Mapping the complexity of counting problems

Heng Guo
Queen Mary, University of London
h.guo@qmul.ac.uk

1 Computational Counting

The study of computational counting was initiated by Leslie Valiant in the late 70s. In the seminal papers [32, 33], he defined the computational complexity class **#P**, the counting counterpart of **NP**, and showed that the counting version of tractable decision problems can be **#P**-complete. One such example is counting perfect matchings (#PM). These work revealed some fundamental differences between counting problems and decision problems and stimulated a lot of research in the directions of both structural complexity theory and algorithmic design. One of the crowning results in the first direction is Toda's theorem [31], stating that **#P** contains the whole polynomial hierarchy, and in the second direction we have witnessed the success of Markov chain Monte Carlo algorithms [24, 23].

A typical counting problem is #Sat, which asks the number of satisfying assignments of a given CNF formula. An equivalent way to recast this problem is to give each assignment weight 1 if it satisfies the formula, or 0 otherwise. Then the goal is to compute the sum of all weights. Moreover, this weight can be decomposed into a product of weights of all clauses. Thus we are interested in a sum-of-product quantity (e.g. see Eq. (1) in Section 2), which is usually called the partition function.

In fact, the partition function has received immense attention by statistical physicists long before computer scientists. It is a central quantity from which one can deduce various properties of a field or a system. For example, the famous Lee-Yang theorem [27] showed the lack of phase transitions by studying the (complex) zeroes of the partition function. A goal of particular interest is to give explicit formulas of partition functions for various models [22, 28, 30, 25, 26, 1]. When such a formula is found, the model is called "exactly solved".

In computational complexity terms, exactly solvable models are tractable in the sense that we have polynomial-time algorithms to compute the partition functions. A classical gem is the Fisher-Kasteleyn-Temperley (FKT) algorithm, which counts perfect matchings over planar graphs in polynomial time [30, 25, 26].

Valiant introduced matchgates [35, 34] and holographic reductions to extend the reach of the FKT algorithm [36, 37]. These reductions differ from classical ones by introducing quantum-like superpositions. This novel technique yields polynomial time algorithms for a number of problems for which only exponential-time algorithms were previously known.

One natural question arises: what is the true power of the holographic algorithm? In particular, can we solve #**P**-hard problems by holographic algorithms? Since holographic algorithms can solve quite a few seemingly hard problems, it is difficult to rule out this possibility before giving them a systematic study. Holant problems were proposed as a natural framework to answer this question [13, 14]. The key feature of Holant problems is that they include problems like #PM which is central to the study of holographic algorithms but difficult to express in the traditional counting constraint satisfaction problems (#CSP) framework.[1]

Without settling **P** vs #**P**, we hope to answer this question by achieving computational complexity classifications. In other words, we want to map out the landscape of counting problems in terms of their intrinsic complexity and then understand the holographic algorithm by looking at its "territory". In [20], classifications are obtained for #CSP problems as well as under the Holant framework.

A preponderance of evidence suggests the following putative classification of all counting problems defined by local constraints into *exactly* three categories:

(1) those that are **P**-time solvable over general graphs;

(2) those that are **P**-time solvable over planar graphs but #**P**-hard over general graphs;

(3) those that remain #**P**-hard over planar graphs.

Moreover, category (2) usually consists of precisely those problems solvable by holographic algorithms. It has been unfailing in the classification of Tutte polynomials [39], of spin systems [9], and of #CSP [12, 21]. However, this turns out to be false for Holant problems [5], though only in a technical sense. An additional planar tractable case was found in [5], but holographic algorithms remain the most important (albeit not the only) subroutine to solve this case.

In the following we will survey several classification theorems reported in [20]. In Section 2 we review necessary definitions and notations. In Section 3 we give the theorems. At last, we give some interesting examples of holographic transformations in Section 4.

[1] It is provably impossible to express #PM in certain "vertex" models. See [19].

