Holant Problems and Counting CSP

Jin-Yi Cai*
Computer Sciences Department
University of Wisconsin
Madison, WI 53706. USA
jyc@cs.wisc.edu

Pinyan Lu †
Microsoft Research Asia
Beijing, 100190, P.R. China
lupinyan@gmail.com

Mingji Xia‡
University of Wisconsin
Madison, WI 53706. USA
and Institute of Software, CAS
Beijing, 100190, P. R. China
xmjljx@gmail.com

ABSTRACT

We propose and explore a novel alternative framework to study the complexity of counting problems, called Holant Problems. Compared to counting Constrained Satisfaction Problems (#CSP), it is a refinement with a more explicit role for the function constraints. Both graph homomorphism and #CSP can be viewed as special cases of Holant Problems. We prove complexity dichotomy theorems in this framework. Because the framework is more stringent, previous dichotomy theorems for #CSP problems no longer apply. Indeed, we discover surprising tractable subclasses of counting problems, which could not have been easily specified in the #CSP framework. The main technical tool we use and develop is holographic reductions. Another technical tool used in combination with holographic reductions is polynomial interpolations. The study of Holant Problems led us to discover and prove a complexity dichotomy theorem for the most general form of Boolean #CSP where every constraint function takes values in the complex number field $\mathbb{C}$.

Categories and Subject Descriptors

F.2.0 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory

Keywords

Holant problem, #CSP, holographic reduction, polynomial interpolation

*Supported by NSF CCF-0830488 and CCF-0511679.

†Work done in part while the author was a graduate student at Tsinghua University.

‡Supported by Hundred Talent Program of Chinese Academy of Sciences under Angsheng Li.

STOC'09, May 31–June 2, 2009, Bethesda, Maryland, USA.

1. INTRODUCTION

In order to study the complexity of counting problems, several interesting frameworks have been proposed. One is called counting Constrained Satisfaction Problems (#CSP) [1, 2, 3, 13, 18]. Another well studied framework is called Graph Homomorphisms or H-coloring problems, which can be viewed as a special case of #CSP problems [4, 5, 6, 14, 15, 16, 17, 20, 21]. One reason such frameworks are interesting is because the language is *expressive* enough so that they can express many natural counting problems, while *specific* enough so that we can prove complete complexity classifications for them [10]. The natural counting problems which can be expressed as graph homomorphism problems include counting the number of vertex covers, the number of k-colorings in a graph, and many others. However, there are some natural and important counting problems, which can not be expressed as a graph homomorphism problem. In [19], it is proved that counting the number of perfect matchings in a graph *cannot* be expressed as a graph homomorphism function. Additionally, sometimes a problem can be expressed in the existing framework, such as #CSP, but only with some contrived restrictions.

In this paper, we propose and explore an alternative framework to study the complexity of counting problems, called Holant Problems. This notion is motivated by holographic reductions proposed by Valiant [27, 28]. Compared to #CSP, it is a refinement with a more explicit role for the function constraints. Both graph homomorphism and #CSP can be viewed as special cases of Holant Problems. We give a brief description here and a more formal definition is given in Section 2. A *signature grid* $\Omega = (G, \mathcal{F}, \pi)$ is a tuple, where $G = (V, E)$ is a graph, and π maps each $v \in V(G)$ to a function $f_v \in \mathcal{F}$. We consider all edge assignments. An assignment σ for every $e \in E$ gives an evaluation $\prod_{v\in V} f_v(\sigma \mid_{E(v)})$, where $E(v)$ denotes the incident edges of v. The counting problem on the instance Ω is to compute

$$\text{Holant}_\Omega = \sum_\sigma \prod_{v\in V} f_v(\sigma \mid_{E(v)}).$$

We use the notation Holant($\mathcal{F}$) to denote the class of Holant problems where all functions are given by $\mathcal{F}$. For example, consider the Perfect Matching problem on G. This problem corresponds to attaching the Exact-One function at every vertex of G, and then consider all 0-1 edge assignments. In this case, Holant_Ω counts the number of perfect matchings. If we use the At-Most-One function at every vertex, then we are counting all (not necessarily perfect) matchings. So this new framework can express some natural counting problems which are not expressible as graph homomorphisms.

To see that Holant is a more expressive framework, we show that every #CSP problem can be simulated by a Holant problem. Represent an instance of a #CSP problem by a bipartite graph where LHS are labeled by variables and RHS are labeled by constraints. Now the signature grid Ω on this bipartite graph is as follows: Every variable node on LHS is attached an EQUALITY function, every constraint node on RHS has the given constraint function. Then Holant_Ω is exactly the answer to the counting CSP problem. In effect, the EQUALITY function on each variable node forces the incident edges to take the same value; this effectively reduces to assigning values to each variable on LHS as in #CSP. It follows that #CSP problems are precisely the special case of Holant problems on bipartite graphs where every node on LHS is attached an EQUALITY function. We can show that the class of #CSP problems is equivalent to Holant problems where all EQUALITY functions are always assumed to be freely available, and implicitly so. Graph homomorphism is a further special case where not only all EQUALITY functions are freely (and implicitly) available, but the function set $\mathcal{F}$ in our signature grid Ω contains exactly one binary function (other than these EQUALITY functions). It turns out that allowing EQUALITY functions has a major influence on the computational complexity of the problems. By making the presence of these EQUALITY functions explicit, the Holant framework of counting problems can make a finer complexity classification, which is difficult to do in #CSP.

Our Holant Problem framework is strongly influenced by the development of holographic algorithms and holographic reductions [27, 28, 7, 9]. Indeed, we will use and develop holographic reductions here as a primary technique. One advantage of our new framework is that one can naturally consider new subclasses of counting problems as special cases of Holant problems other than #CSP problems. For instance, by assuming all unary functions are freely available, we propose an interesting counting problem family called Holant^* Problems. Our first main result is a complexity dichotomy theorem for all Holant^* Problems for arbitrary complex valued symmetric functions over Boolean variables: Each problem in the class is either #P-hard or solvable in P. In this dichotomy theorem, most tractable cases are accomplished by holographic algorithms with Fibonacci gates [9]. And what is more interesting and surprising is that the key technique used in the hardness proof is also holographic reductions. Furthermore, we prove that the theorem holds for planar graphs.

Our second main result is a dichotomy theorem for an even more appealing family of counting problems, called Holant^c Problems, where we only assume two special unary functions Δ_0 and Δ_1 are available. These two unary functions simply set a particular edge (variable) to a constant value 0 and 1. We can prove again that every problem in the class is either #P-hard or solvable in P. However here we can only prove it for all real valued symmetric functions over Boolean variables. (We conjecture that it is still true over $\mathbb{C}$.) Note that when we assume fewer functions are freely available in the framework it makes the specification of the family more stringent. It delineates more precisely what functions and in what combinations lead to #P-hardness, or to tractability, respectively. The fewer functions are assumed free, the more tractable cases there might be. It turns out that this is indeed the case. In addition to the tractable cases as in Holant^* Problems, we discover that the following three families of functions are tractable. (We list the functions by their truth tables, and where $i = \sqrt{-1}$, $\lambda \in \mathbb{C}$, $k = 1, 2, \ldots$, and $r = 0, 1, 2, 3$.)

$$\begin{aligned}\mathcal{F}_1 &= \{\lambda([1,0]^{\otimes k} + i^r[0,1]^{\otimes k})\};\\ \mathcal{F}_2 &= \{\lambda([1,1]^{\otimes k} + i^r[1,-1]^{\otimes k})\};\\ \mathcal{F}_3 &= \{\lambda([1,\ i]^{\otimes k} + i^r[1,-i]^{\otimes k})\}.\end{aligned}$$

We prove that $\text{Holant}^c(\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3)$ is polynomial time computable. The tractability crucially depends on algebraic cancelations.

However, the fewer functions are assumed free, the more challenging it is to prove #P-hardness. The main technique for the proof of the second dichotomy theorem is polynomial interpolations. We make *essential* use of the dichotomy theorem just proved for Holant^* Problems, as a launching station to prove our dichotomy theorem for Holant^c Problems. Once we can interpolate all the unary functions, we can apply the result for Holant^* Problems.

The Holant^c Problems are basically generic Holant Problems with the ability to fix the assignments of some edges. In many natural counting problems, this is indeed the case, such as counting problems for perfect matchings. By the Pinning Lemma in [13], in any #CSP problem, Δ_0, Δ_1 can be simulated, and as a result can be viewed as freely available. In other words EQUALITY functions are stronger than Δ_0 and Δ_1. Therefore Holant^c Problems already subsume #CSP, and in the meanwhile provide a way for a more exacting account of what makes a problem tractable or #P-hard. Our dichotomy theorems have already paid dividend in the study of classifications of #CSP problems. Since #CSP can be viewed as a special case of Holant^c Problems, the dichotomy theorem for Holant^c Problems automatically implies a dichotomy theorem for Boolean #CSP problems with real symmetric constraints. Motivated by this, we investigated how one might generalize the tractable cases $(\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3)$ to unsymmetric ones. Surprisingly it turns out that the symmetric tractable cases already supplied the essential ingredients for all possible (including unsymmetric) tractable ones. This led us to a dichotomy theorem for the whole family of complex weighted Boolean #CSP.

