a < b \le n$ such that the following $d^a \times d^{b-a}$ matrix $\mathbf{M}$ is not block-rank-1: The rows are indexed by $\mathbf{u} \in D^a$, the columns are indexed by $\mathbf{v} \in D^{b-a}$, and

$$M(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{w}\in D^{n-b}} F(\mathbf{u}, \mathbf{v}, \mathbf{w}) \quad \text{for all } \mathbf{u} \in D^a \text{ and } \mathbf{v} \in D^{b-a}.$$

Let $\mathbf{A} = \mathbf{M}\mathbf{M}^{\mathrm{T}}$, which is a symmetric, nonnegative $d^a \times d^a$ matrix, with both its rows and columns indexed by tuples $\mathbf{u} \in D^a$. As $\mathbf{M}$ is not block-rank-1, it follows from Lemma 2.2 that $\mathbf{A}$ is not block-rank-1.

To finish the proof we give a polynomial-time reduction from $Z_{\mathbf{A}}(\cdot)$ to #CSP($\mathcal{F}$). As the former is #P-hard by Theorem 2.3 (since $\mathbf{A}$ is not block-rank-1), we have that #CSP($\mathcal{F}$) is also #P-hard.

Let $G = (V, E)$ be an input undirected graph of $Z_{\mathbf{A}}(\cdot)$. We construct an instance I_G of #CSP($\mathcal{F}$) from G using I (which is considered as a constant here since it does not depend on G) as follows:

1. For each $v \in V$, we create a variables over D, denoted by $x_{v,1}, \ldots, x_{v,a}$.

2. For each $e = vv' \in E$, we add $(b-a)+2(n-b)$ variables over D, denoted by

$$y_{e,a+1},\ldots,y_{e,b},z_{e,b+1},\ldots,z_{e,n},z'_{e,b+1},\ldots,z'_{e,n}.$$

Then we make a copy of I over the n variables

$$\big(x_{v,1},\ldots,x_{v,a},y_{e,a+1},\ldots,y_{e,b},z_{e,b+1},\ldots,z_{e,n}\big),$$

as well as the n variables

$$\big(x_{v',1},\ldots,x_{v',a},y_{e,a+1},\ldots,y_{e,b},z'_{e,b+1},\ldots,z'_{e,n}\big).$$

This finishes the construction of I_G.

It is easy to show that $Z_{\mathbf{A}}(G) = Z(I_G)$. This gives a polynomial-time reduction from $Z_{\mathbf{A}}(\cdot)$ to $\#\mathrm{CSP}(\mathcal{F})$ since I_G can be constructed from G in polynomial time. ∎

9. Equivalence of balance and strong balance. In [27], Dyer and Richerby used the following notion of strong balance for unweighted constraint languages, and showed that $\#\mathrm{CSP}(\Gamma)$ is in polynomial time if Γ is strongly balanced, and is #P-hard otherwise.

DEFINITION 9.1. *Let Γ be an unweighted constraint language over the domain set D. We call Γ* strongly balanced *if, for every instance I of $\#\mathrm{CSP}(\Gamma)$ which defines an n-ary relation R and any $a,b,c : 1 \le a < b \le c \le n$, the following $d^a \times d^{b-a}$ matrix $\mathbf{M}$ is block-rank-1: The rows and columns are indexed by $\mathbf{u} \in D^a$ and $\mathbf{v} \in D^{b-a}$, and*

$$M(\mathbf{u},\mathbf{v}) = \Big|\big\{\mathbf{w} \in D^{c-b} : \exists\, \mathbf{z} \in D^{n-c} \text{ such that } (\mathbf{u},\mathbf{v},\mathbf{w},\mathbf{z}) \in R\big\}\Big| \tag{9.1}$$

for all $\mathbf{u} \in D^a$ and $\mathbf{v} \in D^{b-a}$. There are two special cases. When $c = b$, $M(\mathbf{u},\mathbf{v})$ is 1 if there exists a $\mathbf{z} \in D^{n-c}$ with $(\mathbf{u},\mathbf{v},\mathbf{z}) \in R$, and is 0 otherwise. When $n = c$, $M(\mathbf{u},\mathbf{v})$ is the number of $\mathbf{w} \in D^{c-b}$ such that $(\mathbf{u},\mathbf{v},\mathbf{w}) \in R$.