2 Holant Problems

A *signature grid* $\Omega = (G, \mathcal{F}, \pi)$ is a tuple, where $G = (V, E)$ is a graph, $\mathcal{F}$ is a set of functions, and π is a mapping from the vertex set V to $\mathcal{F}$. A Boolean function $f \in \mathcal{F}$ with arity k is a mapping $\{0,1\}^k \to \mathbb{C}$, and the mapping π satisfies that the arity of $\pi(v)$ (which is a function $f \in \mathcal{F}$) is the same as the degree of v for any $v \in V$. Here we may consider any function with the range of a ring rather than just $\mathbb{C}$, but we choose $\mathbb{C}$ in this survey for clarity. Let $f_v := \pi(v)$ be the function on v. An assignment σ of edges is a mapping $E \to \{0,1\}$.[2] The weight of σ is the evaluation $\prod_{v \in V} f_v(\sigma\mid_{E(v)})$, where $E(v)$ denotes the set of incident edges of v. The (counting version of) Holant problem on the instance Ω is to compute the sum of weights of all assignments; namely,

$$\mathrm{Holant}_\Omega = \sum_\sigma \prod_{v \in V} f_v(\sigma\mid_{E(v)}). \tag{1}$$

We also write $\mathrm{Holant}(\Omega; \mathcal{F})$ when we want to emphasize the function set $\mathcal{F}$.

The term Holant was first coined by Valiant in [37] to denote an exponential sum of the above form. Cai, Xia and Lu first formally introduced this framework of counting problems in [10, 11]. We can view each function f_v as a truth table, and then we can represent it by a vector in $\mathbb{C}^{2^{d(v)}}$, or a tensor in $(\mathbb{C}^2)^{\otimes d(v)}$. The vector or the tensor is called the *signature* of a function. When we say "function", we put a slight emphasis on that it is a mapping. When we say "signature", we put a slight emphasis on that it is ready to go through linear transformations. However most of the time in this article, we use the two terms "function" and "signature" interchangeably without special attention.

A Holant problem is parameterized by a set of functions.

Definition 2.1. *Let $\mathcal{F}$ be a set of functions. Define a counting problem* $\mathrm{Holant}(\mathcal{F})$*:*
Input: *A signature grid $\Omega = (G, \mathcal{F}, \pi)$;*
Output: Holant_Ω.

We will use $\mathrm{Pl\text{-}Holant}(\mathcal{F})$ to denote the problem where the input graph is planar.

The main goal here is to characterize what kind of function set $\mathcal{F}$ makes the problem $\mathrm{Holant}(\mathcal{F})$ tractable (or hard).

We use the following notations to denote some special functions. Let $=_k$ denote the equality function of arity k. Let $\textsc{ExactOne}_k$ denote the function that is one if the input has Hamming weight 1 and zero otherwise. Let $\mathcal{EO}$ be the set

[2] In general we may consider non-Boolean functions $E \to [q]$ for a positive integer $q > 2$. For simplicity in this article we focus on the Boolean case.

of $\textsc{ExactOne}_k$ functions for all integers k. Then Holant($\mathcal{EO}$) is the same as the problem of counting perfect matchings.

A function is symmetric iff its function value is preserved under any permutation of its inputs. A symmetric function f on Boolean variables can be expressed by a compact signature $[f_0, f_1, \ldots, f_k]$, where f_i is the value of f on inputs of Hamming weight i. For the Boolean domain $[2] = \{0, 1\}$, $=_k$ function has the signature $[1, 0, \ldots, 0, 1]$ with $k + 1$ entries, and $\textsc{ExactOne}_k$ has signature $[0, 1, 0, \ldots, 0]$ of $k + 1$ entries.

Multiplying a signature $f \in \mathcal{F}$ by a scaler $c \neq 0$ only changes Holant_Ω by an easy to compute factor. Thus it does not change the complexity of Holant($\mathcal{F}$). So we always view f and cf as the same signature. In other words, we consider the projective space of vectors or tensors.

We use Holant $(\mathcal{F} \mid \mathcal{G})$ to denote the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Signatures in $\mathcal{F}$ are considered as row vectors (or covariant tensors); signatures in $\mathcal{G}$ are considered as column vectors (or contravariant tensors) (see, for example [16]). Let Pl-Holant $(\mathcal{F} \mid \mathcal{G})$ denote the Holant problem over signature grids with a planar bipartite graph.

2.1 Holographic Reductions

One key technique in the study of Holant problems is holographic reductions. To introduce the idea, it is convenient to consider bipartite graphs. For a general graph, we can always transform it into a bipartite graph while preserving the Holant value as follows. For each edge in the graph, we replace it by a path of length two. (This operation is called the *2-stretch* of the graph and yields the edge-vertex incidence graph.) Each new vertex is assigned the binary $\textsc{Equality}$ signature $(=_2) = [1, 0, 1]$. Recall that Holant $(\mathcal{F} \mid \mathcal{G})$ denotes the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Hence we have that Holant($\mathcal{F}$) $\equiv_T$ Holant $(=_2 \mid \mathcal{F})$.