This is our third main result. We prove a complexity dichotomy theorem for complex valued Boolean #CSP. This generalizes a theorem by Dyer, Goldberg and Jerrum [13] where each constraint function takes non-negative values. We remark that this third result is incomparable with our dichotomy theorem for Holant^c Problems because it works for all the complex valued functions (not only real symmetric ones). We have to rule out all other manners of fortuitous cancelations similar to that of $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$, this part of the proof is delicate. Due to space limit, many details are omitted. We isolate a property we call Congruity and Semi-congruity, which provides a key insight and plays a decisive role in both the tractability and hardness proofs. We also give a refinement of this result, by restricting the maximum occurrence of each variable to 3 times. In our holant framework, this means that not all the EQUALITY function but these with arity less or equal to 3 are freely available. This part of the proof is more demanding and proof techniques are also interesting.

2. DEFINITIONS AND BACKGROUND

Our functions take values in $\mathbb{C}$ by default. We will mostly be concerned with symmetric functions on Boolean variables, however the framework of Holant Problems is defined for functions mapping any $[q]^k \to \mathbb{C}$ for a finite q. Our results in this paper are for the Boolean case $q = 2$.

As stated, a *signature grid* $\Omega = (H, \mathcal{F}, \pi)$ consists of a graph $H = (V, E)$ with each vertex labeled by a function $f_v \in \mathcal{F}$. We use $\mathcal{F}_q$ when variables range over $[q]$. The Holant problem on instance Ω is to compute $\text{Holant}_\Omega = \sum_\sigma \prod_{v \in V} f_v(\sigma \mid_{E(v)})$, a sum over all edge assignments. A function f_v can be represented as a truth table, or a tensor in $(\mathbb{C}^q)^{\otimes \deg(v)}$. This is called a *signature*. We denote by $=_k$ the Equality signature of arity k. For $q = 2$, let Δ_0 (resp. Δ_1) denote the unary signature which takes value 1 on input 0 (resp. 1), and 0 otherwise. A symmetric function f on k Boolean variables can be expressed by $[f_0, f_1, \ldots, f_k]$, where f_j is the value of f on inputs of Hamming weight j. Thus, $(=_k) = [1, 0, \ldots, 0, 1]$, $\Delta_0 = [1, 0]$ and $\Delta_1 = [0, 1]$. A Holant problem is parameterized by a set of signatures.

Definition 2.1. *Given a set of signatures $\mathcal{F}$, we define a counting problem* Holant($\mathcal{F}$)*:*
Input: A signature grid $\Omega = (G, \mathcal{F}, \pi)$;
Output: Holant_Ω.

We would like to characterize the complexity of Holant problems in terms of its signature sets. Some special families of Holant problems have already been widely studied under other names. For example, if $\mathcal{F}_q$ contains all Equality signatures $\{=_1, =_2, =_3, \cdots\}$, then this is exactly the weighted #CSP problem. Graph homomorphism is a further special case, where we only allow a single binary function in $\mathcal{F}_q$ other than all the Equality functions.

We now define two more special families of Holant problems by assuming some signatures are freely available. We define them for $q = 2$; they can be easily extended to arbitrary $[q]$.

Definition 2.2. *let $\mathcal{U}$ denote the set of all unary signatures. Given a set of signatures $\mathcal{F}$, we use* $\text{Holant}^*(\mathcal{F})$ *to denote* $\text{Holant}(\mathcal{F} \cup \mathcal{U})$.

Definition 2.3. *Given a set of signatures $\mathcal{F}$, we use* $\text{Holant}^c(\mathcal{F})$ *to denote* $\text{Holant}(\mathcal{F} \cup \{\Delta_0, \Delta_1\})$.

Replacing a signature $f \in \mathcal{F}$ by a constant multiple cf, where $c \neq 0$, does not change the complexity of Holant($\mathcal{F}$). So we view f and cf as the same signature. An important property of a signature is whether it is degenerate.

Definition 2.4. *A signature is degenerate iff it is a tensor product of unary signatures.*

In particular, a symmetric signature in $\mathcal{F}$ is degenerate iff it can be expressed as $\lambda[x, y]^{\otimes k}$. Also a symmetric signature $[x_0, x_1, \ldots, x_n]$ is non-degenerate iff $\text{rank}\begin{bmatrix} x_0 & \ldots & x_{n-1} \\ x_1 & \ldots & x_n \end{bmatrix} = 2$.

To introduce the idea of holographic reductions, it is convenient to consider bipartite graphs. This is without loss of generality. For any general graph, we can make it bipartite by adding an additional vertex on each edge, and giving each new vertex the Equality function $=_2$ on 2 inputs.

We use $\#\mathcal{G}_q|\mathcal{R}_q$ to denote all counting problems, expressed as Holant problems on bipartite graphs $H = (U, V, E)$, where each signature for a vertex in U or V is from $\mathcal{G}_q$ or $\mathcal{R}_q$, respectively. An input instance for the bipartite Holant problem is a bipartite signature grid and is denoted as $\Omega = (H, \mathcal{G}_q|\mathcal{R}_q, \pi)$. Signatures in $\mathcal{G}_q$ are denoted by column vectors (or contravariant tensors); signatures in $\mathcal{R}_q$ are denoted by row vectors (or covariant tensors) [12].

One can perform (contravariant and covariant) tensor transformations on the signatures, which may produce exponential cancelations in tensor spaces. We will define a simple version of holographic reductions, which are invertible. Suppose $\#\mathcal{G}_q|\mathcal{R}_q$ and $\#\mathcal{G}'_q|\mathcal{R}'_q$ are two Holant problems defined for the same family of graphs, and $T \in \mathbf{GL}_q(\mathbb{C})$ is a basis. We say that there is a holographic reduction from $\#\mathcal{G}_q|\mathcal{R}_q$ to $\#\mathcal{G}'_q|\mathcal{R}'_q$, if the *contravariant* transformation $G' = T^{\otimes g}G$ and the *covariant* transformation $R = R'T^{\otimes r}$ map $G \in \mathcal{G}_q$ to $G' \in \mathcal{G}'_q$ and $R \in \mathcal{R}_q$ to $R' \in \mathcal{R}'_q$, where G and R have arity g and r respectively. (Notice the reversal of directions when the transformation $T^{\otimes n}$ is applied. This is the meaning of *contravariance* and *covariance*.)

Theorem 2.1 (Valiant's Holant Theorem). *Suppose there is a holographic reduction from $\#\mathcal{G}_q|\mathcal{R}_q$ to $\#\mathcal{G}'_q|\mathcal{R}'_q$ mapping signature grid Ω to Ω', then* $\text{Holant}_\Omega = \text{Holant}_{\Omega'}$.

In particular, for invertible holographic reductions from $\#\mathcal{G}_q|\mathcal{R}_q$ to $\#\mathcal{G}'_q|\mathcal{R}'_q$, one problem is in P iff the other one is, and similarly one problem is #P-hard iff the other one is also.

Theorem 2.2. *Let $\mathcal{F}_q$ be a set of signatures and M be a $q \times q$ orthogonal matrix, i.e., $MM^{\mathrm{T}} = I_q$. For any signature grid $\Omega = (G, \mathcal{F}_q, \pi)$, replacing every signature $F \in \mathcal{F}_q$ by $M^{\otimes n}F$, where n is the arity of F, we can get a new signature grid Ω'. Then* $\text{Holant}_\Omega = \text{Holant}_{\Omega'}$.

Proof. First we use a standard technique to reformulate the signature grid $\Omega = (G, \mathcal{F}_q, \pi)$. We insert a new vertex at each edge of G with signature $=_2$. This will not change the value of the signature grid. Then for the new bipartite signature grid $(G', \mathcal{F}_q|\{=_2\}, \pi)$, we apply a holographic reduction with basis M. This will map a signature $F \in \mathcal{F}_q$ to $M^{\otimes n}F$, where n is the arity of F. It is an algebraic fact that $=_2$ will map to itself. Now we can replace each new $=_2$ node back to an edge to revert back to G. This gives the signature grid Ω' as required. By the Holant theorem, its value is the same as Ω. □

This theorem is very useful as a way to normalize a given signature set $\mathcal{F}_q$.

Starting from next section, we will exclusively focus on Boolean variables. A technical issue is the model of computation for $\mathbb{C}$. Strictly speaking we must only use computable numbers. We will state our results for all $\mathbb{C}$, assuming all numbers in a particular instance (signature) are computable.

3. HOLANT* PROBLEMS

Theorem 3.1. *Let $\mathcal{F}$ be a set of symmetric signatures over $\mathbb{C}$. Then* $\text{Holant}^*(\mathcal{F})$ *is computable in polynomial time in the following three Classes. In all other cases,* $\text{Holant}^*(\mathcal{F})$ *is #P-hard.*

1. *Every signature in $\mathcal{F}$ is of arity no more than two;*

2. There exist two constants a and b (not both zero, depending only on $\mathcal{F}$), such that for all signatures $[x_0, x_1, \ldots, x_n] \in \mathcal{F}$ one of the two conditions is satisfied: (1) for every $k = 0, 1, \ldots, n-2$, we have $ax_k + bx_{k+1} - ax_{k+2} = 0$; (2) $n = 2$ and the signature $[x_0, x_1, x_2]$ is of the form $[2a\lambda, b\lambda, -2a\lambda]$.