THEOREM 9.2 (see [27]). *$\#\mathrm{CSP}(\Gamma)$ is in polynomial time if Γ is strongly balanced, and is #P-hard otherwise.*

A key difference between the notion of balance we used for weighted languages $\mathcal{F}$ (Definition 3.2) and the one above for unweighted languages Γ [27] is that we do not allow the use of existential quantifiers in the former. One can similarly introduce the following notion of balance for unweighted languages.

DEFINITION 9.3. *Let Γ be an unweighted constraint language over the domain set D. We say Γ is* balanced *if for every instance I of $\#\mathrm{CSP}(\Gamma)$ which defines an n-ary relation R and any $a,b : 1 \le a < b \le n$, the following $d^a \times d^{b-a}$ matrix $\mathbf{M}$ is block-rank-1: The rows are indexed by $\mathbf{u} \in D^a$, the columns are indexed by $\mathbf{v} \in D^{b-a}$, and*

$$M(\mathbf{u},\mathbf{v}) = \Big|\big\{\mathbf{w} \in D^{n-b} : (\mathbf{u},\mathbf{v},\mathbf{w}) \in R\big\}\Big| \quad \text{for all } \mathbf{u} \in D^a, \mathbf{v} \in D^{b-a}. \tag{9.2}$$

It is clear from the definitions that strong balance implies balance for unweighted languages Γ. We show below that these two notions are indeed equivalent.

LEMMA 9.4 (equivalence of balance and strong balance). *If Γ is balanced, then it is also strongly balanced.*

Proof. We assume that Γ is balanced. Let I be any instance of $\#\mathrm{CSP}(\Gamma)$ which defines an n-ary relation R. Let a, b, and c be integers such that $1 \le a < b \le c \le n$. It suffices to show that the matrix $\mathbf{M}$ in (9.1) is block-rank-1.

For this purpose, we define a new input instance I_k of $\#\mathrm{CSP}(\Gamma)$ for each $k \ge 1$:

1. First, I_k has $c + k(n - c)$ variables in the following order:

$$x_1, \dots, x_c, y_{1,c+1}, \dots, y_{1,n}, \dots, y_{k,c+1}, \dots, y_{k,n}.$$

Below we let $\mathbf{y}_i$, $i \in [k]$, denote $(y_{i,c+1}, \dots, y_{i,n})$ for convenience.

2. For each $i \in [k]$, we add a copy of I on the following n variables of I_k:

$$x_1, \dots, x_c, y_{i,c+1}, \dots, y_{i,n}.$$

It is clear that I_1 is I (up to renaming of variables). We use R_k to denote the relation that I_k defines, $k \geq 1$.

As Γ is balanced, the following $d^a \times d^{b-a}$ matrix $\mathbf{M}^{[k]}$ is block-rank-1: For $\mathbf{u} \in D^a$ and $\mathbf{v} \in D^{b-a}$, we have

$$M^{[k]}(\mathbf{u}, \mathbf{v}) = \Big|\Big\{(\mathbf{w}, \mathbf{y}_1, \dots, \mathbf{y}_k) : \mathbf{w} \in D^{c-b}, \mathbf{y}_1, \dots, \mathbf{y}_k \in D^{n-c} \text{ and } (\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{y}_1, \dots, \mathbf{y}_k) \in R_k\Big\}\Big|.$$

From the definition of I_k, we have $M(\mathbf{u}, \mathbf{v}) > 0$ if and only if $M^{[k]}(\mathbf{u}, \mathbf{v}) > 0$ for all $\mathbf{u} \in D^a$ and $\mathbf{v} \in D^{b-a}$.