For a 2-by-2 matrix T and a signature set $\mathcal{F}$, define $T\mathcal{F} = \{g \mid \exists f \in \mathcal{F}$ of arity $n,\ g = T^{\otimes n} f\}$, and similarly for $\mathcal{F}T$. Whenever we write $T^{\otimes n} f$ or $T\mathcal{F}$, we view the signatures as column vectors; similarly for $fT^{\otimes n}$ or $\mathcal{F}T$ as row vectors.

Let T be an invertible 2-by-2 matrix. The holographic transformation defined by T is the following operation: given a signature grid $\Omega = (H, \pi)$ of Holant $(\mathcal{F} \mid \mathcal{G})$, for the same bipartite graph H, we get a new grid $\Omega' = (H, \pi')$ of Holant $\left(\mathcal{F}T \mid T^{-1}\mathcal{G}\right)$ by replacing each signature in $\mathcal{F}$ or $\mathcal{G}$ with the corresponding signature in $\mathcal{F}T$ or $T^{-1}\mathcal{G}$.

Theorem 2.2 (Valiant's Holant Theorem [37]). *If $T \in \mathbb{C}^{2\times 2}$ is an invertible matrix, then we have* $\text{Holant}_\Omega(\mathcal{F} \mid \mathcal{G}) = \text{Holant}_{\Omega'}(\mathcal{F}T \mid T^{-1}\mathcal{G})$.

Therefore, an invertible holographic transformation does not change the complexity of the Holant problem in the bipartite setting. By Theorem 2.2, we have that

$$\text{Holant}(\mathcal{F}) \equiv \text{Holant}\left([1,0,1]T^{\otimes 2} \mid T^{-1}\mathcal{F}\right)$$

where $T \in \mathbf{GL}_2(\mathbb{C})$ is nonsingular. This leads to the notion of $\mathscr{C}$*-transformable*.

Definition 2.3. *Let $\mathcal{F}$ and $\mathscr{C}$ be two sets of signatures. Say $\mathcal{F}$ is $\mathscr{C}$-transformable if there exists a $T \in \mathbf{GL}_2(\mathbb{C})$ such that $[1,0,1]T^{\otimes 2} \in \mathscr{C}$ and $\mathcal{F} \subseteq T\mathscr{C}$.*

The following lemma is immediate.

Lemma 2.4. *If $\mathcal{F}$ is $\mathscr{C}$-transformable, then we have the following reductions.*

$$\text{Holant}(\mathcal{F}) \leq_T \text{Holant}(\mathscr{C}); \qquad \text{Pl-Holant}(\mathcal{F}) \leq_T \text{Pl-Holant}(\mathscr{C}).$$

A consequence of the lemma above is that, if $\text{Holant}(\mathscr{C})$ (or $\text{Pl-Holant}(\mathscr{C})$) is tractable, then $\text{Holant}(\mathcal{F})$ (or $\text{Pl-Holant}(\mathcal{F})$) is tractable for any $\mathscr{C}$-transformable set $\mathcal{F}$.

2.2 Counting Constraint Satisfaction Problems

An instance of counting constraint satisfaction problems ($\#\text{CSP}(\mathcal{F})$) has the following bipartite view. We have a set of vertices standing for variables and another set for functions (or constraints). Connect a variable vertex to a constraint vertex if the variable appears in the constraint. This bipartite graph is also known as the *constraint graph*. Moreover, each variable can be viewed as an Equality function, as it forces the same value for all adjacent edges. Under this view, we see that

$$\#\text{CSP}(\mathcal{F}) \equiv_T \text{Holant}\left(\mathcal{EQ} \mid \mathcal{F}\right),$$

where $\mathcal{EQ} = \{=_1, =_2, =_3, \dots\}$ is the set of Equality functions of all arities.