3. For every signature $[x_0, x_1, \ldots, x_n] \in \mathcal{F}$ one of the two conditions is satisfied: (1) For every $k = 0, 1, \ldots, n-2$, we have $x_k + x_{k+2} = 0$; (2) $n = 2$ and the signature $[x_0, x_1, x_2]$ is of the form $[\lambda, 0, \lambda]$.

The dichotomy is still true even if the inputs are restricted to planar graphs.

Remark: Since all unary signatures can be used for free, we always assume the arity of every signature in $\mathcal{F}$ is larger than one. And since all the degenerate signatures can be decomposed to unary signatures, we also assume that every signature in $\mathcal{F}$ is non-degenerate.

Proof Outline: It is easy to show that the first Class is computable in P. One can compute the signature of a path by matrix multiplication. The other two polynomial time computable Classes follow from our previous work on Fibonacci gates [9].

Now for the hardness, we first prove in Lemma 3.1 that the theorem holds if $\mathcal{F}$ contains a single symmetric signature of arity three. The main technique is holographic reductions. We make use of the signature theory developed in holographic algorithms [8, 7]. This theory gives us three Categories in a certain parametrization for the signature according to some eigenvalues. For each Category, we choose one #P-hard problem to reduce from, all using holographic reductions. In Lemma 3.2, we prove that if one signature has the form in Class *2* of Theorem 3.1, and we combine it with another signature which is not in this Class, then the Holant* problem is #P-hard. The main idea of the proof is to reduce it to Lemma 3.1 with holographic reductions. In Lemma 3.3, we prove the same thing is true for Class *3*. Finally we extend the above proofs to a set of signatures of arbitrary arities and finish the whole proof.

The following lemma is the first important step towards the proof of Theorem 3.1. Holographic reductions play a decisive role in the proof. This Lemma serves as the foundation for all subsequent lemmas.

LEMMA 3.1. *Let $[x_0, x_1, x_2, x_3]$ be a symmetric signature with arity 3, then $\text{Holant}^*([x_0, x_1, x_2, x_3])$ is #P-hard unless one of the following two statements is true: (1) there exist two constants a, b (not both zero) such that $ax_0 + bx_1 - ax_2 = 0$ and $ax_1 + bx_2 - ax_3 = 0$; (2) $x_0 + x_2 = 0$ and $x_1 + x_3 = 0$.*

Proof: Assume $[x_0, x_1, x_2, x_3]$ does not satisfy either of the two statements, we prove that $\text{Holant}^*([x_0, x_1, x_2, x_3])$ is #P-hard. Our starting point is that $\#[0,1,1]|[1,0,0,1]$ and $\#[1,0,1]|[1,1,0,0]$ are both #P-Complete [29]. The first problem is simply counting the number of vertex covers for 3-regular graphs; while the second is to count the number of (not necessarily perfect) matchings for 3-regular graphs. We remark that both of them remain #P-Hard even for planar graphs.

First we use the signature theory from holographic algorithms to give a better parametrization. Given a non-degenerate signature $[x_0, x_1, x_2, x_3]$, there are three Categories:

- Category 1. $x_i = \alpha_1^{3-i}\alpha_2^i + \beta_1^{3-i}\beta_2^i$, where $\alpha_1\beta_2 - \alpha_2\beta_1 \neq 0$;
- Category 2. $x_i = Ai\alpha^{i-1} + B\alpha^i$, where $A \neq 0$; or
- Category 3. $x_i = A(3-i)\alpha^{2-i} + B\alpha^{3-i}$, where $A \neq 0$.

Category 3 can be viewed as the reversal of Category 2, so we will omit the proof for Category 3. The choices made here in this particular parametrization is informed by the "signature theory" [8, 7] that we have developed in previous work. (But one can directly check that for any non-degenerate signature $[x_0, x_1, x_2, x_3]$, one of these three parameterizations is always possible. Note that, if $\alpha = 0$ then we take the convention that the expression $i\alpha^{i-1} = 0, 1, 0, 0$ for $i = 0, 1, 2, 3$ respectively.)

For Category 1, we have

$$X = [x_0, x_1, x_2, x_3] = \begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}^{\otimes 3} + \begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}^{\otimes 3}.$$

We restate our conditions from the Lemma statement in this new parametrization. The fact that X is non-degenerate implies that $\alpha_1\beta_2 - \alpha_2\beta_1 \neq 0$. The fact that X is not in the case indicated in statement (1) implies that $\alpha_1\beta_1 + \alpha_2\beta_2 \neq 0$. The fact that X is not in the case indicated in statement (2) implies that $\alpha_1^2 + \alpha_2^2 \neq 0$ or $\beta_1^2 + \beta_2^2 \neq 0$. By symmetry, we can assume that $\alpha_1^2 + \alpha_2^2 \neq 0$.

Under the condition $\alpha_1^2 + \alpha_2^2 \neq 0$, we can apply an orthogonal transformation to map the vector (α_1, α_2) to be of the form $(\alpha_1', 0)$, where $\alpha_1' \neq 0$. We may use a (complex orthogonal) Householder matrix for this purpose. Then under this orthogonal basis, the signature becomes

$$X' = [x_0', x_1', x_2', x_3'] = \begin{bmatrix}\alpha_1'\\ 0\end{bmatrix}^{\otimes 3} + \begin{bmatrix}\beta_1'\\ \beta_2'\end{bmatrix}^{\otimes 3}.$$

By Theorem 2.2, this transformation does not change the complexity of the Holant problem. So it suffices to prove the #P-hardness result for this signature. By a scalar multiplication we assume $\alpha_1' = 1$. So, reuse the notation X, we can assume the signature is of this form

$$X = [x_0, x_1, x_2, x_3] = \begin{bmatrix}1\\ 0\end{bmatrix}^{\otimes 3} + \begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}^{\otimes 3}.$$

The two conditions from the statement of the Lemma become simply $\beta_1\beta_2 \neq 0$.

Now under the basis $T = \begin{bmatrix}1 & \beta_1\\ 0 & \beta_2\end{bmatrix}$, signature $[1, 0, 0, 1]$ becomes $[x_0, x_1, x_2, x_3]$. This is the result of the contravariant transformation (on truth tables) $(x_0, x_1, x_1, x_2, x_1, x_2, x_2, x_3)^{\mathrm{T}} = T^{\otimes 3}(1, 0, 0, 0, 0, 0, 0, 1)^{\mathrm{T}}$, namely $X = T^{\otimes 3}\left(\begin{bmatrix}1\\ 0\end{bmatrix}^{\otimes 3} + \begin{bmatrix}0\\ 1\end{bmatrix}^{\otimes 3}\right)$.

Under the same basis, $[0, 1, 1]$ undergoes a covariant transformation, we have

$$(0, 1, 1, 1)(T^{-1})^{\otimes 2} = \frac{1}{\beta_2^2}(0, \beta_2, \beta_2, 1 - 2\beta_1).$$

Again, we can ignore the scalar factor $1/\beta_2^2$. So by the holographic reduction defined by T, the complexity of the problem $\#[0, \beta_2, 1 - 2\beta_1]|[x_0, x_1, x_2, x_3]$ is the same as $\#[0,1,1]|[1,0,0,1]$, which is #P-Hard (vertex cover). In order to prove that $\text{Holant}^*([x_0, x_1, x_2, x_3])$ is #P-Hard, we only need to show that the signature $[0, \beta_2, 1 - 2\beta_1]$ can be realized by $[x_0, x_1, x_2, x_3]$ with some unary signatures.

For a binary signature F we can write it in a matrix form $\begin{bmatrix} F(00) & F(01) \\ F(10) & F(11) \end{bmatrix}$. We use the gadget in Figure 1 to realize

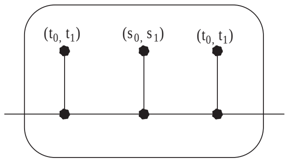

Figure 1: We use this gadget to realize the signature $[0, \beta_2, 1-2\beta_1]$**. All (three) nodes of degree 3 in this gadget have the signature** $[x_0, x_1, x_2, x_3]$**.**

$[0, \beta_2, 1-2\beta_1]$, where the two unary signatures (t_0, t_1) and (s_0, s_1) will be determined later. Let

$$A = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix} = \begin{bmatrix} \beta_1^2 & \beta_1\beta_2 \\ \beta_1\beta_2 & \beta_2^2 \end{bmatrix}.$$

In X, if one input is 0, the induced binary signature has a matrix form $A + \beta_1 B$. If one input is 1, the induced binary signature has a matrix form $\beta_2 B$. It follows that the signature of the above gadget is

$$\begin{aligned} & (t_0(A+\beta_1 B) + t_1\beta_2 B)(s_0(A+\beta_1 B) + s_1\beta_2 B) \\ & (t_0(A+\beta_1 B) + t_1\beta_2 B) \\ = \ & (t_0 A + (t_0\beta_1 + t_1\beta_2)B)(s_0 A + (s_0\beta_1 + s_1\beta_2)B) \\ & (t_0 A + (t_0\beta_1 + t_1\beta_2)B). \end{aligned}$$

Now we use a new set of variables $x = t_0, y = t_0\beta_1 + t_1\beta_2, z = s_0, w = s_0\beta_1 + s_1\beta_2$, and write the above matrix as $(xA + yB)(zA + wB)(xA + yB)$. We note that for any given x, y, z, w, we can find t_0, t_1, s_0, s_1 to satisfy the above relationships. Then, to realize $[0, \beta_2, 1-2\beta_1]$, we just want to choose some x, y, z, w such that

$$(xA + yB)(zA + wB)(xA + yB) = \begin{bmatrix} 0 & \beta_2 \\ \beta_2 & 1-2\beta_1 \end{bmatrix}.$$

We show that we can find some x, y, z, w to satisfy the above condition.

