2008 49th Annual IEEE Symposium on Foundations of Computer Science

Holographic Algorithms by Fibonacci Gates and Holographic Reductions for Hardness

Jin-Yi Cai* Pinyan Lu† Mingji Xia‡

Abstract

We propose a new method to prove complexity dichotomy theorems. First we introduce Fibonacci gates which provide a new class of polynomial time holographic algorithms. Then we develop holographic reductions. We show that holographic reductions followed by interpolations provide a uniform strategy to prove #P-hardness.

1 Introduction

The study of counting problems and their classifications is a major theme in computational complexity theory. Some counting problems are computable in P, while others appear hard. Valiant introduced the class #P to capture most of these counting problems [15]. Some well known examples in this class of problems are counting perfect matchings, and counting vertex covers. Over the past several years a uniform framework to address a large class of counting problems has emerged [5, 10, 2].

Consider the problem of counting all vertex covers on a graph $G = (V, E)$. One way to express this problem is as follows: For every edge $(x, y) \in E$ we attach an OR function on two bits, and consider all 0-1 assignments σ of the vertex set V. The OR function is represented by its truth table $F = (0, 1, 1, 1)$, which is called a "signature". Then σ is a vertex cover iff $\prod_{(x,y)\in E} F(\sigma(x), \sigma(y)) = 1$, and the total number of vertex covers is $\sum_\sigma \prod_{(x,y)\in E} F(\sigma(x), \sigma(y))$.

This framework can be generalized to H-colorings or H-homomorphisms [10]. Here H is a fixed directed or undirected graph (with possible self loops) given by a Boolean adjacency matrix. A mapping $\sigma : V(G) \to V(H)$ is a homomorphism iff for every edge $(x, y) \in E(G)$, $H(\sigma(x), \sigma(y)) = 1$. Then the quantity $\sum_\sigma \prod_{(x,y)\in E(G)} H(\sigma(x), \sigma(y))$ counts the number of H-homomorphisms. Vertex cover is the special case where the two-vertex graph $H = (\{0, 1\}, \{(0, 1), (1, 0), (1, 1)\})$. Dichotomy theorems for H-coloring problems with undirected H and directed acyclic H are given in [10] and [9] respectively.

When it comes to matchings or perfect matchings, the more natural framework will be to consider assignments to the edge set of G instead of the vertex set, and the "evaluation" F happens at each vertex, which is either a Boolean OR function (for matchings) or the EXACT-ONE function (for perfect matchings). Thus a Boolean assignment σ of E is a matching (resp. a perfect matching) iff at every vertex v the assignment σ at the incident edges $E(v)$ evaluates to 1 according to F, and the sum $\sum_{\sigma \text{ on } E} \prod_{v\in V} F(\sigma |_{E(v)})$ is the number of matchings or perfect matchings, respectively.

We remark that assigning values on edges can be viewed as a generalization of assigning values on vertices. To see this, let's temporarily consider the following further generalization where we assign a value at each *end* of an edge $e = (x, y)$, i.e., we assign a value $\sigma(e, x)$ and $\sigma(e, y)$. Then we may attach an "evaluation" function F at each edge as well as at each vertex. The overall evaluation is done for all $v \in V$ and all $e \in E$, and $\sum_\sigma \prod_{v,e} F$, the sum over all σ of products over all v and e, is then *the counting problem*. In this set-up, evaluating over vertex assignments is the special case where F at each vertex is the EQUALITY function, and evaluating over edge assignments is the special case where F at each edge is the EQUALITY function. However, we claim that this further generalization can be easily simulated by the following construction, which remains in the framework of edge assignments: Replace each edge by a path of length two and introduce a new vertex of degree 2 in the middle. This substitution makes G a bipartite graph, where on one side every vertex (the new ones) has degree 2. In this paper we will study our counting problems in the framework of edge assignments.

It turns out that studying counting problems in this

*Computer Sciences Department, University of Wisconsin-Madison, and Radcliffe Institute, Harvard University jyc@cs.wisc.edu. Supported by NSF CCR-0511679 and a Radcliffe Fellowship.

†Institute for Theoretical Computer Science, Tsinghua University lpy@mails.tsinghua.edu.cn. Supported by the National Natural Science Foundation of China Grant 60553001 and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

‡Institute of Software, Chinese Academy of Sciences, xmjljx@gmail.com. Supported by the NSFC Grant no. 60325206 and no. 60310213.

0272-5428/08 $25.00 © 2008 IEEE
DOI 10.1109/FOCS.2008.34

IEEE computer society

framework has a close connection with holographic algorithms and reductions. Holographic algorithms have been introduced by Valiant [17]. This beautiful theory has two main ingredients. The first is the use of matchgates to encode computations, which allows a P-time computation over planar graphs using the FKT method [12, 13] in terms of Pfaffians. The second ingredient is to use linear algebra to create exponential sums of perfect matchings in a "holographic mix", and achieve exponential cancelations in the process. Here we introduce another family of P-time computable primitives called Fibonacci gates. Holographic transformations with Fibonacci gates also create exponential cancelations to yield P-time algorithms.

We apply holographic algorithms and reductions to classify which counting problems are in P and which are #P-complete in the framework discussed above. In order to obtain clearly stated results we restrict our attention here to the class of 2-3 regular graphs. A 2-3 regular graph is a bipartite graph $G = (U, V, E)$, where $\deg(u) = 2$ and $\deg(v) = 3$ for all $u \in U$ and $v \in V$. As indicated above, evaluating over edge assignments for this class of graphs already encompasses all 3-regular graphs. The reason for this restriction is that (a) in this simplest case we can already show #P-completeness, and (b) we can exhibit a dichotomy theorem. Our method can be generalized to non-Boolean assignments and signatures on arbitrary graphs, and will reported in future work.

Our main technical contributions are as follows. Over the class of 2-3 regular graphs we will consider each vertex $u \in U$ (resp. $v \in V$) is given a Boolean signature, $[x_0, x_1, x_2]$ (resp. $[y_0, y_1, y_2, y_3]$). This notation (see [17]) means that at $u \in U$ of degree 2, a Boolean function F takes the value x_0, x_1 and x_2 respectively when the Hamming weight of the Boolean assignment at its two incident edges are $0, 1$ and 2 respectively. The meaning of the signature $[y_0, y_1, y_2, y_3]$ at $v \in V$ is similar. We denote by $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ the counting problem over all 2-3 regular graphs using these signatures. Our starting point is the observation that both $\#[0, 1, 1]|[1, 0, 0, 1]$ and $\#[1, 0, 1]|[1, 1, 0, 0]$ are #P-complete. (Perceptive readers will notice that $\#[0, 1, 1]|[1, 0, 0, 1]$ is just counting vertex covers, and $\#[1, 0, 1]|[1, 1, 0, 0]$ is counting matchings, both over 3-regular graphs [19].) To consider a general counting problem $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$, we apply *holographic reductions* to transform either the signature $[1, 0, 0, 1]$ or the signature $[1, 1, 0, 0]$ to the signature $[y_0, y_1, y_2, y_3]$. This uses some signature theory of holographic algorithms [3, 4]. Under this holographic reduction, the signatures $[0, 1, 1]$ or $[1, 0, 1]$ respectively are transformed to some new signature $[x'_0, x'_1, x'_2]$. This transformation will be an invertible map which shows that the counting problem $\#[x'_0, x'_1, x'_2]|[y_0, y_1, y_2, y_3]$ has the same complexity as either $\#[0, 1, 1]|[1, 0, 0, 1]$ or $\#[1, 0, 1]|[1, 1, 0, 0]$, thus #P-complete.