The relationship between #CSP and Holant problems is the following:

$$\begin{aligned} \#\text{CSP}(\mathcal{F}) &\equiv_T \text{Holant}(\mathcal{EQ} \cup \mathcal{F}); \\ \text{Pl-}\#\text{CSP}(\mathcal{F}) &\equiv_T \text{Pl-Holant}(\mathcal{EQ} \cup \mathcal{F}). \end{aligned} \tag{2}$$

Reductions from left to right are trivial. For the other direction, we take a signature grid Ω for the problem on the right and create a bipartite signature grid Ω' for the problem on the left such that both signature grids have the same Holant value. We simply create the equivalent bipartite grid Ω'' of Ω by replace each edge with a path of length 2 with $=_2$ in the middle point, as described earlier. Then we contract all Equality signatures that are connected with each other, resulting in Ω' where Equality signatures are on one side and signatures from $\mathcal{F}$ on the other.

3 Dichotomy Theorems

In this section we survey several classification theorems. We will discuss #CSP first and then turn our attention to Holant problems. We will restrict our attention mostly to symmetric functions throughout this section.

3.1 Counting Constraint Satisfaction Problems

The first classification theorem for #CSP is the dichotomy by Creignou and Hermann [8], which is for unweighted Boolean functions. This dichotomy has been later generalized for real weights [2], and complex weights [15]. Moreover, even beyond the Boolean domain, the complexity of #CSP have been successfully classified. A complete dichotomy theorem for complex weighted functions with any constant domain was obtained after a series of research [3, 17, 4]. However, in the following we will only look at the more relevant Boolean case.

There are two basic tractable cases for complex Boolean #CSP. The first kind is called "*product-type*". A function is of the product-type if it can be decomposed into a product of unary functions, weighted equality functions, and weighted binary disequalities. The algorithm for this case is simple. We just pick an initial assignment and then propagate. The structure of this problem dictates that there are at most 2 assignments with non-zero weights for each component.

The other tractable case is called "*affine-type*". It is a generalization of the fact that we can count the number of solutions to a linear system by computing the rank of the system. (Notice that a linear system can be characterized by parity functions.) However, it is non-trivial to generalize this simple fact to complex functions. When we are dealing with complex weights, there are more potential cancellations that are beneficial to improve the efficiency. For the complete detail, see [15].

The sweeping power of dichotomy theorems is that we know the above two cases are the only tractable cases. Any other problem in this framework is #**P**-hard.

To understand the power of holographic algorithms, we want to consider #CSP defined on planar graphs. The first result on this direction is by Cai, Lu, and Xia [12], and it is later generalized to complex Boolean symmetric functions [21].

The new tractable cases in planar graphs are solved exactly by holographic algorithms. There are two main ingredients in Valiant's holographic algorithm [37]. The overall strategy is to reduce to the FKT algorithm, which counts perfect matchings in planar graphs. The first ingredient is what functions can be expressed by perfect matchings. These functions are named *matchgates*. (For a complete theory of matchgates, see [6].) The second ingredient is holographic transformations, which we have explained in Section 2.1. The crucial observation for #CSP is that, the only transformation that we need to consider is the Hadamard matrix, $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.

An informal summary of the main theorem in [21] is the following.

Theorem 3.1 (informal)**.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued functions in Boolean variables. Then* Pl-#CSP($\mathcal{F}$) *is* #**P**-*hard unless*

1. *$\mathcal{F}$ is of product-type;*
2. *$\mathcal{F}$ is of affine-type;*
3. *$\mathcal{F}$ belongs to matchgates under $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.*

In all exceptional cases Pl-#CSP($\mathcal{F}$) *is computable in polynomial time.*

3.2 Holant Problems

Because of the reduction in (2), if #CSP($\mathcal{F}$) is tractable, then Holant($\mathcal{F}$) is as well, since the Holant problem is tractable even for the larger set of functions $\mathcal{EQ} \cup \mathcal{F}$. Thus we see immediately that any tractable cases in Theorem 3.1 translates to tractable cases for Holant problems as well. However, the key feature of Holant problem is the availability of holographic transformations. Indeed, Lemma 2.4 indicates that the "transformable" set of any tractable function set is also tractable.

This is not the only difference between Holant problems and #CSP. The structure of Holant problems allows more cancellation and more potential algorithms. It is indeed the case for Boolean functions, by the discovery of "vanishing" functions [7]. Vanishing functions are constraints, that when applied to any signature grid, produce a zero Holant value. We can not introduce the whole theory here. Instead, we will illustrate the idea by going through examples.