Substituting A and B, and denote by $\gamma = \beta_1^2 + \beta_2^2$, we have the following:

$$\begin{aligned} & (xA + yB)(zA + wB)(xA + yB) \\ = \ & w \begin{bmatrix} \beta_1^2(x+y\gamma)^2 & y\beta_1\beta_2\gamma(x+y\gamma) \\ y\beta_1\beta_2\gamma(x+y\gamma) & y^2\beta_2^2\gamma^2 \end{bmatrix} + \\ & z \begin{bmatrix} (x+y\beta_1^2)^2 & y\beta_1\beta_2(x+y\beta_1^2) \\ y\beta_1\beta_2(x+y\beta_1^2) & y^2\beta_1^2\beta_2^2 \end{bmatrix} \end{aligned}$$

We may choose $w = (x + y\beta_1^2)^2$ and $z = -\beta_1^2(x+y\gamma)^2$ to make the $(1,1)$ entry zero. The $(1,2)$ (and $(2,1)$) entry is

$$g_1 = xy\beta_1\beta_2^3(x + \beta_1^2 y)(x + y\gamma);$$

and the $(2,2)$ entry is

$$g_2 = xy^2\beta_2^4(x(2\beta_1^2 + \beta_2^2) + 2y(\beta_1^4 + \beta_1^2\beta_2^2)).$$

We want to choose some x, y such that $[g_1, g_2] = [\beta_2, 1-2\beta_1]$. We have $\beta_2 \neq 0$. We will choose $xy \neq 0$. As both g_1 and g_2 are homogenous in x and y, we can ignore the common factor $xy\beta_2^3$ of g_1 and g_2. It follows that we only have to satisfy that $g_2/g_1 = (1-2\beta_1)/\beta_2$ with $y = 1$. We need the following

$$\begin{aligned} 0 &= \beta_2 g_2 - (1-2\beta_1)g_1 \\ &= \beta_1(2\beta_1 - 1)x^2 + (2\beta_1^2 - \beta_1 + \beta_2^2)(2\beta_1^2 + \beta_2^2)x + \\ & \quad \beta_1^2(\beta_1^2 + \beta_2^2)(2\beta_1^2 - \beta_1 + 2\beta_2^2). \qquad (1) \end{aligned}$$

What we have to prove is that at least one of the roots to the equation in (1) is not a root of $g_1 = g_1(x, 1) = 0$. The roots of $g_1 = 0$ are $x = 0, x = -\beta_1^2$ and $x = -\gamma$. Firstly we can verify that $x = -\beta_1^2$ can not be a root of (1). This is because when $x = -\beta_1^2$, the expression in (1) can be simplified to $\beta_1^2\beta_2^4 \neq 0$. Secondly if $x = -\gamma$ is a root of (1), the expression in (1) can be simplified to $-\beta_2^4\gamma$, and this would force $\gamma = 0$. So, assuming the expression in (1) as a polynomial in x is indeed of degree 2, then the only case we need to worry about is that $x = 0$ is a double root of (1). In fact, suppose (1) is indeed quadratic, and $x = 0$ is not a double root, then we may let $\xi \neq 0$ be a root of (1). This $\xi \neq -\beta_1^2$, because $-\beta_1^2$ is not a root of (1); ξ can't be $-\gamma$ either, for otherwise $-\gamma$ would be a root of (1) which we had proved it would force $\gamma = 0$, and thus $\xi = -\gamma = 0$, a contradiction. Thus ξ is a root of (1) but not a root of g_1, as is needed.

Now let's consider the exceptional cases: either $x = 0$ is a double root of (1), or (1) has degree less than 2. If $x = 0$ is a double root of (1), we have

$$(2\beta_1^2 - \beta_1 + \beta_2^2)(2\beta_1^2 + \beta_2^2) = \beta_1^2(\beta_1^2 + \beta_2^2)(2\beta_1^2 - \beta_1 + 2\beta_2^2) = 0.$$

To satisfy this, there are only four exceptional cases (A1 to A4): $\beta_1 = 1, \beta_2 = \pm i$ or $\beta_1 = -\frac{1}{2}, \beta_2 = \pm\frac{i}{\sqrt{2}}$. On the other hand, if the polynomial in (1) has degree less than 2, then $\beta_1 = \frac{1}{2}$. In this case, the polynomial becomes

$$(1/2 + \beta_2^2)x + (1/4 + \beta_2^2)/2 = 0.$$

This gives us four additional exceptional cases (B1 to B4): $\beta_1 = \frac{1}{2}, \beta_2 = \pm\frac{i}{2}$, in which case the polynomial is linear with root $x = 0$; or $\beta_1 = \frac{1}{2}, \beta_2 = \pm\frac{i}{\sqrt{2}}$, in which case the polynomial degenerates to a (non-zero) constant. In all other cases, there is a root of (1) which is not a root of g_1, which completes the #P-hardness proof.

For the cases A1 and A2, we use a new starting problem $\#[1,1,0]|[1,0,0,1]$, which is the reversal of the previous problem and therefore it is also #P-Hard. Then all previous part of the proof is still valid, except that the signature of arity two we would like to realize is

$$(1,1,1,0)(T^{-1})^{\otimes 2} = (1, \frac{1-\beta_1}{\beta_2}, \frac{1-\beta_1}{\beta_2}, \frac{\beta_1^2 - 2\beta_1}{\beta_2^2}).$$

Substituting $\beta_1 = 1, \beta_2 = \pm i$, the signature is $[1, 0, 1]$ which is trivially realizable by one edge. So we have proved that it is #P-Hard in the cases A1 and A2. Now consider the cases A3 and A4, $\beta_1 = -\frac{1}{2}, \beta_2 = \pm\frac{i}{\sqrt{2}}$. We will give a different parametrization. For case A3, we apply an orthogonal transformation $M = \begin{bmatrix} -i & -\sqrt{2} \\ \sqrt{2} & -i \end{bmatrix}$ and a scalar multiplier $2i$ on the signature and it becomes $\begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\otimes 3} + \begin{bmatrix} 2 \\ 2\sqrt{2}i \end{bmatrix}^{\otimes 3}$. This is not one of the exceptional cases and we have proved that it is #P-Hard. For case A4, we apply another orthogonal transformation $M' = \begin{bmatrix} i & -\sqrt{2} \\ \sqrt{2} & i \end{bmatrix}$ and a scalar multiplier $-2i$ on the signature and it becomes $\begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\otimes 3} + \begin{bmatrix} 2 \\ -2\sqrt{2}i \end{bmatrix}^{\otimes 3}$.

The cases B3 and B4 can be shown by the same method as in A4 and A3, using M' and M respectively. The only cases left are B1 and B2. Here we will use another gadget similar to the one in Figure 1 except we remove the middle edge (including the node labeled (s_0, s_1) and the middle node of degree 3). For B1, the signature of this gadget is

$$(t_0(A+\beta_1 B)+t_1\beta_2 B)^2 = (xA+yB)^2,$$

where A and B are as before, and with the specific values of β_1, β_2, $B = \frac{1}{4}\begin{bmatrix}1 & i\\ i & -1\end{bmatrix}$. By setting $x = i$ and $y = -2i$, we have $(xA+yB)^2 = \begin{bmatrix}0 & i/2\\ i/2 & 0\end{bmatrix}$, which is the matrix form of the target signature $[0, \beta_2, 1-2\beta_1] = [0, \frac{i}{2}, 0]$. This finishes case B1. The case B2 can be done with $x = 1$ and $y = -2$.