Next we try to show that our given signature pairs $[x_0, x_1, x_2]$ and $[y_0, y_1, y_2, y_3]$ can simulate $[x'_0, x'_1, x'_2]$. To do this we develop an algebraic lemma, and apply the powerful technique of *interpolation* initiated by Valiant [15]. The lemma gives a sufficient condition for this interpolation to succeed. The proof of this lemma uses some basic Galois theory. The actual interpolation is accomplished by a couple of versatile combinatorial gadgets (but the theory is strong enough that the particular gadgets are almost generic.) When this interpolation succeeds, we will have proved that the counting problem $\#[x_0, x_1, x_2] \mid [y_0, y_1, y_2, y_3]$ is #P-complete. All our hardness results are proved by this single universal strategy.

Along the way we will discover that for certain cases of signature pairs $[x_0, x_1, x_2]$ and $[y_0, y_1, y_2, y_3]$ this hardness proof via interpolation does not work. Then we will see that these cases are in fact computable in P. They come in three categories: (1) They can be solved by matchgates over planar graphs; (2) They can be solved by Fibonacci gates over general graphs; and (3) Some special cases solvable in P for obvious reasons. This gives us a dichotomy theorem. To sum up we show that *holographic reductions followed by interpolatability imply hardness*. In the class of problems we considered the converse is also true, namely failure to interpolate also implies solvability in P. Due to space limit, many proof details are omitted and will be presented in the full paper.

2 Definitions and Background

A *signature grid* $\Omega = (G, \mathcal{F})$ is a tuple, where $G = (V, E)$ is a graph, and each $v \in V(G)$ is assigned a function $F_v \in \mathcal{F}$. A Boolean 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

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

(The term Holant was first introduced by Valiant in [17] to denote a related exponential sum.) We can view each function F_v as a truth table, and then we can represent it by a vector in $\mathbf{F}^{2^{d(v)}}$, or a tensor in $(\mathbf{F}^2)^{\otimes d(v)}$, over some field $\mathbf{F}$. This is called a *signature*.

As discussed in the previous section, many important counting problems can be viewed as computing Holant_Ω for appropriate signatures at each vertex, such as counting (perfect) matchings and counting vertex covers. Many counting problems not directly defined in terms of graphs can also be formulated as holant problems, e.g., the #SAT problem.

In this paper we will mainly consider symmetric signatures. A signature is called symmetric, if each signature entry only depends on the Hamming weight of the input. The signatures we defined above for matching or perfect matching or Boolean OR all have this property. We use a more compact notation $[f_0, f_1, \ldots, f_n]$ to denote a symmetric signature on n inputs, where f_i is the value on inputs of weight i.

2.1 $\mathcal{F}$-Gate

A signature from $\mathcal{F}$ at a vertex is considered as a basic realizable function. Instead of a single vertex, we can use graph fragments to generalize this notion. An $\mathcal{F}$-gate Γ is a pair $(H, \mathcal{F})$, where $H = (V, E, D)$ is a graph with some dangling edges D. Other than these dangling edges, an $\mathcal{F}$-gate is the same as a signature grid. The role of dangling edges is similar to that of external nodes in Valiant's notion [16], however we allow more than one dangling edges for a node. In $H = (V, E, D)$ each node is assigned a function in $\mathcal{F}$ (we do not consider "dangling" leaf nodes at the end of a dangling edge among these), E are the regular edges, denoted as $1, 2, \ldots, m$, and D are the dangling edges, denoted as $m+1, m+2, \ldots, m+n$. Then we can define a function for this $\mathcal{F}$-gate $\Gamma = (H, \mathcal{F})$,

$$\Gamma(y_1, y_2, \ldots, y_n) = \sum_{x_1 x_2 \cdots x_m} H(x_1 x_2 \cdots x_m y_1 y_2 \cdots y_n),$$

where $(y_1, y_2, \ldots, y_n) \in \{0,1\}^n$ denotes an assignment on the dangling edges and $H(x_1 x_2 \cdots x_m y_1 y_2 \cdots y_n)$ denotes the value of the signature grid on an assignment of all edges. We will call this function the signature of the $\mathcal{F}$-gate Γ. An $\mathcal{F}$-gate can be used in a signature grid as if it is just a single node with the particular signature. We note that even for a very simple signature set $\mathcal{F}$, the signatures for all $\mathcal{F}$-gates can be quite complicated and expressive. Matchgate signatures are an example.

2.2 Holographic Reduction

To introduce the idea of holographic reductions, it is convenient (but not necessary) to consider bipartite graphs. We note that 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 on 2 inputs.

We use $\#\mathcal{G}|\mathcal{R}$ to denote all the 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 $\mathcal{G}$ or $\mathcal{R}$, respectively. An input instance of the holant problem is a signature grid and is denoted as $\Omega = (H, \mathcal{G}|\mathcal{R})$. Signatures in $\mathcal{G}$ are called generators, which are denoted by column vectors (or contravariant tensors); signatures in $\mathcal{R}$ are called recognizers, which are denoted by row vectors (or covariant tensors).

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 $\#\mathcal{G}|\mathcal{R}$ and $\#\mathcal{G}'|\mathcal{R}'$ are two holant problems defined for the same family of graphs, and $T \in \mathbf{GL}_2(\mathbf{C})$ is a basis. We say that there is a holographic reduction from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$, if the *contravariant* transformation $G' = T^{\otimes g} G$ and the *covariant* transformation $R = R' T^{\otimes r}$ map $G \in \mathcal{G}$ to $G' \in \mathcal{G}'$ and $R \in \mathcal{R}$ to $R' \in \mathcal{R}'$, 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 (Holant Theorem)**.** *Suppose there is a holographic reduction from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$ mapping signature grid Ω to Ω', then* $\text{Holant}_\Omega = \text{Holant}_{\Omega'}$.

The proof of this theorem follows from general principles of contravariant and covariant tensors. In particular, for invertible holographic reductions from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$, one problem is in P iff the other one is, and similarly one problem is #P-complete iff the other one is also.

2.3 Related Work

Our counting problems are closely related to Constrained Satisfaction Problems (CSP). A uniform treatment of CSP is given in [6] by Creignou, Khanna and Sudan. When $\mathcal{R}$ is fixed to be the set of EQUALITY of all arities, $\#\mathcal{G}|\mathcal{R}$ is called a #Weighted CSP problem. The following table lists some known dichotomy theorems about the complexity of some subclasses of #Weighted CSP.

Domain	Range	Arity	Num. of functions	Name in literature	Ref.
Boolean	Boolean	any	arbitrary	#Boolean CSP	[5]
any finitary	Boolean	two	one symmetric	#H-coloring	[10]
any finitary	non-negative	two	one symmetric	partition function	[2]
any finitary	Boolean	two	one acyclic	#H-coloring	[9]
Boolean	non-negative	any	arbitrary	#Weighted Boolean CSP	[8]
any finitary	Boolean	any	arbitrary	#CSP	[1]
any finitary	real	two	one symmetric	partition function	[11]

When $\mathcal{G}$ and $\mathcal{R}$ contain some functions other than EQUALITY, $\#\mathcal{G}|\mathcal{R}$ become our typical graph counting problems. Results on #Weighted CSP do not cover $\#\mathcal{G}|\mathcal{R}$ problems when EQUALITY is not (implicity assumed to be) present.