A simple example of vanishing functions is a tensor product of the unary function $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)$, i.e., a constraint function of the form $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)^{\otimes k}$ on k variables. The function value is i^t if $0 \le t \le k$ many of the inputs are 1. This function on a vertex (of degree k) can be replaced by k copies of the unary function $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)$ on k new vertices, each connected to an incident edge. Whenever two copies of $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)$ meet in the evaluation of Holant_Ω in (1), they annihilate each other since they give the value $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right) = 0$.

Now consider a function f which is a sum of tensor products of unary functions, where in each product there are more than half $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)$'s. We view f as in a "superposition" of these tensor products. In a grid composed by f, we may first assign one of the tensor products to each vertex, then evaluate the whole grid. There are exponentially many ways to assign the products, but if we sum over all possible assignments, the Holant value is recovered. On the other hand, in each of these exponentially many terms, there are more than half $\left(\begin{smallmatrix} 1 & i \end{smallmatrix}\right)$'s and at least two of them meet. It results in making the whole evaluation 0. This argument is valid for any assignment. In summary, we managed to rewrite the Holant sum into a sum of exponentially many terms, each of which is 0. Hence this function is vanishing. In general, it is shown that all vanishing functions are generalizations of the kind we described here [7].

These ghostly vanishing functions are like the elusive dark matter. They do not actually contribute any value to the Holant sum. However in order to give a complete dichotomy for Holant problems, it turns out to be essential that we capture these vanishing functions. There is another similarity with dark matter. Their contribution to the Holant sum is not directly observed. Yet in terms of the dimension of the algebraic variety they constitute, they make up the vast majority of the tractable symmetric functions. Furthermore, when combined with others, they provide a large substrate to produce non-vanishing and tractable function sets.

Similarly to vanishing functions, there is an extra tractable case when we move from planar #CSP to planar Holant problems. The complete description is quite technical and may require several pages. We just briefly explain the idea here. The main cause for the tractability is the structure of planar graphs. We perform some "global" operation to find edges that have to be fixed to 0 or 1. Such edges must exist due to planarity, unless the instance has already fall into one of the known tractable cases. Thus, the strategy is to do this "fixing" move until it is solvable by known algorithms. This is a strange case in that it escapes the usual formulation of holographic algorithms, yet holographic algorithms are the main non-trivial ingredient once the "fixing" is done.

Informally, the main theorems of [7, 5] can be summarized as follows.

Theorem 3.2 (informal)**.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued functions in Boolean variables. Then* Pl-Holant($\mathcal{F}$) *is* #**P**-*hard unless*

1. *$\mathcal{F}$ is transformable to product-type functions;*
2. *$\mathcal{F}$ is transformable to affine-type functions;*
3. *$\mathcal{F}$ is tractable due to vanishing functions;*
4. *$\mathcal{F}$ is transformable to matchgates;*
5. *$\mathcal{F}$ belongs to the extra planar tractable case.*

In all exceptional cases Pl-Holant($\mathcal{F}$) *is computable in polynomial time. If $\mathcal{F}$ belongs to cases 1,2,3, then* Holant($\mathcal{F}$) *is tractable, and otherwise* Holant($\mathcal{F}$) *is* #**P**-*hard.*

4 Holographic Transformations

Holographic transformations are the central technique to obtain Theorem 3.1 and Theorem 3.2. In this last section, let us see some holographic transformations in action.

4.1 Ising model and the "subgraphs" world

We first review a classical equivalence between the Ising model and even subgraphs. It was observed as early as in [38] and has been rediscovered several times. In the seminal approximation algorithm for the Ising model [24], this equivalence played a central role. We show that it can be easily obtained using the "modern" language of holographic transformations.

The partition function of a ferromagnetic Ising model on a graph $G = (V, E)$ with parameter β is defined as

$$Z_{Ising}(\beta) = \sum_{\sigma \in \{0,1\}^V} \beta^{m(\sigma)}.$$

where $\sigma \in \{0, 1\}^V$ is an assignment of vertices and $m(\sigma)$ is the number of monochromatic edges (either (0,0) or (1,1)) under σ. On the other hand, a subgraph $S \subseteq E$ is called *even* if every vertex in the induced subgraph (V, S) has an even degree. Denote by Ω_{even} the state space of all such even subgraphs of G. Define the partition function with parameter p:

$$Z_{even}(p) = \sum_{S \in \Omega_{even}} p^{|S|}(1 - p)^{|E \setminus S|}.$$

We will show that

$$Z_{Ising}(\beta) = 2^{|V|}\beta^{|E|} Z_{even}\left(\frac{1}{2}\left(1 - \frac{1}{\beta}\right)\right). \tag{3}$$

Let Ising be a binary function so that

$$\textsc{Ising}(x_1, x_2) := \begin{cases} \beta & \text{if } x_1 = x_2; \\ 1 & \text{otherwise.} \end{cases}$$

Using our notation, Ising is $[\beta, 1, \beta]$. Then we may express Z_{Ising} as Equality functions on vertices and the Ising function on edges; namely, Holant($\mathcal{EQ}$ | Ising).