As a result, there exist pairwise disjoint and nonempty subsets $A_1, \dots, A_s$ of D^a and pairwise disjoint and nonempty subsets $B_1, \dots, B_s$ of D^{b-a} for some $s \geq 0$, with

$$M(\mathbf{u}, \mathbf{v}) > 0 \iff M^{[k]}(\mathbf{u}, \mathbf{v}) > 0 \iff \mathbf{u} \in A_\ell \text{ and } \mathbf{v} \in B_\ell \text{ for some } \ell \in [s].$$

Now to prove that $\mathbf{M}$ is block-rank-1, we need only show that for every $\ell \in [s]$,

$$M(\mathbf{u}_1, \mathbf{v}_1) \cdot M(\mathbf{u}_2, \mathbf{v}_2) = M(\mathbf{u}_1, \mathbf{v}_2) \cdot M(\mathbf{u}_2, \mathbf{v}_1) \quad \text{for } \mathbf{u}_1, \mathbf{u}_2 \in A_\ell,\ \mathbf{v}_1, \mathbf{v}_2 \in B_\ell. \tag{9.3}$$

To prove (9.3), we let

$$W_{i,j} = \Big\{\mathbf{w} \in D^{c-b} : \exists\, \mathbf{y} \in D^{n-c} \text{ such that } (\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}, \mathbf{y}) \in R\Big\} \quad \text{for } i, j \in \{1, 2\}.$$

Moreover, for each $\mathbf{w} \in W_{i,j}$, let $Y_{i,j,\mathbf{w}}$ denote the (nonempty) set of $\mathbf{y} \in D^{n-c}$ such that $(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}, \mathbf{y}) \in R$. Using $W_{i,j}$ and $Y_{i,j,\mathbf{w}}$, it follows from the definition of I_k that

$$M^{[k]}(\mathbf{u}_i, \mathbf{v}_j) = \sum_{\mathbf{w} \in W_{i,j}} \Big|Y_{i,j,\mathbf{w}}\Big|^k.$$

Because $\mathbf{M}^{[k]}$ is block-rank-1, we have the following equation for every $k \geq 1$:

$$\sum_{\mathbf{w} \in W_{1,1}, \mathbf{w}' \in W_{2,2}} \Big(\big|Y_{1,1,\mathbf{w}}\big| \cdot \big|Y_{2,2,\mathbf{w}'}\big| \Big)^k = \sum_{\mathbf{w} \in W_{1,2}, \mathbf{w}' \in W_{2,1}} \Big(\big|Y_{1,2,\mathbf{w}}\big| \cdot \big|Y_{2,1,\mathbf{w}'}\big| \Big)^k.$$

Because the equation above holds for every $k \geq 1$, the two sides must have the same number of positive terms. As a consequence of the definition, $Y_{i,j,\mathbf{w}}$ is nonempty for all $\mathbf{w} \in W_{i,j}$. As a result, we have $|W_{1,1}| \cdot |W_{2,2}| = |W_{1,2}| \cdot |W_{2,1}|$, and (9.3) follows.

This completes the proof of Lemma 9.4. ∎

REFERENCES

[1] P. Austrin and E. Mossel, *Approximation resistant predicates from pairwise independence*, Comput. Complexity, 18 (2009), pp. 249–271, https://doi.org/10.1007/s00037-009-0272-6.

[2] A. Bulatov and V. Dalmau, *A simple algorithm for Mal'tsev constraints*, SIAM J. Comput., 36 (2006), pp. 16–27, https://doi.org/10.1137/050628957.

[3] A. Bulatov, M. Dyer, L. A. Goldberg, M. Jalsenius, M. Jerrum, and D. Richerby, *The complexity of weighted and unweighted #CSP*, J. Comput. System Sci., 78 (2012), pp. 681–688, https://doi.org/10.1016/j.jcss.2011.12.002.

[4] A. Bulatov and M. Grohe, *The complexity of partition functions*, Theoret. Comput. Sci., 348 (2005), pp. 148–186, https://doi.org/10.1016/j.tcs.2005.09.011.

[5] A. A. Bulatov, *Tractable conservative constraint satisfaction problems*, in Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science, 2003, pp. 321–330, https://doi.org/10.1109/LICS.2003.1210072.

[6] A. A. Bulatov, *A dichotomy theorem for constraints on a 3-element set*, J. ACM, 53 (2006), pp. 66–120, https://doi.org/10.1145/1120582.1120584.