3 Fibonacci Gates

In this section, we introduce a set of signatures called Fibonacci gates. Then we give a polynomial time algorithm for holant problems on these signatures. A preliminary form of this idea was studied in [19].

Let $\{f_i\}_{i=0}^n$ be a sequence, satisfying $f_{k+2} = f_{k+1} + f_k$ for all $k = 0, 1, \ldots, n-2$. For any initial values f_0 and f_1, such a sequence will be called a Fibonacci sequence. For any arity n a Fibonacci sequence defines a symmetric signature $F = [f_0, f_1, \ldots, f_n]$. This defines a function on n Boolean inputs $F : \{0,1\}^n \to \mathbf{F}$ such that $F(\sigma) = f_{\mathrm{wt}(\sigma)}$, for all $\sigma \in \{0,1\}^n$. We call such functions Fibonacci gates or Fibonacci signatures. We use $\mathcal{F}$ to denote all the Fibonacci signatures.

Theorem 3.1. *For any graph H, the holant problem $\#(H, \mathcal{F})$ can be computed in polynomial time.*

Proof: If $H_1, H_2, \ldots, H_l$ are all the connected components of a graph H, then $\mathrm{Holant}_H = \prod_{i=1}^{l} \mathrm{Holant}_{H_i}$. So we only need to consider connected graphs as inputs.

Suppose H has n nodes and m edges. First we cut all the edges in H. A node with degree d can be viewed as an $\mathcal{F}$-gate with d dangling edges. Now step by step we connect two dangling edges into one regular edge in the original graph, until we recover H after m steps. Our plan is to prove that all the intermediate $\mathcal{F}$-gates still have Fibonacci signatures and at every step we can compute the intermediate signature (we only need to compute the first two values of the signature) in P. Finally we get H, an $\mathcal{F}$-gate without any dangling edges, its signature (only one value) is the holant we want to compute. To carry out this plan, we only need to prove that it is true for one single step. There are two cases, depending on whether the two dangling edges to be connected are in the same component or not.

In the first case, the two dangling edges belong to two components before their merging. Let F have dangling edges $y_1, \ldots, y_s, z$ and G have dangling edges $y_{s+1}, \ldots, y_{s+t}, z'$. After merging z with z', we have a new gate H with dangling edges $y_1, \ldots, y_s, \ldots, y_{s+t}$. Inductively the signatures of gates F and G are both Fibonacci functions. We show that the resulting gate H also has a Fibonacci signature.

Let's prove H is symmetric. We only need to show that the value of H is not changed if the value of two inputs are exchanged. Because F and G are symmetric, if both inputs are from $\{y_1, \ldots, y_s\}$ or from $\{y_{s+1}, \ldots, y_{s+t}\}$, the value of H is clearly not changed. Suppose one input is from $\{y_1, \ldots, y_s\}$ and the other is from $\{y_{s+1}, \ldots, y_{s+t}\}$. By symmetry of F and G we may assume these two inputs are y_1 and y_{s+1}. Thus we will fix an arbitrary assignment for $y_2, \ldots, y_s, y_{s+2}, \ldots, y_{s+t}$, and we want to show $H(0, y_2, \ldots, y_s, 1, y_{s+2}, \ldots, y_{s+t}) = H(1, y_2, \ldots, y_s, 0, y_{s+2}, \ldots, y_{s+t})$.

We can suppress the fixed $y_2, \ldots, y_s, y_{s+2}, \ldots, y_{s+t}$ and denote $F_{y_1 z} = F(y_1, y_2, \ldots, y_s, z)$, $G_{y_{s+1} z} = G(y_{s+1}, y_{s+2}, \ldots, y_{s+t}, z)$, and $H_{y_1 y_{s+1}} = H(y_1, \ldots, y_s, y_{s+1}, \ldots, y_{s+t})$. Then by the definition of Holant, $H_{ab} = F_{a0} G_{b0} + F_{a1} G_{b1}$, for $a, b \in \{0,1\}$. Because F and G are Fibonacci functions, $F_{11} = F_{01} + F_{00}$ and $G_{11} = G_{01} + G_{00}$. By symmetry of F and G, it follows easily $H_{01} = H_{10}$.

Now we show that $H(y_1, \ldots, y_{s+t})$ is also a Fibonacci function. Since we have proved that H is symmetric, we can choose any two inputs to prove it being Fibonacci. Again, we choose y_1 and y_{s+1}. For any fixed value of all the other inputs, we have $H_{00} = F_{00} G_{00} + F_{01} G_{01}$, $H_{01} = F_{00} G_{10} + F_{01} G_{11}$, and $H_{11} = F_{10} G_{10} + F_{11} G_{11}$. Now using the fact that both F and G are Fibonacci functions, it is easy to show that $H_{00} + H_{01} = H_{11}$.

If the first two terms of the signatures of F and G are f_0, f_1 and g_0, g_1 respectively, then the first two terms of the signature H can be easily computed as follows: $h_0 = f_0 g_0 + f_1 g_1$ and $h_1 = f_1 g_0 + f_2 g_1 = f_1 g_0 + (f_0 + f_1) g_1$.

Next we consider the second case, where the two dangling edges to be merged are in the same component. In this case, obviously the signature for the new gate H is also symmetric. If $F = [f_0, f_1, \ldots, f_n]$ is the Fibonacci signature before the merging operation, then the signature after the merging operation is $H = [f_0 + f_2, f_1 + f_3, \ldots, f_{n-2} + f_n]$. It follows that H is also Fibonacci and we have already computed its signature. ∎

Definition 3.1. *A generator G (resp. recognizer R) with arity n is realizable as a Fibonacci gate on basis T iff there exists a Fibonacci signature F such that $F^{\mathrm{T}} = T^{\otimes n} G$ (respectively $R = F T^{\otimes n}$). (Here F is written as a 2^n dimensional row vector, and F^{T} is its transpose.)*

4 Realizability

In this section, we characterize all holant problems which can be solved by holographic algorithms with Fibonacci gates.

Let ϕ (the golden ratio) and $\bar{\phi}$ be the two roots of $X^2 - X - 1 = 0$. Then for any Fibonacci sequence $\{f_i\}_{i=0}^n$, there exist two numbers A and B such that $f_i = A\phi^i + B\bar{\phi}^i$, where $i = 0, 1, \ldots, n$. It follows that for any Fibonacci signature F, there exist two numbers A and B such that $F = A(1, \phi)^{\otimes n} + B(1, \bar{\phi})^{\otimes n}$.

Let $T = \begin{bmatrix} n_0 & p_0 \\ n_1 & p_1 \end{bmatrix} \in \mathbf{GL}_2$ be a basis, then for any realizable recognizer signature R, we have

$$R = (A(1, \phi)^{\otimes n} + B(1, \bar{\phi})^{\otimes n}) T^{\otimes n}$$

$= A(n_0 + n_1\phi, p_0 + p_1\phi)^{\otimes n} + B(n_0 + n_1\bar{\phi}, p_0 + p_1\bar{\phi})^{\otimes n}.$

So R is also symmetric, and writing in symmetric notation $R = [x_0, x_1, \ldots, x_n]$, in which x_i is

$$A(n_0+n_1\phi)^{n-i}(p_0+p_1\phi)^i+B(n_0+n_1\bar{\phi})^{n-i}(p_0+p_1\bar{\phi})^i. \quad (1)$$

A matrix $T \in \mathbf{GL}_2$ defines a Möbius function $\ell_T(z) = \frac{p_1 z+p_0}{n_1 z+n_0}$, then $x_i = A'(\ell_T(\phi))^i + B'(\ell_T(\bar{\phi}))^i$, for some constants A' and B'.