The even subgraph constraint can be formulated as choosing a subset of edges subject to the following Even constraint on vertices (of degree d):

$$\textsc{Even}_d(x_1, \cdots, x_d) := \begin{cases} 1 & \text{if } \bigoplus_i x_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

Choosing edges can be interpreted as binary EQUALITY functions $[1, 0, 1]$ on edges (cf. Section 2.2). Moreover, the parameter p makes these functions weighted. Define WEQ as the binary weighted Equality function $[1 - p, 0, p]$. Then Z_{Even} is just $\text{Holant}\,(\text{Even}_1, \text{Even}_2, \cdots \mid \text{WEQ})$. To recover (3), set $p = \frac{1}{2}\left(1 - \frac{1}{\beta}\right)$.

To see the equivalence (3), do a holographic transformation by $H = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$:

$$\text{Holant}\,(\mathcal{EQ} \mid \text{Ising}) \equiv \text{Holant}\left(\mathcal{EQ}H \mid (H^{-1})^{\otimes 2}\text{Ising}\right).$$

Now we just need to verify

$$\begin{aligned}(=_d)H^{\otimes d} &= 2\text{Even}_d;\\ (H^{-1})^{\otimes 2}\text{Ising} &= \beta\text{WEQ}.\end{aligned}$$

The first line can be verified as

$$\begin{aligned}(=_d)H^{\otimes d} &= \left([1, 0]^{\otimes d} + [0, 1]^{\otimes d}\right)\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]^{\otimes d}\\ &= [1, 1]^{\otimes d} + [1, -1]^{\otimes d}\\ &= [2, 0, 2, 0, \dots] = 2\text{Even}_d.\end{aligned}$$

Noticing that $H^{-1} = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, the second line can be verified as

$$\begin{aligned}(H^{-1})^{\otimes 2}\text{Ising} &= \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}\begin{pmatrix} \beta \\ 1 \\ 1 \\ \beta \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 2\beta + 2 \\ 0 \\ 0 \\ 2\beta - 2 \end{pmatrix}\\ &= \beta\left[\frac{1}{2}\left(1 + \frac{1}{\beta}\right), 0, \frac{1}{2}\left(1 - \frac{1}{\beta}\right)\right]\\ &= \beta[1 - p, 0, p] = \beta\text{WEQ}.\end{aligned}$$

4.2 Shearer's Condition

Lovász local lemma [18] is an important tool in combinatorics. Shearer [29] gave the optimal condition for the Lovász local lemma on a fixed dependency graph $G = (V, E)$. Verifying Shearer's condition is a non-trivial task and usually boils down to deciding whether the independence polynomial is positive for G, where each vertex has a negative weight. Let $\mathbf{p} = (p_v)_{v \in V}$ be the set of weights. Let $\mathcal{I}$ be the collection of independent sets of G. Define the following quantity:

$$q(\mathbf{p}) := \sum_{I \in \mathcal{I}} (-1)^{|I|} \prod_{v \in I} p_v.$$

We are interested in whether $q(\mathbf{p})$ is positive or not.

Here we discuss a particular example such that $p_v = 2^{-d_v}$ where d_v is the degree of vertex v.[3] We will see that $q(\mathbf{p}) > 0$ if and only if G is a tree. Holographic transformations will play a central role in the proof below.

We first claim that $q(\mathbf{p})$ is equivalent to

$$\operatorname{Holant}\left([1, 1/2, 0] \mid [1, 0, \ldots, 0, -1]\right).$$

In this bipartite view, variables are half-edges of the graph. If v is chosen in the independent set, then all of its adjacent half-edges are assigned 1. Since we are taking an independent sets, we cannot take both endpoints simultaneously. This is why the edge function forbids $(1, 1)$. The weight of 2^{-d_v} can be viewed as distributed to edges adjacent to v, where each edge gets $1/2$. If a vertex is chosen, it also contributes an extra -1 factor to $q(\mathbf{p})$. Hence the Equality function on the right hand side is weighted.