[7] A. A. Bulatov, *The complexity of the counting constraint satisfaction problem*, J. ACM, 60 (2013), 34, https://doi.org/10.1145/2528400.

[8] A. A. Bulatov and V. Dalmau, *Towards a dichotomy theorem for the counting constraint satisfaction problem*, Inform. and Comput., 205 (2007), pp. 651–678, https://doi.org/10.1016/j.ic.2006.09.005.

[9] A. A. Bulatov and P. Jeavons, *An algebraic approach to multi-sorted constraints*, in Principles and Practice of Constraint Programming – CP 2003, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 183–198, https://doi.org/10.1007/978-3-540-45193-8_13.

[10] A. A. Bulatov and M. A. Valeriote, *Recent results on the algebraic approach to the CSP*, in Complexity of Constraints, Lecture Notes in Comput. Sci. 5250, N. Creignou, P. G. Kolaitis, and H. Vollmer, eds., Springer-Verlag, Berlin, Heidelberg, 2008, pp. 68–92, https://doi.org/10.1007/978-3-540-92800-3_4.

[11] S. Burris and H. P. Sankappanavar, *A Course in Universal Algebra*, Grad. Texts in Math. 78, Springer-Verlag, New York, Berlin, 1981.

[12] J.-Y. Cai and X. Chen, *A decidable dichotomy theorem on directed graph homomorphisms with non-negative weights*, in Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 437–446, https://doi.org/10.1109/FOCS.2010.49.

[13] J.-Y. Cai and X. Chen, *Complexity of counting CSP with complex weights*, in Proceedings of the 44th Annual ACM Symposium on Theory of Computing, 2012, pp. 909–920, https://doi.org/10.1145/2213977.2214059.

[14] J.-Y. Cai, X. Chen, and P. Lu, *Nonnegatively weighted #CSPs: An effective complexity dichotomy*, in Proceedings of the 25th IEEE Conference on Computational Complexity, 2011, pp. 45–54.

[15] J.-Y. Cai, X. Chen, and P. Lu, *Graph homomorphisms with complex values: A dichotomy theorem*, SIAM J. Comput., 42 (2013), pp. 924–1029, https://doi.org/10.1137/110840194.

[16] J.-Y. Cai, Z. Fu, H. Guo, and T. Williams, *A Holant dichotomy: Is the FKT algorithm universal?*, in Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, 2015, pp. 1259–1276, https://doi.org/10.1109/FOCS.2015.81.

[17] J.-Y. Cai, H. Guo, and T. Williams, *A complete dichotomy rises from the capture of vanishing signatures*, SIAM J. Comput., 45 (2016), pp. 1671–1728, https://doi.org/10.1137/15M1049798.

[18] J.-Y. Cai, H. Guo, and T. Williams, *The complexity of counting edge colorings and a dichotomy for some higher domain Holant problems*, in Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, 2014, pp. 601–610, https://doi.org/10.1109/FOCS.2014.70.

[19] J.-Y. Cai, S. Huang, and P. Lu, *From Holant to #CSP and back: Dichotomy for Holant^c problems*, Algorithmica, 64 (2012), pp. 511–533, https://doi.org/10.1007/s00453-012-9626-6.

[20] J.-Y. Cai, P. Lu, and M. Xia, *Holant problems and counting CSP*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009, pp. 715–724, https://doi.org/10.1145/1536414.1536511.

[21] J.-Y. Cai, P. Lu, and M. Xia, *Holographic algorithms with matchgates capture precisely tractable planar #CSP*, in Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 427–436, https://doi.org/10.1109/FOCS.2010.48.

[22] J.-Y. Cai, P. Lu, and M. Xia, *Dichotomy for Holant* problems of Boolean domain*, in Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, 2011, pp. 1714–1728, https://doi.org/10.1137/1.9781611973082.132.

[23] J.-Y. Cai, P. Lu, and M. Xia, *Dichotomy for Holant* problems with a function on domain size* 3, in Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, 2013, pp. 1278–1295, https://doi.org/10.1137/1.9781611973105.93.