When we replace T by $(T^{-1})^{\mathtt{T}}$, all results for recognizers work for generators. In particular, if $G = [x_0, x_1, \ldots, x_n]^{\mathtt{T}}$ is realizable as a Fibonacci gate on a basis T, then x_i is

$$A(p_1-p_0\phi)^{n-i}(-n_1+n_0\phi)^i+B(p_1-p_0\bar{\phi})^{n-i}(-n_1+n_0\bar{\phi})^i. \quad (2)$$

Theorem 4.1. *A symmetric signature $[x_0, x_1, \ldots, x_n]$ (for a generator or a recognizer) is realizable as a Fibonacci gate on some basis iff there exist three constants a, b and c, such that $b^2 - 4ac \neq 0$, and for all k, where $0 \leq k \leq n-2$,*

$$ax_k + bx_{k+1} + cx_{k+2} = 0. \quad (3)$$

Proof: Here we only prove it for recognizers; the case for generator is similar.
"$\Rightarrow$": From (1), we choose $a = (p_0 + p_1\phi)(p_0 + p_1\bar{\phi})$, $b = -(n_0 + n_1\phi)(p_0 + p_1\bar{\phi}) - (p_0 + p_1\phi)(n_0 + n_1\bar{\phi})$ and $c = (n_0 + n_1\phi)(n_0 + n_1\bar{\phi})$. Then $b^2 - 4ac \neq 0$ and we can verify that (3) is satisfied.
"$\Leftarrow$": If $c \neq 0$, then $\{x_i\}$ is a second-order homogeneous linear recurrence sequence. Since $b^2 - 4ac \neq 0$, $\{x_i\}$ has the form $x_i = A'\alpha^i + B'\beta^i$ for some $\alpha \neq \beta$. By the theory of Möbius transformations, there exists a $T \in \mathbf{GL}_2$ such that $\ell_T(\phi) = \alpha$ and $\ell_T(\bar{\phi}) = \beta$. More explicitly, in (1), we can choose $A = A'$, $B = B'$, $n_0 = 1$, $n_1 = 0$, $p_0 = (\beta\phi - \alpha\bar{\phi})/(\phi - \bar{\phi})$ and $p_1 = (\alpha - \beta)/(\phi - \bar{\phi})$. This implies that $\{x_i\}$ is realizable. The case $a \neq 0$ is similar. If $a = c = 0$, then $b \neq 0$. In this case all the $x_i = 0$ except x_0 and x_n. Then in (1), choosing $A = x_0/(\bar{\phi}-\phi)^n$, $B = x_n/(\phi-\bar{\phi})^n$, $n_0 = \bar{\phi}$, $n_1 = -1$, $p_0 = \phi$ and $p_1 = -1$, we can show that $\{x_i\}$ is realizable. ∎

Theorem 4.2. *A set of symmetric generators $G_1, G_2, \ldots, G_s$ and recognizers $R_1, R_2, \ldots, R_t$ are simultaneously realizable as Fibonacci gates on some basis iff there exist three constants a, b and c, such that $b^2 - 4ac \neq 0$ and the following two conditions are satisfied:*
1. For any recognizer $R_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}]$ and any $k = 0, 1, \ldots, n_i - 2$, $ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$.
2. For any generator $G_j = [y_1^{(j)}, y_2^{(j)}, \ldots, y_{m_j}^{(j)}]$ and any $k = 0, 1, \ldots, m_j - 2$, $cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$.

Proof: "$\Rightarrow$": Let $T = \begin{bmatrix} n_0 & p_0 \\ n_1 & p_1 \end{bmatrix}$ be a basis on which they are simultaneously realizable. Then all the recognizers $R_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}]$ have the form (1). For each R_i, we can choose the same a, b and c as in Theorem 4.1. Then for any $k = 0, 1, \ldots, n_i - 2$, $ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$.

For the generators, replace T by $(T^{-1})^{\mathtt{T}}$, we have the same result. If we define a', b' and c' according to $(T^{-1})^{\mathtt{T}}$, then we can verify that $a' = -c/\det^2(T)$, $b' = b/\det^2(T)$ and $c' = -a/\det^2(T)$. This uses properties of ϕ and $\bar{\phi}$, where ϕ is the golden ratio. It follows that $cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$.
"$\Leftarrow$": If $c \neq 0$, then each recognizer sequence is a second-order homogeneous linear recurrence sequence. Since $b^2 - 4ac \neq 0$, let α, β be the two distinct roots of $cX^2 + bX + a$. Each $\{x_k^{(i)}\}$ has the form $x_k^{(i)} = A_i\alpha^k + B_i\beta^k$. Then all the $R_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}]$ are realizable on $T = \begin{bmatrix} 1 & (\beta\phi - \alpha\bar{\phi})/(\phi - \bar{\phi}) \\ 0 & (\alpha - \beta)/(\phi - \bar{\phi}) \end{bmatrix}$ as in the above proof.

Since $cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$ and $c \neq 0$, each reversed generator sequence is a second-order homogeneous linear recurrence sequence. Then $-\alpha$ and $-\beta$ are the two roots of $cX^2 - bX + a$. As a result, we know that each generator $\{y_k^{(j)}\}$ has the form $y_k^{(j)} = A'_j(-\alpha)^{m_j-k} + B'_j(-\beta)^{m_j-k}$. Then it is easy to verify that they are also realizable on T as generators.

The case $a \neq 0$ is similar. Finally if $a = c = 0$, then $b \neq 0$. In this case all the sequences have the form $[*, 0, 0, \ldots, 0, *]$, and they are all realizable on $T = \begin{bmatrix} \bar{\phi} & \phi \\ -1 & -1 \end{bmatrix}$. ∎

5 Interpolation Method

Polynomial interpolation is a powerful tool in the study of counting problems initiated by Valiant [15] and further developed by Vadhan, Dyer and Greenhill [14, 10]. We discuss the interpolation method we will use in this paper.

Let $\Omega = (G, \mathcal{F})$ be a signature grid. Suppose $g \in \mathcal{F}$ is a symmetric signature with arity 2, and we denote it as $[x, y, z]$. Thus $g(00) = x$, $g(01) = g(10) = y$ and $g(11) = z$. Let V_g be the subset of vertices assigned g in Ω. Suppose $|V_g| = n$. Then the holant value can be expressed as

$$\mathrm{Holant}_\Omega = \sum_{i+j+k=n} c_{i,j,k} x^i y^j z^k, \quad (4)$$

where $c_{i,j,k}$ is the sum over all edge assignments σ, of products of evaluations at all $v \in V(G) - V_g$, where σ satisfies the property that the number of vertices in V_g having exactly 0 or 1 or 2 incident edges assigned 1 is i or j or k, respectively. If we can evaluate these $c_{i,j,k}$, we can evaluate Holant_Ω.

Now suppose $\{f_s\}$ is a sequence of symmetric functions of arity 2, with signatures $[x_s, y_s, z_s]$, for $s = 0, 1, \ldots$. If we replace each occurrence of g by f_s in Ω we get a new signature grid Ω_s with

$$\text{Holant}_{\Omega_s} = \sum_{i+j+k=n} c_{i,j,k} x_s^i y_s^j z_s^k. \quad (5)$$

Note that the same set of values $c_{i,j,k}$ occur. We can treat $c_{i,j,k}$ in (5) as a set of unknowns in a linear system. The idea of interpolation is to find a suitable sequence $\{f_s\}$ such that we can evaluate Holant_{Ω_s}, and then to find all $c_{i,j,k}$ by solving a linear system (5).