Let $H = H^{-1} = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$ be the Hadamard matrix. Then we can do the holographic transformation by H.

$$\begin{aligned}
&\operatorname{Holant}\left([1, 1/2, 0] \mid [1, 0, \ldots, 0, -1]\right) \\
\equiv\; &\operatorname{Holant}\left([1, 1/2, 0](H^{-1})^{\otimes 2} \mid H^{\otimes d_v}[1, 0, \ldots, 0, -1]\right) \\
\equiv\; &\operatorname{Holant}\left([1, 1/2, 0](H^{-1})^{\otimes 2} \mid H^{\otimes d_v}[1, 0, \ldots, 0, -1]\right) \\
\equiv\; &\operatorname{Holant}\left([1, 1/2, 0] \mid 2^{1-d_v/2}\textsc{Odd}_{d_v}\right),
\end{aligned}$$

where $\textsc{Odd}_{d_v}$ denotes the function of arity d_v that is satisfied only if an odd number of input variables is 1. This can be easily verified as

$$\begin{pmatrix} 1 & 1/2 & 1/2 & 0 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1/2 & 1/2 & 0 \end{pmatrix},$$

and

$$\begin{aligned}
H^{\otimes d_v}[1, 0, \ldots, 0, -1] &= 2^{-d_v/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes d_v} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes d_v} - \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes d_v} \right) \\
&= 2^{-d_v/2} \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}^{\otimes d_v} - \begin{pmatrix} 1 \\ -1 \end{pmatrix}^{\otimes d_v} \right) = 2^{1-d_v/2}\textsc{Odd}_{d_v}.
\end{aligned}$$

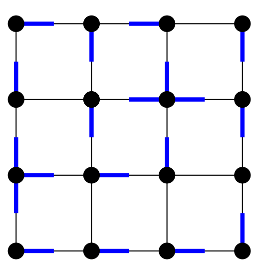
Notice that all functions in $\operatorname{Holant}\left([1, 1/2, 0] \mid \textsc{Odd}_{d_v}\right)$ are non-negative. This implies that $q(\mathbf{p}) \geq 0$ for any graph G.

[3] For those who are interested, this is the setting of sink-free orientations. Random variables are directions of edges and sinks are "bad" events.

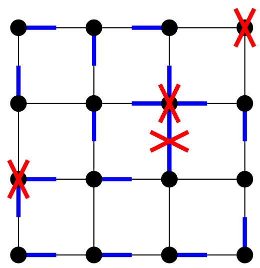
Let Holant(G) denote the partition function of Holant $([1, 1/2, 0] \mid \textsc{Odd}_{d_v})$ on a graph G. Due to the discussion above, we have that

$$q(\mathbf{p}) = \prod_{v \in V} 2^{1-d_v/2} \,\mathrm{Holant}(G) = 2^{n-m} \,\mathrm{Holant}(G), \tag{4}$$

where $n = |V|$ and $m = |E|$.



(a) A satisfying assignment



(b) An unsatisfying assignment

Figure 1: An example of the constraint satisfaction problem on G. Blue half-edges are assigned 1. Red crosses mark constraints that are not satisfied.

Consider the following constraint satisfaction problem on the graph G. Assign each half-edge 0 or 1, but not both 1. Denote the assignment by σ. For each vertex v, we require odd number of adjacent half-edges assigned 1. An example can be found in Figure 1. Let S be the set of satisfying assignments. It is easy to see that

$$\mathrm{Holant}(G) = \sum_{\sigma \in S} 2^{-|\sigma|}, \tag{5}$$

where $|\sigma|$ is the number of half-edges assigned 1.

Theorem 4.1. *Let G be a connected graph. If G is a tree, then $q(\boldsymbol{p}) = 0$; otherwise $q(\boldsymbol{p}) > 0$.*

Proof. By (4) and (5), $q(\mathbf{p}) > 0$ if and only if S is not empty.

If G is a tree, then $S = \emptyset$. To see this, choose an arbitrary vertex v as the root of G. All leaves of G has degree 1. Hence their adjacent half-edges must be assigned 1. Remove all leaves. Again the new leaves also force their adjacent half-edges to be 1. Repeat this procedure until we have only the root v left. Then there is no assignment to satisfy the odd parity constraint of v.