Now we prove for Category 2. In this case $x_i = Ai\alpha^{i-1} + B\alpha^i$, the condition that it does not satisfy statement (2) in Lemma 3.1 implies that $\alpha \neq \pm i$. This is because rank $(\begin{bmatrix}x_0 - x_2 & x_1\\ x_1 - x_3 & x_2\end{bmatrix}) = 2$ and its determinant can be shown to be $-A^2(1+\alpha^2)$. Under this condition, we can choose some orthogonal transformation to make it in the form $[x, y, 0, 0]$ where $y \neq 0$. In fact, if we let $T = \begin{bmatrix}1 & \frac{B-1}{3}\\ \alpha & A + \frac{B-1}{3}\alpha\end{bmatrix}$, then the signature $[x_0, x_1, x_2, x_3]$ can be expressed as

$$(x_0, x_1, x_1, x_2, x_1, x_2, x_2, x_3)^{\mathrm{T}} = T^{\otimes 3}(1,1,1,0,1,0,0,0)^{\mathrm{T}}.$$

(We chose these basis transformations based on an underlying signature theory of holographic algorithms, not "out of blue". But for brevity of exposition we state these transformations *as is* without discussing the background. They can be directly verified, albeit a bit tedious.) Let $T = QR$ be its QR factorization, where Q is orthogonal and R is upper triangular. In fact if we denote $T = \begin{bmatrix}1 & *\\ \alpha & *\end{bmatrix}$, then we can choose our Q as the (orthogonal) Householder matrix, which is a (complex) reflection, $Q = Q^{\mathrm{T}} = \frac{1}{\sqrt{1+\alpha^2}}\begin{bmatrix}1 & \alpha\\ \alpha & -1\end{bmatrix}$. Then $QT = R = \begin{bmatrix}u & w\\ 0 & v\end{bmatrix}$ is upper triangular, where $u = \sqrt{1+\alpha^2}$. As $\det Q = -1$, $\det R = -\det T = -A \neq 0$, we have $uv \neq 0$. This Q is our choice of the orthogonal transformation. It follows that

$$\begin{aligned}
& Q^{\otimes 3}(x_0, x_1, x_1, x_2, x_1, x_2, x_2, x_3)^{\mathrm{T}}\\
= \;& (QT)^{\otimes 3}(1,1,1,0,1,0,0,0)^{\mathrm{T}}\\
= \;& R^{\otimes 3}(1,1,1,0,1,0,0,0)^{\mathrm{T}}\\
= \;& R^{\otimes 3}\left\{\begin{bmatrix}1\\0\end{bmatrix}^{\otimes 3} + \begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} + \right.\\
& \left.\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}\right\}\\
= \;& \begin{bmatrix}u\\0\end{bmatrix}^{\otimes 3} + \begin{bmatrix}u\\0\end{bmatrix}\otimes\begin{bmatrix}u\\0\end{bmatrix}\otimes\begin{bmatrix}w\\v\end{bmatrix} + \begin{bmatrix}u\\0\end{bmatrix}\otimes\begin{bmatrix}w\\v\end{bmatrix}\otimes\begin{bmatrix}u\\0\end{bmatrix} +\\
& \begin{bmatrix}w\\v\end{bmatrix}\otimes\begin{bmatrix}u\\0\end{bmatrix}\otimes\begin{bmatrix}u\\0\end{bmatrix}
\end{aligned}$$

This can be written as a symmetric signature form $[u^3 + 3u^2w, u^2v, 0, 0]$. Note that the entry $u^2v \neq 0$.

By a scalar multiplication, we can make the entry u^2v equal to 1. So we only have to deal with a signature of the form $[v, 1, 0, 0]$ for an arbitrary given v.

For this signature, we can apply a holographic transformation defined by the matrix $T' = \begin{bmatrix}1 & \frac{v-1}{3}\\ 0 & 1\end{bmatrix}$ with inverse $T'^{-1} = \begin{bmatrix}1 & -\frac{v-1}{3}\\ 0 & 1\end{bmatrix}$. To prove #P-hardness, we will reduce from the MATCHING problem $\#[1,0,1] \mid [1,1,0,0]$. Under a contravariant transformation $(v,1,1,0,1,0,0,0)^{\mathrm{T}} = T'^{\otimes 3}(1,1,1,0,1,0,0,0)^{\mathrm{T}}$, the signature $[1,1,0,0]$ becomes $[v,1,0,0]$. Under the same basis, $[1,0,1]$ undergoes the covariant transformation to become $(1,0,0,1)(T'^{-1})^{\otimes 2} = ((1,0)^{\otimes 2} + (0,1)^{\otimes 2})(T'^{-1})^{\otimes 2} = (1, \frac{1-v}{3}, \frac{1-v}{3}, 1 + \frac{(1-v)^2}{9})$. I.e., the signature $[1,0,1]$ becomes a new symmetric signature $[1, \frac{1-v}{3}, 1+\frac{(1-v)^2}{9}]$. The proof is then to use the same gadget as in Figure 1 to realize this signature, using unary signatures and $[v,1,0,0]$.

We will rename the values $x = t_0$, $y = t_1$, $z = s_0$ and $w = s_1$ in Figure 1. The signature of this gadget in matrix form is $(xA+yB)(zA+wB)(xA+yB)$, where $A = \begin{bmatrix}v & 1\\ 1 & 0\end{bmatrix}$ and $B = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}$. After some calculations we found that this signature in symmetric form is $[w\cdot(x^2v^2 + 2xyv + y^2) + z\cdot(x^2(v^3+2v) + 2xy(v^2+1) + y^2v) + xyv), w\cdot(x^2v + xy) + z\cdot(x^2(v^2+1) + xyv), w\cdot x^2 + z\cdot x^2v]$. Our goal is to choose x, y, z and w such that it is equal to $[1, \frac{1-v}{3}, 1+\frac{(1-v)^2}{9}]$. We can write this as a system of three linear equations in z and w. Then we can complete the proof, if we can choose x and y such that the following matrix has determinant 0, yet the first two columns have rank 2.

$$\begin{bmatrix}x^2v^2 + 2xyv + y^2 & x^2(v^3+2v) + 2xy(v^2+1) + y^2v & 1\\ x^2v + xy & x^2(v^2+1) + xyv & \frac{1-v}{3}\\ x^2 & x^2v & 1 + \frac{(1-v)^2}{9}\end{bmatrix}.$$

After some row operations it becomes $\begin{bmatrix}y^2 & 2xy + y^2v & f_3\\ xy & x^2 + xyv & f_2\\ x^2 & x^2v & f_1\end{bmatrix}$, where f_1, f_2, f_3 are polynomials in v, and explicitly, $f_1 = (10 - 2v + v^2)/9$ and $f_2 = (3 - 13v + 2v^2 - v^3)/9$. Subtracting from the second column the first column multiplied by v, we get $\begin{bmatrix}y^2 & 2xy & f_3\\ xy & x^2 & f_2\\ x^2 & 0 & f_1\end{bmatrix}$. We will set $x = 1$; this guarantees that the first two columns have rank 2, and gives the matrix $\begin{bmatrix}y^2 & 2y & f_3\\ y & 1 & f_2\\ 1 & 0 & f_1\end{bmatrix}$. Now the determinant is easily calculated, (subtract the first row by the second row multiplied by y, and the second from the third multiplied by y). The determinant is $-(f_1y^2 - 2f_2y + f_3)$. As long as f_1 and f_2 are not simultaneously 0, we can always choose a y to make this determinant 0.

However it is easy to show that f_1 and f_2 have no common zero in v, as $3(f_2 + vf_1) = 1 - v$ and $v = 1$ is not a zero of either f_1 or f_2. This completes the proof. □

Lemma 3.1 shows us what happens when there is a single non-degenerate symmetric signature of arity 3. It explicitly lists two exceptional cases whose #P-Hardness can not be deduced from Lemma 3.1. The next Lemma addresses what happens if one signature of arity 3 happens to be in the first exceptional case, but some other signature *does not quite fit*.

LEMMA 3.2. *Let $[x_0, x_1, x_2, x_3]$ and $[y_0, y_1, y_2]$ be non-degenerate symmetric signatures with arity three and two respectively. Suppose there exist two constants a, b (not both zero), such that $ax_0 + bx_1 - ax_2 = 0$ and $ax_1 + bx_2 - ax_3 = 0$,* *but* $ay_0 + by_1 - ay_2 \neq 0$ *and* $[y_0, y_1, y_2]$ *is not of the form* $[2a\lambda, b\lambda, -2a\lambda]$. *Then* $\text{Holant}^*(\{[x_0, x_1, x_2, x_3], [y_0, y_1, y_2]\})$ *is #P-hard.*

Lemma 3.3 does the same thing as Lemma 3.2 for the other exceptional case of the arity 3 signature.

Lemma 3.3. *Let* $[x, y, -x, -y]$ *be a symmetric signature with arity three and* $[y_0, y_1, y_2]$ *be a symmetric signature with arity two. Suppose they are both non-degenerate. If* $y_0 + y_2 \neq 0$ *and* $[y_0, y_1, y_2]$ *is not of the form* $[\lambda, 0, \lambda]$*, then* $\text{Holant}^*(\{[x, y, -x, -y], [y_0, y_1, y_2]\})$ *is #P-hard.*

Finally we further extend this result to a set of signatures with arbitrary arities and finish the proof for Theorem 3.1. The proofs are omitted here and will be presented in the full paper.