In this paper, the sequence $\{f_s\}$ will be constructed recursively using a suitable gadget. Let $\mathcal{F}' = \mathcal{F} - \{g\}$. A sequence of $\mathcal{F}'$-gates N_s will be constructed, such that its signature is f_s. Recursively from the construction, f_s will be symmetric. Let this signature be denoted by $[x_s, y_s, z_s]$, then the construction will yield a linear recurrence:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{s-1} \\ y_{s-1} \\ z_{s-1} \end{bmatrix}. \quad (6)$$

Let A denote the 3×3 matrix. This A will be independent of s. Suppose A has distinct eigenvalues α, β and γ, and $A = T^{-1}\text{diag}(\alpha, \beta, \gamma)T$, where the rows of T are the row eigenvectors of A.

Let $(u, v, w)^{\text{T}} = T(x_0, y_0, z_0)^{\text{T}}$ be the inner products of the row eigenvectors with the initial values. Then

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = T^{-1} \begin{bmatrix} \alpha^s & 0 & 0 \\ 0 & \beta^s & 0 \\ 0 & 0 & \gamma^s \end{bmatrix} T \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = T^{-1} \begin{bmatrix} u\alpha^s \\ v\beta^s \\ w\gamma^s \end{bmatrix}$$

$$= T^{-1} \begin{bmatrix} u & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & w \end{bmatrix} \begin{bmatrix} \alpha^s \\ \beta^s \\ \gamma^s \end{bmatrix}.$$

Let $B = T^{-1}\text{diag}(u, v, w)$. B is non-singular iff $uvw \neq 0$, which we will assume in the following. It follows that

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}^{\otimes n} = B^{\otimes n} \begin{bmatrix} \alpha^s \\ \beta^s \\ \gamma^s \end{bmatrix}^{\otimes n}. \quad (7)$$

The rows and columns of $B^{\otimes n}$ are indexed by $t_1 t_2 \cdots t_n \in \{1, 2, 3\}^n$. There are 3^n equalities in (7). Let $\kappa = \{1^i 2^j 3^k \mid i + j + k = n\}$ be the set of "types" for all $t_1 t_2 \cdots t_n$, and $|\kappa| = \binom{n+2}{2}$. Define an equivalence relation on the indices, $t_1 t_2 \cdots t_n \sim t'_1 t'_2 \cdots t'_n$ if they have the same numbers of 1's and 2's and 3's. We identify the equivalence classes with κ.

Define $\widehat{B^{\otimes n}}$ to be the $3^n \times \binom{n+2}{2}$ matrix obtained from $B^{\otimes n}$ by adding all columns in each equivalence class. We claim that this matrix $\widehat{B^{\otimes n}}$ has full column rank $\binom{n+2}{2}$. This is easy to see, since any non-trivial linear combination of the columns of $\widehat{B^{\otimes n}}$ can be also obtained as a non-trivial linear combination of the columns of $B^{\otimes n}$, which is non-singular. The crucial point is that $\widehat{B^{\otimes n}}$ is obtained from $B^{\otimes n}$ by adding all columns within each class in a *disjoint partition* of columns.

Next we claim that there are exactly $\binom{n+2}{2}$ distinct rows in $\widehat{B^{\otimes n}}$, and if we select a set of $\binom{n+2}{2}$ distinct representatives to form a new $\binom{n+2}{2} \times \binom{n+2}{2}$ matrix $\widetilde{B^{\otimes n}}$, it is of full rank. We only need to prove that if $t_1 t_2 \cdots t_n \sim t'_1 t'_2 \cdots t'_n$ then the two rows of $\widehat{B^{\otimes n}}$ indexed by $t_1 t_2 \cdots t_n$ and $t'_1 t'_2 \cdots t'_n$ are the same. Since $t_1 t_2 \cdots t_n \sim t'_1 t'_2 \cdots t'_n$, there is a permutation σ such that σ maps $t_1 t_2 \cdots t_n$ to $t'_1 t'_2 \cdots t'_n = t_{\sigma(1)} t_{\sigma(2)} \cdots t_{\sigma(n)}$. If we perform a simultaneous permutation of rows and columns of $B^{\otimes n}$ by σ, the entries $(B^{\otimes n})_{t_1 t_2 \cdots t_n, c_1 c_2 \cdots c_n} = B_{t_1,c_1} B_{t_2,c_2} \cdots B_{t_n,c_n}$ is mapped to $(B^{\otimes n})_{t_{\sigma(1)} t_{\sigma(2)} \cdots t_{\sigma(n)}, c_{\sigma(1)} c_{\sigma(2)} \cdots c_{\sigma(n)}} = B_{t_{\sigma(1)},c_{\sigma(1)}} B_{t_{\sigma(2)},c_{\sigma(2)}} \cdots B_{t_{\sigma(n)},c_{\sigma(n)}}$. By permuting the factors, it is $B_{t_1,c_1} B_{t_2,c_2} \cdots B_{t_n,c_n}$. That is, a simultaneous permutation of rows and columns of $B^{\otimes n}$ by σ leaves it invariant. But the permutation of the columns by σ certainly induces a permutation within each equivalence class of κ, and thus keeps its sum invariant. It follows that the two rows of $\widehat{B^{\otimes n}}$ indexed by $t_1 t_2 \cdots t_n$ and $t'_1 t'_2 \cdots t'_n$ are the same. Since $\widehat{B^{\otimes n}}$ has full column rank $\binom{n+2}{2}$, $\widetilde{B^{\otimes n}}$ also has full rank $\binom{n+2}{2}$ (and exactly $\binom{n+2}{2}$ distinct rows).

Now we return to the linear system (5), for $0 \leq s < \binom{n+2}{2}$. If we consider this as a linear equation system with unknowns $c_{i,j,k}$, indexed by κ, it has a coefficient matrix which is the product of $\widetilde{B^{\otimes n}}$ with a Vandermonde matrix $\mathbf{V}$. The rows of $\mathbf{V}$ are indexed by κ and columns are indexed by $0 \leq s < \binom{n+2}{2}$. The entry of $\mathbf{V}$ at $(1^i 2^j 3^k, s)$ is $(\alpha^i \beta^j \gamma^k)^s$. This Vandermonde matrix will be of full rank if all entries $\alpha^i \beta^j \gamma^k$ are distinct.

We summarize this as follows:

Theorem 5.1. *Suppose the recurrence matrix A of the construction N_s satisfies*

1. $\det(A) \neq 0$*,*
2. *The initial signature $[x_0, y_0, z_0]$ is not orthogonal to any row eigenvector of A, and*
3. *For all $(i, j, k) \in \mathbf{Z}^3 - \{(0, 0, 0)\}$ with $i + j + k = 0$, $\alpha^i \beta^j \gamma^k \neq 1$.*

Then all $c_{i,j,k}$ in (4), where $1^i 2^j 3^k \in \kappa$, can be computed in polynomial time.

6 Interpolatability Implies Hardness

A signature $[x_0, x_1, \ldots, x_n]$ is called degenerate iff it is of the form $[s^0 t^n, s^1 t^{n-1}, \ldots, s^n t^0]$, for some s and t.

Lemma 6.1. *For any non-degenerate signature* $[y_0, y_1, y_2, y_3]$, *there exists a symmetric signature* $[x_0, x_1, x_2]$ *of arity two, such that* $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ *is #P-Complete. Furthermore this remains true even for planar graphs.*

Proof: Our starting point is that $\#[0, 1, 1]|[1, 0, 0, 1]$ and $\#[1, 0, 1]|[1, 1, 0, 0]$ are both #P-Complete. 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 [19]. We remark that both of them remain #P-Complete even for planar graphs.