Otherwise G is not a tree, and there exists a cycle C in G. We will construct a satisfying assignment. Pick an arbitrary half-edge in C and assign 1 to it. Then we

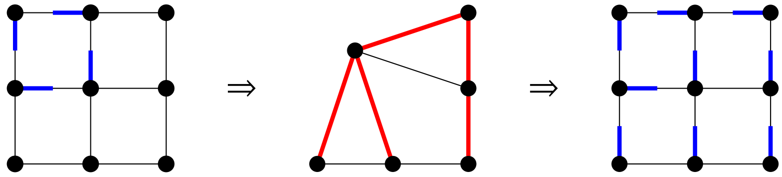


Figure 2: An example of constructing the constraint satisfaction problem on a 3-by-3 grid: (1) choose the top-left cycle and contract; (2) find a spanning tree of the new graph (red); (3) map the assignment back to the original.

may follow C to assign 1's so that every vertex in C has exactly one adjacent half-edge chosen. Contract all vertices on C to one vertex v and get a new graph G'. Pick an arbitrary spanning tree T of G'. Similarly to the last case, we may assign 1 to half-edges adjacent to leaves recursively until v is left. Edges that are not in T are assigned $(0, 0)$ to its half-edges. This gives an assignment of all edges in G. It is not hard to verify that in this assignment, all vertices are adjacent to exactly one chosen half-edge. Hence this is a satisfying assignment and $q(\mathbf{p}) > 0$. $\square$

References

[1] R.J. Baxter. *Exactly solved models in statistical mechanics*. Academic press London, 1982.

[2] A. Bulatov, M. Dyer, L.A. Goldberg, M. Jalsenius, and D. Richerby. The complexity of weighted boolean #CSP with mixed signs. *Theoretical Computer Science*, 410(38-40):3949–3961, 2009.

[3] Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2008.

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

[5] Jin-Yi Cai, Zhiguo Fu, Heng Guo, and Tyson Williams. A Holant dichotomy: Is the FKT algorithm universal? In *FOCS*, pages 249–260, 2015.

[6] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory Comput.*, 10(7):167–197, 2014.

[7] Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. In *STOC*, pages 249–260, 2013.

[8] Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.

[9] Jin-Yi Cai and Michael Kowalczyk. Spin systems on k-regular graphs with complex edge functions. *Theor. Comput. Sci.*, 461:2–16, 2012.

[10] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *FOCS*, pages 644–653, 2008.

[11] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In Michael Mitzenmacher, editor, *STOC*, pages 715–724. ACM, 2009.

[12] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar #CSP. In *FOCS*, pages 427–436, 2010.

[13] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM J. Comput.*, 40(4):1101–1132, 2011.

[14] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic reduction, interpolation and hardness. *Computational Complexity*, 21(4):573–604, 2012.

[15] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *J. Comput. System Sci.*, 80(1):217–236, 2014.

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

[17] Martin E. Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013.

[18] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets, volume 10 of Colloquia Mathematica Societatis János Bolyai*, pages 609–628, 1975.

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

[20] Heng Guo. *Complexity Classification of Exact and Approximate Counting Problems*. PhD thesis, University of Wisconsin-Madison, 2015.

[21] Heng Guo and Tyson Williams. The complexity of planar Boolean #CSP with complex weights. In *ICALP*, pages 516–527, 2013.

[22] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[23] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[24] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

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

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

[27] T.D. Lee and C.N. Yang. Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model. *Physical Review*, 87(3):410–419, 1952.

[28] L. Onsager. Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117–149, 1944.

[29] James B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985.

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

[31] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[32] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

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

[34] Leslie G. Valiant. Expressiveness of matchgates. *Theor. Comput. Sci.*, 289(1):457–471, 2002.

[35] Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002.

[36] Leslie G. Valiant. Accidental algorthims. In *FOCS*, pages 509–517, 2006.

[37] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[38] Bartel L. van der Waerden. Die lange Reichweite der regelmäßigen Atomanordnung in Mischkristallen. *Zeitschrift für Physik*, 118(7):473–488, 1941.

[39] Dirk Vertigan. The computational complexity of Tutte invariants for planar graphs. *SIAM J. Comput.*, 35(3):690–712, 2005.