[24] I. Dinur, E. Mossel, and O. Regev, *Conditional hardness for approximate coloring*, SIAM J. Comput., 39 (2009), pp. 843–873, https://doi.org/10.1137/07068062X.

[25] M. Dyer, L. A. Goldberg, and M. Paterson, *On counting homomorphisms to directed acyclic graphs*, J. ACM, 54 (2007), 27, https://doi.org/10.1145/1314690.1314691.

[26] M. Dyer and C. Greenhill, *The complexity of counting graph homomorphisms*, Random Structures Algorithms, 17 (2000), pp. 260–289, https://doi.org/10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W.

[27] M. Dyer and D. Richerby, *An effective dichotomy for the counting constraint satisfaction problem*, SIAM J. Comput., 42 (2013), pp. 1245–1274, https://doi.org/10.1137/100811258.

[28] T. Feder and M. Y. Vardi, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory*, SIAM J. Comput., 28 (1998), pp. 57–104, https://doi.org/10.1137/S0097539794266766.

[29] R. Freese and R. McKenzie, *Commutator Theory for Congruence Modular Varieties*, Cambridge University Press, Cambridge, UK, 1987.

[30] L. A. Goldberg, M. Grohe, M. Jerrum, and M. Thurley, *A complexity dichotomy for partition functions with mixed signs*, SIAM J. Comput., 39 (2010), pp. 3336–3402, https://doi.org/10.1137/090757496.

[31] J. Håstad, *Some optimal inapproximability results*, J. ACM, 48 (2001), pp. 798–859, https://doi.org/10.1145/502090.502098.

[32] P. Hell and J. Nešetřil, *On the complexity of H-coloring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110, https://doi.org/10.1016/0095-8956(90)90132-J.

[33] D. Hobby and R. McKenzie, *The Structure of Finite Algebras*, Contemp. Math. 76, American Mathematical Society, Providence, RI, 1988, https://doi.org/10.1090/conm/076.

[34] P. G. Jeavons, *On the algebraic structure of combinatorial problems*, Theoret. Comput. Sci., 200 (1998), pp. 185–204, https://doi.org/10.1016/S0304-3975(97)00230-2.

[35] P. G. Jeavons, D. A. Cohen, and M. C. Cooper, *Constraints, consistency and closure*, Artificial Intelligence, 101 (1998), pp. 251–265, https://doi.org/10.1016/S0004-3702(98)00022-8.

[36] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, *Optimal inapproximability results for MAX-CUT and other* 2*-variable CSPs?*, SIAM J. Comput., 37 (2007), pp. 319–357, https://doi.org/10.1137/S0097539705447372.

[37] G. Kun and M. Szegedy, *A new line of attack on the dichotomy conjecture*, European. J. Combin., 52 (2016), pp. 338–367, https://doi.org/10.1016/j.ejc.2015.07.011.

[38] H. W. Lenstra, Jr., *Algorithms in algebraic number theory*, Bull. Amer. Math. Soc. (N.S.), 26 (1992), pp. 211–244, https://doi.org/10.1090/S0273-0979-1992-00284-7.

[39] L. Lovász, *Operations with structures*, Acta Math. Acad. Sci. Hungar., 18 (1967), pp. 321–328.

[40] P. Raghavendra, *Optimal algorithms and inapproximability results for every CSP?*, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008, pp. 245–254, https://doi.org/10.1145/1374376.1374414.

[41] P. Raghavendra and D. Steurer, *How to round any CSP*, in Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, 2009, pp. 586–594, https://doi.org/10.1109/FOCS.2009.74.

[42] T. J. Schaefer, *The complexity of satisfiability problems*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226, https://doi.org/10.1145/800133.804350.

[43] M. Thurley, *The Complexity of Partition Functions on Hermitian Matrices*, preprint, https://arxiv.org/abs/1004.0992, 2010.

[44] M. Tulsiani, *CSP gaps and reductions in the Lasserre hierarchy*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009, pp. 303–312, https://doi.org/10.1145/1536414.1536457.

[45] L. G. Valiant, *Holographic algorithms*, SIAM J. Comput., 37 (2008), pp. 1565–1594, https://doi.org/10.1137/070682575.