Our technique here is to use the theory of holographic reductions. Given a non-degenerate signature $[y_0, y_1, y_2, y_3]$, we can give a parameterization in terms of a homogeneous 2nd order recurrence relation. There are three cases: (1) $y_i = \alpha_1^{3-i}\alpha_2^i + \beta_1^{3-i}\beta_2^i$, where $\alpha_1\beta_2 - \alpha_2\beta_1 \neq 0$; (2) $y_i = Ai\alpha^{i-1} + B\alpha^i$, where $A \neq 0$; or (3) $y_i = A(3-i)\alpha^{2-i} + B\alpha^{3-i}$, where $A \neq 0$. The last case can be viewed as the reversal of the second case, so we will omit the proof for this case. Note that for any non-degenerate signature one of these parameterizations is always possible. (In the expression $i\alpha^{i-1}$, if $\alpha = 0$, we take the convention that $i\alpha^{i-1} = 0, 1, 0, 0$ for $i = 0, 1, 2, 3$ respectively.)

For the first case, under the basis $T = \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix}$, the signature $[1, 0, 0, 1]$ becomes $[y_0, y_1, y_2, y_3]$. This is the result of the contravariant transformation $(y_0, y_1, y_1, y_2, y_1, y_2, y_2, y_3)^{\mathrm{T}} = T^{\otimes 3}(1, 0, 0, 0, 0, 0, 0, 1)^{\mathrm{T}}$. Under the same basis, $[0, 1, 1]$ undergoes the covariant transformation $(x_0, x_1, x_1, x_2) = (0, 1, 1, 1)(T^{-1})^{\otimes 2}$, to become a new symmetric signature $[x_0, x_1, x_2]$. So by the holographic reduction the complexity of $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ and $\#[0, 1, 1]|[1, 0, 0, 1]$ is the same. Since $\#[0, 1, 1]|[1, 0, 0, 1]$ is #P-Complete, we know that $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ is also #P-Complete.

For the second case, we choose the following basis $T = \begin{bmatrix} 1 & \frac{B-1}{3} \\ \alpha & A + \frac{B-1}{3}\alpha \end{bmatrix}$. Then under the contravariant transformation $(y_0, y_1, y_1, y_2, y_1, y_2, y_2, y_3)^{\mathrm{T}} = T^{\otimes 3}(1, 1, 1, 0, 1, 0, 0, 0)^{\mathrm{T}}$, the signature $[1, 1, 0, 0]$ becomes $[y_0, y_1, y_2, y_3]$. Under the same basis, $[1, 0, 1]$ undergoes the covariant transformation $(x_0, x_1, x_1, x_2) = (1, 0, 0, 1)(T^{-1})^{\otimes 2}$, to become a new symmetric signature $[x_0, x_1, x_2]$. (We chose these basis transformations not "out of blue", but rather they are informed by an underlying signature theory of holographic algorithms [3, 4]. But for brevity of exposition we state these transformations *as is* without discussing the background. They can be directly verified, albeit a bit tedious.)

It follows from holographic reductions the complexity of $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ and $\#[1, 0, 1]|[1, 1, 0, 0]$ is the same. Since $\#[1, 0, 1]|[1, 1, 0, 0]$ is #P-Complete, $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ is also #P-Complete. □

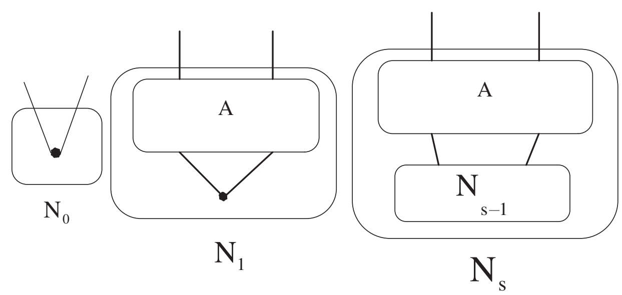
This lemma directly gives the following theorem:

Theorem 6.1. *If* $[y_0, y_1, y_2, y_3]$ *is non-degenerate, and if* $[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ *can be used to interpolate all symmetric signatures of arity 2, then* $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ *is #P-Complete.*

This theorem gives a sufficient condition for $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$ to be hard. In the next section, we will state an algebraic lemma that guarantees this interpolatability, and then in Section 8 we use this theorem to prove all the hardness results for Boolean symmetric signatures.

7 An Algebraic Lemma

Fix a signature set $\mathcal{F}$. Our general recursive construction of a series of gadgets is depicted in the following figure.



Every gadget N_s will have arity 2. (In this paper we restrict to interpolations for signatures of arity 2. But the general theory can be applied to arbitrary arity.) The first gadget is just a vertex with some signature in $\mathcal{F}$. The key of this construction is the $\mathcal{F}$-gate $\mathcal{A}$ in the figure with arity 4. The specific $\mathcal{A}$'s we will use are depicted in the next section. In each step, we will connect a copy of $\mathcal{A}$ to make a new gadget. In order to make use of Theorem 5.1, we choose our $\mathcal{F}$-gate $\mathcal{A}$ such that all the signatures are symmetric. We denote by $[x_s, y_s, z_s]$ the signature of the s-th gadget. Then there is a linear recursive relation in the constructed gadgets, that is, $(x_s, y_s, z_s)^{\mathrm{T}} = A(x_{s-1}, y_{s-1}, z_{s-1})^{\mathrm{T}}$ for some matrix A as in (6). We can use the same A because the matrix is completely determined by the $\mathcal{F}$-gate $\mathcal{A}$.

According to Theorem 5.1, the interpolatability of the signature requires three conditions, of which the main condition is: For no $i, j, k \in \mathbf{Z}$ with $i + j + k = 0$, other than the trivial $(0, 0, 0)$, do we have

$$\alpha^i \beta^j \gamma^k = 1. \tag{8}$$

This condition ensures that a Vandermonde matrix is nonsingular. Let $f(x)$ be the characteristic polynomial of A.

The following algebraic lemma gives a sufficient condition that condition (8) is satisfied. The proof of this lemma uses some basic Galois theory. Due to space limitation, the proof is omitted here and will be presented in the full paper.

Lemma 7.1. *Let $f(x) = x^3 + c_2x^2 + c_1x + c_0 \in \mathbf{Q}[x]$ be a given polynomial with rational coefficients. It is decidable in polynomial time whether any non-trivial solution to (8) exists, where α, β and γ are its roots, and if so, find all solutions (in terms of a short basis of the lattice). If f is irreducible, except of the form $x^3 + c$ for some $c \in \mathbf{Q}$, there are no non-trivial solutions to (8).*

8 Boolean Symmetric Signatures

In this section, we give a dichotomy theorem for all counting problems of the form $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$, where each $x_i, y_j \in \{0, 1\}$. Such signatures are called Boolean symmetric signatures [3]. This family of signatures is particularly important because they have clear combinatorial meanings and many combinatorial constraints can be described by these signatures.