4. HOLANT^C PROBLEMS

Theorem 4.1. *Let* $\mathcal{F}$ *be a set of real symmetric signatures, and let* $\mathcal{F}_1, \mathcal{F}_2$ *and* $\mathcal{F}_3$ *be three families of signatures defined as*

$$\begin{aligned}
\mathcal{F}_1 &= \{\lambda([1,0]^{\otimes k} + i^r[0,\ \ 1]^{\otimes k}) \mid \lambda \in \mathbb{C}, k = 1, 2, \dots, \\
&\quad \text{and } r = 0, 1, 2, 3\}; \\
\mathcal{F}_2 &= \{\lambda([1,1]^{\otimes k} + i^r[1,-1]^{\otimes k}) \mid \lambda \in \mathbb{C}, k = 1, 2, \dots, \\
&\quad \text{and } r = 0, 1, 2, 3\}; \\
\mathcal{F}_3 &= \{\lambda([1,\ i]^{\otimes k} + i^r[1,\ -i]^{\otimes k}) \mid \lambda \in \mathbb{C}, k = 1, 2, \dots, \\
&\quad \text{and } r = 0, 1, 2, 3\}.
\end{aligned}$$

Then $\text{Holant}^c(\mathcal{F})$ *is computable in polynomial time if (1) After removing unary signatures from* $\mathcal{F}$*, it falls in one of the three Classes of Theorem 3.1 (this implies* $\text{Holant}^*(\mathcal{F})$ *is computable in polynomial time) or (2) (Without removing any unary signature)* $\mathcal{F} \subseteq \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$*. Otherwise,* $\text{Holant}^c(\mathcal{F})$ *is #P-hard.*

Proof Outline: By definition, every instance of $\text{Holant}^c(\mathcal{F})$ is also an instance of $\text{Holant}^*(\mathcal{F})$. So it is obvious that if $\text{Holant}^*(\mathcal{F})$ is computable in polynomial time then so is $\text{Holant}^c(\mathcal{F})$. The polynomial time algorithm for $\mathcal{F} \subseteq \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$ is non-trivial. This problem is a special case of a polynomial time computable problem #CSP($\mathcal{A}$), where $\mathcal{A}$ is defined in Section 5 as an unsymmetric generalization of $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$. The algorithm for #CSP($\mathcal{A}$) will be given in Section 5 Theorem 5.2 as part of the proof for complex weighted #CSP dichotomy theorem.

The main result here is hardness. We want to prove that aside from these tractable cases, all remaining problems are #P-hard. Here the main technique is polynomial interpolation. We prove the second dichotomy theorem (Theorem 4.1) by a reduction to the first (Theorem 3.1). We will show how to interpolate all the unary signatures. Once we can interpolate all unary signatures, we can make use of the dichotomy theorem for $\text{Holant}^*(\mathcal{F})$. The whole proof is organized as a sequence of lemmas. In each lemma, we prove the theorem for a larger family of $\mathcal{F}$, and the remaining unproved ones are the beginning of the next lemma. Finally we prove the theorem for all possible signature sets $\mathcal{F}$. (In this Extended Abstract, we only present the first of these lemmas, Lemma 4.3, and leave others to the full paper.) In some cases, the attempt to interpolate all unary signatures does not work. In these cases, we employ yet another (the third) starting point of #P-hardness, which is the problem of counting Perfect Matchings on 3-regular graphs [11]. We reduce the Perfect Matching problem also by polynomial interpolation , which is done in Lemma 4.4. However, note that counting Perfect Matchings is computable in polynomial time for planar graphs [22, 23, 24], therefore our dichotomy theorem for Holant^c problems here does not extend to planar graphs as our dichotomy theorem for Holant^* problems does.

The interpolation method was introduced by Valiant [26]. The interpolation method we use here is essentially the same as Vadhan [25]. We construct a sequence of gadgets with unary signatures. These signatures are denoted by $\{G_s\} = [x_s, y_s]$ and are related by the following recurrence

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{s-1} \\ y_{s-1} \end{bmatrix}. \qquad (2)$$

We denote by $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $g = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$. We call this pair (A, g) a recursive construction. It follows from lemma 6.1 in [25] that

Lemma 4.1. *Let* α, β *be the two eigenvalues of* A*. If the following three conditions are satisfied (1)* $\det(A) \neq 0$*; (2)* g *is not a column eigenvector of* A *(nor the zero vector); (3)* α/β *is not a root of unity, then the recursive construction* (A, g) *can be used to interpolate all unary signatures.*

Since two unary signatures $[1, 0]$ and $[0, 1]$ are freely available, we can get the following lemma from Lemma 4.1 easily.

Lemma 4.2. *If we can construct a gadget with signature* $[a, b, c]$*, where* $b^2 \neq ac$*,* $b \neq 0$ *and* $a + c \neq 0$*, then we can interpolate all the unary function. (Hence Theorem 4.1 holds.)*

We define some families of symmetric signatures, which will be used in our proof.

$$\begin{aligned}
\mathcal{G}_1 &= \{[a, 0, 0, \cdots, 0, b] | ab \neq 0\} \\
\mathcal{G}_2 &= \{[x_0, x_1, \cdots, x_k] | \forall i \text{ even (or } \forall i \text{ odd)}, x_i = 0\} \\
\mathcal{G}_3 &= \{[x_0, x_1, \cdots, x_k] | \forall i, x_i + x_{i+2} = 0\}
\end{aligned}$$

We note that $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ are supersets of $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ respectively. As the first step for the proof of Theorem 4.1, we prove that if $\mathcal{F}$ contains any signature which is not in $\mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$, then we can interpolate all the unary signatures.

Lemma 4.3. *If* $\mathcal{F} \not\subseteq \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$*, then Theorem 4.1 holds.*

Proof. Since $\mathcal{F} \not\subseteq \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$, there exists a $f \in \mathcal{F}$ and $f \notin \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$. Since all the unary signatures are in $\mathcal{G}_3$, the arity of f is larger than 1 and f is non-degenerate. There are two cases according to whether f has a zero entry or not.

(1) f has some zero entries. If there exists a sub signature of f has the form $[0, a, b]$ or $[a, b, 0]$, where $ab \neq 0$, then we are done by Lemma 4.2. Otherwise, we can conclude that there is no two successive non-zero entries. So the signature f has this form $[0^{i_0} x_1 0^{i_1} x_2 0^{i_2} \cdots x_k 0^{i_k}]$, where $x_j \neq 0$ and for all $1 \leq j \leq k-1$, $i_j \geq 1$. If for all $1 \leq j \leq k-1$, i_j is odd, then $f \in \mathcal{G}_2$, a contradiction. Otherwise there exists a sub signature of the form $[x, 0, 0, \cdots, 0, y]$, where $xy \neq 0$

and there are an even number of 0s between x and y. If this is the whole f, then $f \in \mathcal{G}_1$, a contradiction. So there is one 0 before x or after y. By symmetry, we assume there is a 0 before x, so we have a sub signature $[0, x, 0, 0, \cdots, 0, y]$, whose arity is even and larger than 3. We call its dangling edges $1, 2, \cdots, 2k$. Then for every $i = 1, 2, \cdots, k-1$, we connect dangling edges $2i+1$ and $2i+2$ together to a regular edge. After that, we have a $\mathcal{F}$-gate with arity 2, and its signature is $[0, x, y]$. Then we are done by Lemma 4.2.

(2) f has no zero entry. We only need to prove that we can construct a function $[a', b', c']$ satisfying the three conditions in Lemma 4.2. Suppose all sub signatures of f with arity 2 do not satisfy all the three conditions. For each sub-signature $[a', b', c']$, either $a' + c' = 0$, or $b'^2 = a'c'$. If all of them satisfy $a' + c' = 0$, then $f \in \mathcal{G}_3$. A contradiction. If all of them satisfy $b'^2 = a'c'$, then f is degenerate. A contradiction. W.l.o.g, we can assume there is a sub-signature $[a, b, c, d]$ of f, such that $a + c = 0$, $b + d \neq 0$, and $c^2 = bd$. Combining two $[a, b, c, d]$, we can get a function $[a', b', c'] = [a^2 + 2b^2 + c^2, ab + 2bc + cd, b^2 + 2c^2 + d^2]$. $b' = c(b + d) \neq 0$. $a' + c' = a^2 + 3b^2 + 3c^2 + d^2 > 0$. Because $c^2 = bd$, $a'c' - b'^2 = a^2b^2 + 2a^2c^2 + a^2d^2 + 2b^4 + 4b^2c^2 + 2b^2d^2 \neq 0$. We are done by Lemma 4.2. $\square$

LEMMA 4.4. *If $a \neq \pm 1$, $\text{Holant}^c([0, 1, 0, a])$ is #P Hard.*

PROOF. Our starting point here is that $\text{Holant}([0, 1, 0, 0])$ is #P-Hard. This is exactly the perfect matching problem in 3-regular graph [11]. So the problem is #P-Hard if $a = 0$.

Now assume that $a \notin \{-1, 0, 1\}$, and we use this signature to interpolate all the signature of form $[0, 1, 0, x]$, in particular, we can interpolate $[0, 1, 0, 0]$ and finish the hardness reduction.

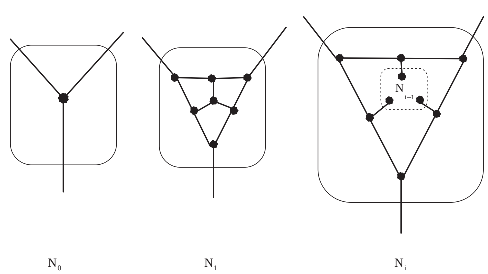

Figure 2: A recursive construction. The signature of every vertex in the gadget is $[0, 1, 0, a]$.