By flipping all 0's and 1's, we see that the problem $\#[x_2, x_1, x_0]|[y_3, y_2, y_1, y_0]$ always has the same complexity as the problem $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$. So we will only consider one problem for each pair. In the following we only enumerate problems $\#[x_0, x_1, x_2]|[y_0, y_1, y_2, y_3]$, where we let (1) $x_0 \geq x_2$, and (2) if $x_0 = x_2$, then $y_0 \geq y_3$, and (3) if $x_0 = x_2$ and $y_0 = y_3$, then $y_1 \geq y_2$. Also when we consider a signature $[y_0, y_1, y_2, y_3]$ we also consider its reversal, in particular in terms of expressibility as a second order recurrence relation involving its eigenvalues. We also will only implicitly verify the other conditions in Theorem 5.1, and not mention it explicitly, i.e., we will only focus explicitly on the condition (8).

8.1 The Tractable Cases

First if at least one side of the signatures is degenerate, then the holant Holant_Ω can be computed in polynomial time. The degenerate Boolean signatures of arity 2 are: $[0, 0, 0], [0, 0, 1], [1, 0, 0], [1, 1, 1]$; and the degenerate Boolean signatures of arity 3 are: $[0, 0, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [1, 1, 1, 1]$. These problems are all trivially solvable; e.g., for $\#[x_0, x_1, x_2]|[1, 1, 1, 1]$, the holant is completely decomposed as a product over identical disjoint paths of length 2, i.e., $\prod_{v \in V:\deg(v)=3}(x_0 + 2x_1 + x_2)$. From now on, we discuss non-degenerate Boolean signatures and rule out these 8 signatures.

Some holants evaluate to 0 by a cardinality argument. For example, in the counting problem $\#[0, 1, 0]|[0, 1, 0, 0]$, signature $[0, 1, 0]$ requires that exactly half of all edges have value 1, while the signature $[0, 1, 0, 0]$ requires that exactly one third of edges have value 1. This is a contradiction. So there are no feasible solutions and the output of the counting problem is 0. These infeasible cases include the following problems: $\#[0, 1, 0]|[1, 1, 0, 0]$, $\#[0, 1, 0]|[0, 1, 0, 0]$, $\#[1, 1, 0]|[0, 0, 1, 1]$, $\#[1, 1, 0]|[0, 0, 1, 0]$.

Similarly, the following two problems are both tractable: $\#[1, 0, 1]|[1, 0, 0, 1]$ and $\#[0, 1, 0]|[1, 0, 0, 1]$, proved by an easy connectivity argument.

The remaining tractable cases are those which can be solved by holographic algorithms with Fibonacci gates. They are $\#[0, 1, 0]|[1, 0, 1, 0]$,$\#[1, 0, 1]|[1, 0, 1, 0]$, $\#[1, 0, 1]|[1, 1, 0, 1]$ and $\#[1, 1, 0]|[1, 1, 0, 1]$. This fact can be easily verified using our characterization Theorem 4.2.

8.2 Tractable for Planar Graphs but Hard in General

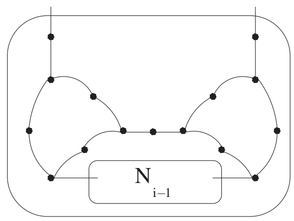
This class contains 3 members: $\#[1, 0, 1]|[0, 1, 0, 0]$, $\#[1, 0, 1]|[0, 1, 1, 0]$ and $\#[0, 1, 0]|[0, 1, 1, 0]$. The problem $\#[1, 0, 1]|[0, 1, 0, 0]$ is counting perfect matchings in a 3-regular graph (Problem PM). The second one, $\#[1, 0, 1]|[0, 1, 1, 0]$ can be viewed as a special edge coloring problem. An edge coloring with 2 colors on a 3-regular graph is called valid if at each vertex the incident edges are not monochromatic, and the problem $\#[1, 0, 1]|[0, 1, 1, 0]$ is counting all the valid colorings for a given graph. The third problem $\#[0, 1, 0]|[0, 1, 1, 0]$ is an Ising problem studied by Valiant in [17] (Problem ICE). For planar graphs, all these three problems are polynomial time computable by holographic algorithms with matchgates.

The #P-hardness for Problem PM for general 3-regular graphs is proved in [7]. What is the complexity of $\#[1, 0, 1]|[0, 1, 1, 0]$ (Problem COLOR) and $\#[0, 1, 0]|[0, 1, 1, 0]$ (Problem ICE) for general 3-regular graphs? We remark that we can not directly use Theorem 6.1 to show their hardness. The reason is that all symmetric signatures of arity 2 realizable or interpolatable are of the form $[a, b, a]$. However using a similar interpolation technique, we can show that all signatures of form $[a, b, a]$ and of form $[a, b, b, a]$ can be interpolated by both problems. In particular, we can realize $[0, 1, 0]$ and $[1, 0, 0, 1]$, with which we can realize all signatures in #NAE-3SAT(see [17]). The fact that #NAE-3SAT is #P-Complete implies that the two problems we considered here are also #P-Complete. All proof details are omitted here.

8.3 The Hard Cases (Hard even for Planar Graphs)

In this section, we make use of the tools we developed in Sections 6 and Section 7 to prove hardness for all the remaining problems.

Here we go over all cases of the form $\#[0,1,0]|[y_0, y_1, y_2, y_3]$ (note that there are two cases for each listed case by symmetry). The first hard case is $\#[0,1,0]|[1,1,1,0]$. We will consider instead its flipped case $\#[0,1,0]|[0,1,1,1]$. Over planar graphs (we are assuming planarity in this subsection) this is called #Pl-Rtw-Opp-3CNF—Satisfiability of planar 3CNF formulae where each variable occurs twice and in opposite signs. We note that #Pl-Rtw-Mon-3CNF is #P-complete and ⊕Pl-Rtw-Mon-3CNF is ⊕P-complete, while $\#_7$Pl-Rtw-Mon-3CNF is P-time computable [18]. Here we use Theorem 6.1 to prove that #Pl-Rtw-Opp-3CNF is also #P-complete.



N_i

We use the above gadget to construct recursively an arity 2 gate N_i using the signatures $[0,1,0]|[0,1,1,1]$. This means that in the construction, every node of degree two (resp. three) is assigned a signature $[0,1,0]$ (resp. $[0,1,1,1]$).

Obviously, the signatures $[a_i, b_i, c_i]$ for N_i are all symmetric. It takes some computation, but it can be verified that the following recursive relation holds:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} 48 & 136 & 96 \\ 28 & 88 & 68 \\ 16 & 56 & 48 \end{bmatrix} \begin{bmatrix} a_{i-1} \\ b_{i-1} \\ c_{i-1} \end{bmatrix}.$$

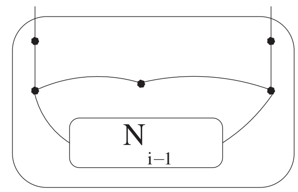
The characteristic polynomial is $x^3 - 184x^2 + 1600x - 512$. It is easy to verify that it is irreducible over $\mathbf{Q}[x]$. Then by Lemma 7.1, we know that this family of gadgets can be used for interpolation. As a result, $\#[0,1,0]|[0,1,1,1]$ is #P-complete.

The next hard case is $\#[0,1,0]|[1,1,0,1]$. This is called #Pl-Rtw-Opp-$F_{0,1,3}$-SAT in the notation of [19]. In [19], they proved that #Pl-Rtw-Mon-$F_{0,1,3}$-SAT is P-time computable and if one does not restrict the occurrence of the variables, then #Pl-Rtw-$F_{0,1,3}$-SAT is #P-complete. Here we improve this result by showing that #Pl-Rtw-Opp-$F_{0,1,3}$-SAT remains #P-complete.