The recursive construction is depicted by Figure 2. By a simple parity argument, every $\mathcal{F}$-gate N_i has a signature of the form $[0, x_i, 0, y_i]$. After some calculation, we can get that they satisfy the following recursive relation:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 3(a^2+1) & a^3+a \\ 3(a^3+a) & a^6+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}.$$

In this case, the signatures we want to interpolate are of arity 3, but since all of them are of form $[0, x_i, 0, y_i]$ with two dimensions freedom, we can also use the interpolation method as in Lemma 4.1. Let $A = \begin{bmatrix} 3(a^2+1) & a^3+a \\ 3(a^3+a) & a^6+1 \end{bmatrix}$, then $(A, [1, a]^{\mathrm{T}})$ forms a recursive construction. Since $\det(A) = 3(a^4-1)^2 \neq 0$, the first condition holds. Its characterize equation is $X^2 - (a^6 + 3a^2 + 4)X + 3(a^4 - 1)^2 = 0$. For this quadratic equation, the discriminant $\Delta = (a^6 - 3a^2 - 2)^2 + 12(a + a^3)^2 > 0$. So A has two distinct real eigenvalues. The sum of the two eigenvalues is $a^6 + 3a^2 + 4$ which is larger than zero. So they are not opposite to each other. Therefore, the ratio of these two eigenvalues is not a root of unity and the third condition holds. Consider the second condition, if the initial vector $[1, a]$ is a column eigenvectors of A. We have $A \begin{bmatrix} 1 \\ a \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ a \end{bmatrix}$, where λ is one eigenvalue of A. From this, we will conclude that $a(a^2 - 1)(a^4 - 1) = 0$, which will not happen given $a \notin \{-1, 0, 1\}$. To sum up, this recursive relation satisfies all the three conditions of Lemma 4.1 and can be used to interpolate all the signatures of form $[0, 1, 0, x]$. This completes the proof. $\square$

5. WEIGHTED BOOLEAN #CSP

We define two classes of functions, for which the complex weighted #CSP problems are tractable.

X denotes the $k+1$ dimensional column vector $(x_1, x_2, \ldots, x_k, 1)$ over Boolean field $\mathbb{F}_2$. Suppose A is a Boolean matrix. χ_{AX} denotes the affine relation on inputs $x_1, x_2, \ldots, x_k$, whose value is 1 if AX is the zero vector, 0 if AX is not the zero vector. Suppose F is a function on input variables $x_1, x_2, \ldots, x_k$. $F^{x_s=c}$ denotes the function $F^{x_s=c}(x_1, \ldots, x_{s-1}, x_{s+1}, \ldots, x_k) = F(x_1, \ldots, x_{s-1}, c, x_{s+1}, \ldots, x_k)$, and $F^{x_s=*}$ denotes the function $F^{x_s=*}(x_1, \ldots, x_{s-1}, x_{s+1}, \ldots, x_k) = \sum_{x_s} F(x_1, \ldots, x_k)$.

We use $\mathcal{A}$ to denote all functions which have the form $\chi_{AX} i^{L_1(X)+L_2(X)+\cdots+L_n(X)}$, where $i = \sqrt{-1}$, L_j is a 0-1 indicator function $\chi_{\langle \alpha_j, X \rangle}$, where α_j is a $k+1$ dimensional vector, the inner product $\langle \cdot, \cdot \rangle$ is over $\mathbb{Z}_2$. The additions among $L_j X$ are just the usual addition in $\mathbb{Z}$. It can be computed mod 4, but not mod 2. (Since we ignore global constant, all functions that are constant multiples of these functions are also in this class.)

$\mathcal{P}$ denotes the class of functions which can be expressed as a product of unary functions, binary equality functions ($[1, 0, 1]$) and binary disequality functions ($[0, 1, 0]$).

THEOREM 5.1. *Suppose $\mathcal{F}$ is a class of functions mapping Boolean inputs to complex numbers. If $\mathcal{F} \subseteq \mathcal{A}$ or $\mathcal{F} \subseteq \mathcal{P}$, then #CSP($\mathcal{F}$) is computable in polynomial time. Otherwise, #CSP($\mathcal{F}$) is #P hard.*

Proof Outline: The polynomial time algorithm for #CSP($\mathcal{P}$) is obvious. Lemma 5.2 gives a polynomial time algorithm for #CSP($\mathcal{A}$). In dichotomy theorems for unweighted and non-negative weighted #CSP problems, the tractable part is relatively obvious. In our dichotomy theorem, we have a more interesting tractable part because of cancelations. In Lemma 5.4, we prove that #CSP($\{F\}$) is #P-hard unless F has affine support. This structure is essential in the proof of Lemma 5.5 and Lemma 5.6, the two key lemmas of the hardness reduction. The common strategy of Lemma 5.5 and Lemma 5.6 is to reduce the arity of a given function. In lemma 5.5, we prove that given a function F, which is not in $\mathcal{A}$, we can simulate (in polynomial time) a unary function $F' \notin \mathcal{A}$; In Lemma 5.6, we prove that given a function G, which is not in $\mathcal{P}$, we can simulate (in polynomial time) a binary or ternary function $G' \notin \mathcal{P}$. Then we prove that #CSP($\{F', G'\}$) is #P-hard. The starting point of the hardness result is Lemma 5.3, which says that if $\mathcal{F}$ contains only one binary symmetric function and is not in $\mathcal{A} \cup \mathcal{P}$, then the

#CSP problem is #P-hard. To complete the proof, we show that we can always combine functions F' and G' to realize a binary symmetric function which is not in $\mathcal{P} \cup \mathcal{A}$.

Now we analyze #CSP($\mathcal{A}$). Firstly, we show how to get rid of the factor χ_{AX}.

LEMMA 5.1. *Let* $F(x_1, x_2, \ldots, x_k) = \chi_{AX} \cdot i^{L_1(X)+L_2(X)+\cdots+L_n(X)} \in \mathcal{A}$. *If* $AX = 0$ *is infeasible over* $\mathbb{Z}_2$, *then* $\sum_{x_1,x_2,\ldots,x_k} F = 0$. *Suppose* $AX = 0$ *is not infeasible. Then in polynomial time, we can construct another function* $H(y_1, y_2, \ldots, y_s) = i^{L'_1(Y)+L'_2(Y)+\cdots+L'_n(Y)} \in \mathcal{A}$, *such that* $0 \le s \le k$, *and* $\sum_{x_1,x_2,\ldots,x_k} F = \sum_{y_1,y_2,\ldots,y_s} H$.

The following lemma gives a key property of the function $i^{L_1(X)+L_2(X)+\cdots+L_n(X)}$. This property plays an important role both in the tractability proof and the hardness proof.

LEMMA 5.2. *Let* $F(x_1, x_2, \ldots, x_k) = i^{L_1(X)+L_2(X)+\cdots+L_n(X)}$. *Exactly one of the following two statements hold:*

1. *(Congruity) There exists a constant* $c \in \{1, -1, i, -i\}$ *such that for all* $x_2, x_3, \ldots, x_k \in \{0, 1\}$ *we have* $F^{x_1=1}/F^{x_1=0}(x_2, x_3, \ldots, x_k) = c$;
2. *(Semi-congruity) There exists a constant* $c \in \{1, i\}$ *and an affine subspace* S *of dimension* $k-2$ *on* $T = \{(x_2, x_3, \ldots, x_k) \mid x_i \in \mathbb{Z}_2\}$, *such that* $F^{x_1=1}/F^{x_1=0}(x_2, x_3, \ldots, x_k) = c$ *on* S, *and* $F^{x_1=1}/F^{x_1=0}(x_2, x_3, \ldots, x_k) = -c$ *on* $T - S$.

PROOF. If for every $1 \le j \le n$, the coefficient for x_1 is zero in the affine linear form for $L_j(X)$, then $F^{x_1=1}/F^{x_1=0}$ is a constant 1. Otherwise, w.l.o.g. suppose the coefficients for x_1 is nonzero in exactly the first m affine linear forms $L_j(X)$. Obviously, the other $L_j(X)$'s cancel in the ratio $F^{x_1=1}/F^{x_1=0}$.

For any assignment to $x_2, x_3, \ldots, x_k$, consider the two assignments $(0, x_2, x_3, \ldots, x_k)$ and $(1, x_2, x_3, \ldots, x_k)$. For each $1 \le j \le m$, $L_j(1, x_2, x_3, \ldots, x_k) = 1 - L_j(0, x_2, x_3, \ldots, x_k)$. Therefore the ratio $F^{x_1=1}/F^{x_1=0} = \prod_{j=1}^{m} i^{1-2L_j(0,x_2,x_3,\ldots,x_k)}$ $= i^m(-1)^{\sum_{j=1}^{m} L_j(0,x_2,x_3,\ldots,x_k)}$. Here m is independent of the assignment on $x_2, x_3, \ldots, x_k$. Since the base is -1 now, the sum can be evaluated as a sum mod 2. Therefore there is an affine linear form $\alpha(X) = \sum_{\ell=2}^{k} \alpha_\ell x_\ell + \alpha_{k+1} \pmod 2$, such that $F^{x_1=1}/F^{x_1=0} = i^m(-1)^{\alpha(X)}$.

If all $\alpha_\ell = 0$, for $2 \le \ell \le k$, then this ratio is a constant and we are in the case of Congruity. If $\alpha_\ell = 1$, for some $2 \le \ell \le k$, then we have Semi-congruity. □

THEOREM 5.2. *#CSP($\mathcal{A}$) is polynomial time computable.*

PROOF. We first observe that $\mathcal{A}$ is closed under multiplication. Therefore given an instance of #CSP($\mathcal{A}$), the value of the output can be expressed as the summation on a single function $F = \chi_{AX} i^{L_1(X)+L_2(X)+\cdots+L_n(X)} \in \mathcal{A}$. We also note that if $F \in \mathcal{A}$, so is $F^{x_s=c}$ and $F^{x_s=*}$.

In each step of our algorithm, we reduce the number of variables by at least one and still get a summation of this form.

If the linear system $AX = 0$ over $\mathbb{Z}_2$ is infeasible, the function is a totally zero function and we just output 0. If $AX = 0$ is feasible then by Lemma 5.1 we can remove the factor χ_{AX} and possibly decrease the number of variables at the same time.

Now we assume it has the form $F = i^{L_1(X)+L_2(X)+\cdots+L_n(X)}$, we apply Lemma 5.2 to remove x_1. There are three cases.

Case 1: We have Congruity in Lemma 5.2. Then $F^{x_1=1}/F^{x_1=0}$ is a constant c, and

$$\sum_{x_1,x_2,\ldots,x_k} F = (1+c) \cdot \sum_{x_2,x_3,\ldots,x_k} F^{x_1=0}.$$

So we get a new summation $\sum_{x_2,x_3,\ldots,x_k} F^{x_1=0}$ and have removed a variable x_1.

Case 2: We have Semi-congruity in Lemma 5.2, and $c = 1$. Then on the affine subspace S, the ratio $F^{x_1=1}/F^{x_1=0} = 1$, and on the complementary subspace $T - S$ the ratio $F^{x_1=1}/F^{x_1=0} = -1$. For all $(x_2, x_3, \ldots, x_k) \in T - S$, the terms cancel, $F^{x_1=1}(x_2, x_3, \ldots, x_k) + F^{x_1=0}(x_2, x_3, \ldots, x_k) = 0$. On S, the terms are equal. It follows that

$$\sum_{x_1,x_2,\ldots,x_k} F = 2 \sum_{x_2,x_3,\ldots,x_k} \chi_S F^{x_1=0}.$$

Note that $\chi_S F^{x_1=0}$ is also a function in $\mathcal{A}$, so we get a new summation of this form and have removed a variable x_1.

Case 3: We have Semi-congruity in Lemma 5.2, and $c = i$. Then for all $(x_2, x_3, \ldots, x_k)$ in the affine subspace S, we have $F^{x_1=1}/F^{x_1=0} = i$, and in $T-S$, we have $F^{x_1=1}/F^{x_1=0} = -i$. It follows that

$$\sum_{x_1,x_2,\ldots,x_k} F = \sum_{S} (1+i) F^{x_1=0} + \sum_{T-S} (1-i) F^{x_1=0}.$$

Now we make a crucial observation. The ratio of $1+i$ and $1-i$ is exactly i. As a result we can rewrite the two sums as follows:

$$\sum_{x_1,x_2,\ldots,x_k} F = \sum_{S} (1-i) F^{x_1=0} i^{L(X')} + \sum_{T-S} (1-i) F^{x_1=0} i^{L(X')},$$

where $L(X')$, on $X' = (x_2, x_3, \ldots, x_k, 1)$, is a 0-1 indicator function which takes the value 1 on S and 0 on $T - S$. Thus we can combine the two sums and get

$$\sum_{x_1,x_2,\ldots,x_k} F = (1-i) \cdot \sum_{x_2,x_3,\ldots,x_k} \left(F^{x_1=0} \cdot i^{L(X')} \right).$$

Note that $F^{x_1=0} \cdot i^{L(X')}$ is also a function in $\mathcal{A}$. So we get a new summation of this form and have removed a variable x_1.

After at most k step we can eliminate all the variables and obtain the value of the initial summation. Both k and n are bounded by input size. In each iteration, we either resolve an affine linear system $AX = 0$ or compute an affine linear equation from Lemma 5.2 representing the affine linear subspace S, both of which can be done in polynomial time. And after one iteration, the formula inside the summation at most grows by a factor of $i^{L(X')}$ or χ_S. So the whole algorithm is in polynomial time. □

The proofs of the following lemmas are left to the full version.

LEMMA 5.3. *If* $[a, b, c] \notin \mathcal{A} \cup \mathcal{P}$, *#CSP*($\{[a, b, c]\}$) *is #P-hard. To be explicit, all tractable functions* $[a, b, c]$ *from* $\mathcal{A} \cup \mathcal{P}$ *have one of the following forms:* $[x, 0, y]$, $[0, x, 0]$, $[x^2, xy, y^2]$, $x[1, \pm i, 1]$ *or* $x[1, \pm 1, -1]$.

The following lemma generalizes Lemma 11 in [13] to complex weights. However the original proof does not work for complex weights, due to possible cancelations.

LEMMA 5.4. *If the support R_F is not affine, then #CSP($\{F\}$) is #P-hard.*

LEMMA 5.5. *If $F \notin \mathcal{A}$, then either #CSP($\{F\}$) is #P-hard, or we can simulate a unary function $H \notin \mathcal{A}$, that is, there is a reduction from #CSP($\{F, H\}$) to #CSP($\{F\}$).*

LEMMA 5.6. *For any function $F \notin \mathcal{P}$, either #CSP($\{F\}$) is #P-hard, or we can simulate, using F, a function $[a, 0, 1, 0]$ (or $[0, 1, 0, a]$), where $a \neq 0$, or a binary function $H \notin \mathcal{P}$ having no zero values.*

We also prove a stronger dichotomy theorem that the hardness result holds even when restricted to those #CSP instances, in which each variable occurs at most three times.

THEOREM 5.3. *If $\mathcal{F} \nsubseteq \mathcal{A}$ and $\mathcal{F} \nsubseteq \mathcal{P}$, #CSP($\mathcal{F}$) where each variable occurs at most three times (that is, #$\{=_1, =_2, =_3\}|\mathcal{F}$) is #P-hard.*

6. REFERENCES

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

[2] Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. In *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2008.

[3] Andrei A. Bulatov and Víctor Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Inf. Comput.*, 205(5):651–678, 2007.

[4] Andrei A. Bulatov and Martin Grohe. The complexity of partition functions. In *ICALP*, volume 3142 of *Lecture Notes in Computer Science*, pages 294–306. Springer, 2004.

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

[6] Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. *manuscript*, 2009.

[7] Jin-Yi Cai and Pinyan Lu. Holographic algorithms: from art to science. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 401–410, 2007.

[8] Jin-Yi Cai and Pinyan Lu. On symmetric signatures in holographic algorithms. In *STACS*, volume 4393 of *Lecture Notes in Computer Science*, pages 429–440. Springer, 2007.

[9] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by fibonacci gates and holographic reductions for hardness. In *FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 644–653, 2008.

[10] N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of boolean constraint satisfaction problems*. SIAM Monographs on Discrete Mathematics and Applications, 2001.

[11] P. Dagum and M. Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102:283–305, 1992.

[12] C. T. J. Dodson and T. Poston. *Tensor Geometry*. Graduate Texts in Mathematics 130. Springer-Verlag, New York, 1991.

[13] Martin E. Dyer, Leslie Ann Goldberg, and Mark Jerrum. The complexity of weighted boolean #csp. *CoRR*, abs/0704.3683, 2007.

[14] Martin E. Dyer, Leslie Ann Goldberg, and Mike Paterson. On counting homomorphisms to directed acyclic graphs. In *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2006.

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

[16] Martin E. Dyer and Catherine S. Greenhill. The complexity of counting graph homomorphisms (extended abstract). In *SODA*, pages 246–255, 2000.

[17] Martin E. Dyer and Catherine S. Greenhill. Corrigendum: The complexity of counting graph homomorphisms. *Random Struct. Algorithms*, 25(3):346–352, 2004.

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

[19] M. Freedman, L. Lovász, and A. Schrijver. Reflection positivity, rank connectivity, and homomorphism of graphs. *J. AMS*, 20:37–51, 2007.

[20] Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *CoRR*, abs/0804.1932, 2008.

[21] P. Hell and J. Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[22] P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.

[23] P. W. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, London, 1967.

[24] H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.

[25] Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001.

[26] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

[27] Leslie G. Valiant. Holographic algorithms (extended abstract). In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 306–315,2004.

[28] Leslie G. Valiant. Accidental algorthims. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 509–517, 2006.

[29] Mingji Xia, Peng Zhang, and Wenbo Zhao. Computational complexity of counting problems on 3-regular planar graphs. *Theor. Comput. Sci.*, 384(1):111–125, 2007.