If we use the same gadget as above, we have the following recursive relation:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} 8 & 8 & 0 \\ 8 & 12 & 4 \\ 8 & 16 & 8 \end{bmatrix} \begin{bmatrix} a_{i-1} \\ b_{i-1} \\ c_{i-1} \end{bmatrix}.$$

Unfortunately this matrix is singular and therefore we cannot use this recursive construction to do interpolation.



N_i

However we can use another gadget as above. Here again each vertex of degree 2 (resp. 3) is assigned a signature $[0,1,0]$ (resp. $[1,1,0,1]$).

Then we have a recursive relation:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 0 \end{bmatrix} \begin{bmatrix} a_{i-1} \\ b_{i-1} \\ c_{i-1} \end{bmatrix}.$$

The characteristic polynomial is $x^3 - x^2 - 4x - 4$. It is easy to verify that it is irreducible over $\mathbf{Q}[x]$, and by Lemma 7.1, we know that this family of gadgets can be used for interpolation. As a result, $\#[0,1,0]|[1,1,0,1]$ and $\#[0,1,0]|[1,0,1,1]$ are #P-complete.

We summarize our treatment of problems of the form $\#[0,1,0]|[y_0, y_1, y_2, y_3]$: The cases where $[y_0, y_1, y_2, y_3] = [0,0,0,0]$, $[0,0,0,1]$, $[1,0,0,0]$, $[1,1,1,1]$ are trivial signatures. The pair $[0,0,1,0]$, $[0,1,0,0]$ and the pair $[0,0,1,1]$, $[1,1,0,0]$ are both trivial by a counting argument. The pair $[0,1,0,1]$ and $[1,0,1,0]$ are solvable in P by Fibonacci gates. The Problem ICE $\#[0,1,0]|[0,1,1,0]$ is solvable in P for planar graphs, but #P-complete for general graphs. The pair where $[y_0, y_1, y_2, y_3] = [0,1,1,1]$ and $[1,1,1,0]$ are #P-complete, dealt with as #Pl-Rtw-Opp-3CNF. The case $[1,0,0,1]$ is trivial by a connectivity argument. Finally the pair $[1,0,1,1]$ and $[1,1,0,1]$ are #P-complete, dealt with as #Pl-Rtw-Opp-$F_{0,1,3}$-SAT. This completes all 16 cases of $\#[0,1,0]|[y_0, y_1, y_2, y_3]$.

All hard cases of the form $\#[1,0,1]|[y_0, y_1, y_2, y_3]$ have been proved in [19] using a different proof. We can reprove them in our framework to give a uniform treatment, but we omit the details here. We omit proofs for problems of the form $\#[1,1,0]|[y_0, y_1, y_2, y_3]$.

To recap for the side $[x_0, x_1, x_2]$ of arity 2, the cases $[0,0,0]$, $[0,0,1]$, $[1,0,0]$ and $[1,1,1]$ are trivial. The case $[0,1,0]$ is discussed above in detail. The proof for the pair $[0,1,1]$ and $[1,1,0]$ is similar and omitted here. The case $[1,0,1]$ has been done in [19].

To sum up, we have the following table (we removed entries for degenerate signatures). In the table "T" means that it is computable in P-time by some trivial reasons; "F" means that it is computable in P-time by holographic algorithms with Fibonacci gates; "P" means that it is computable in P-time for planar graphs (by holographic algorithms with matchgates) but #P-complete for general graphs; and "H" means that it is #P-complete even for planar graphs.

$f_2 \mid g_3$	$[0, 1, 0]$	$[1, 0, 1]$	$[1, 1, 0]$
$[0, 0, 1, 0]$	T	P	T
$[0, 0, 1, 1]$	T	H	T
$[0, 1, 0, 0]$	T	P	H
$[0, 1, 0, 1]$	F	F	H
$[0, 1, 1, 0]$	P	P	H
$[0, 1, 1, 1]$	H	H	H
$[1, 0, 0, 1]$	T	T	H
$[1, 0, 1, 0]$	F	F	H
$[1, 0, 1, 1]$	H	F	H
$[1, 1, 0, 0]$	T	H	H
$[1, 1, 0, 1]$	H	F	F
$[1, 1, 1, 0]$	H	H	H

Theorem 8.1. *Every counting problem $\#[x_0, x_1, x_2] | [y_0, y_1, y_2, y_3]$, where $[x_0, x_1, x_2]$ and $[y_0, y_1, y_2, y_3]$ are Boolean signatures, is either (a) in P; or (b) #P-complete but solvable in P for planar graphs; or (c) #P-complete even for planar graphs. The results are summarized in the table (with some trivial cases removed.)*

Acknowledgments

We sincerely thank Les Valiant for his comments and ideas regarding this work, and regarding holographic algorithms in general. We benefited greatly from discussions with him. We are also very grateful to Martin Dyer and Angsheng Li. Discussions with Martin and Angsheng helped a great deal in developing some of the ideas reported here. We gratefully acknowledge helpful comments by Xi Chen, Madhu Sudan, Endre Szemeredi and Avi Wigderson.

References

[1] A. A. Bulatov, The Complexity of the Counting Constraint Satisfaction Problem. In the proceedings of ICALP (1) 2008, pp 646-661.

[2] A. A. Bulatov, M. Grohe: The complexity of partition functions. Theor. Comput. Sci. 348(2-3): 148-186 (2005).

[3] J-Y. Cai and P. Lu. On Symmetric Signatures in Holographic Algorithms. In the proceedings of STACS 2007, LNCS Vol 4393, pp 429–440.

[4] J-Y. Cai and P. Lu. Holographic Algorithms: From Art to Science. In the proceedings of STOC 2007, pp 401-410.

[5] N. Creignou, M. Hermann: Complexity of Generalized Satisfiability Counting Problems. Inf. Comput. 125(1): 1-12 (1996).

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

[7] P. Dagum and M. Luby. Approximating the permanent of graphs with large factors. Theoretical Computer Science, 102:283-305 (1992).

[8] M. E. Dyer, L. A. Goldberg, M. Jerrum: The Complexity of Weighted Boolean #CSP CoRR abs/0704.3683: (2007).

[9] M. E. Dyer, L. A. Goldberg, M. Paterson: On counting homomorphisms to directed acyclic graphs. J. ACM 54(6): (2007).

[10] M. E. Dyer, C. S. Greenhill: The complexity of counting graph homomorphisms. Random Struct. Algorithms 17(3-4): 260-289 (2000).

[11] L. A. Goldberg, M. Grohe, M. Jerrum and M. Thurley: A complexity dichotomy for partition functions with mixed signs. CoRR abs/0804.1932: (2008).

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

[13] P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27: 1209-1225 (1961).

[14] S. P. Vadhan: The Complexity of Counting in Sparse, Regular, and Planar Graphs. SIAM J. Comput. 31(2): 398-427 (2001).

[15] L. G. Valiant: The Complexity of Computing the Permanent. Theor. Comput. Sci. 8: 189-201 (1979).

[16] L. G. Valiant: Quantum Circuits That Can Be Simulated Classically in Polynomial Time. SIAM J. Comput. 31(4): 1229-1254 (2002).

[17] L. G. Valiant. Holographic Algorithms (Extended Abstract). In *Proc. 45th IEEE Symposium on Foundations of Computer Science*, 2004, 306–315.

[18] L. G. Valiant. Accidental Algorithms. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science* 2006, 509–517.

[19] M. Xia, P. Zhang and W. Zhao: Computational complexity of counting problems on 3-regular planar graphs. Theor. Comput. Sci. 384(1): 111-125 (2007).