A New Class of Holographic Algorithms: Fibonacci Gates

Abstract

We introduce Fibonacci gates as a polynomial time computable primitive, and develop a theory of holographic algorithms based on these gates. The Fibonacci gates play the role of matchgates in Valiant's theory [16]. We develop a signature theory and characterize all realizable signatures for Fibonacci gates. For bases of arbitrary dimensions we prove a universal bases collapse theorem. We apply this theory to give new polynomial time algorithms for certain counting problems. We also use this framework to prove that some slight variations of these counting problems are #P-hard.

1 Introduction

L. Valiant [15, 16] introduced a marvelously original algorithmic design technique, called matchgate computations and holographic algorithms. There are two main ingredients to this new theory. The first is to use perfect matchings of planar graph fragments to encode and process information. The second is to create exponential sums of these perfect matchings in a "holographic mix", and achieve exponential cancelations in the process. They lead ultimately to polynomial time algorithms by applying the Fisher-Kasteleyn-Temperley (FKT) algorithm [9, 10, 13] for planar perfect matchings.

This methodology has produced some quite exotic polynomial time algorithms, called *holographic algorithms*. The problems range from certain constrained satisfiability, to vertex cover, to other graph problems such as edge orientation and node/edge deletion. These problems were not known previously to be in P, and some minor variations are known to be NP-hard. For example, let #Pl-Rtw-Mon-3CNF denote the problem of counting the number of satisfying assignments of a planar read-twice monotone 3CNF formula. This problem is #P-complete; the problem mod 2, $\#_2$Pl-Rtw-Mon-3CNF, is known to be NP-hard. Using holographic algorithms Valiant [17] showed that $\#_7$Pl-Rtw-Mon-3CNF (counting modulo 7) is in P.

We will illustrate with an example the idea of creating a "holographic mix" by values of perfect matchings. Suppose we wish to represent the Boolean OR function of 3 inputs. In the framework of holographic algorithms this means that we want the "signature" $(0,1,1,1,1,1,1,1)$ indexed by three bits $b_1b_2b_3 \in \{0,1\}^3$. While this signature is not directly expressible by perfect matchings, it *is* possible to express it indirectly as a linear "superposition" using perfect matchings. Figure 1 shows a finite planar weighted graph (called a matchgate) Γ with 3 external nodes v_1, v_2, v_3. This matchgate has a "signature" $(\Gamma_{b_1b_2b_3}) = \frac{1}{4}(0,1,1,0,1,0,0,1)_{b_1b_2b_3\in\{0,1\}^3}$. This means that, the "perfect matching polynomial" $\sum_M \prod_{(i,j)\in M} w(i,j) = \Gamma_{b_1b_2b_3} = 0$ or $1/4$, for $b_1b_2b_3 \in \{0,1\}^3$, where the sum is over all perfect matchings M of Γ after the removal of $\{v_i | b_i = 1\}$. (Note that if all edge weights $w(i,j) = 1$ in this sum, then the perfect matching polynomial counts the number of perfect matchings.) Now we can choose two linearly independent basis vectors $\boldsymbol{\beta} = \left[\begin{pmatrix}1+\omega\\1-\omega\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right]$, where $\omega = e^{2\pi i/3}$. In the tensor product space spanned by $\boldsymbol{\beta}^{\otimes 3}$, we can represent the OR signature $(0,1,1,1,1,1,1,1)$ by perfect matchings as follows: We have

$$\left(\begin{bmatrix}1+\omega & 1\\1-\omega & 1\end{bmatrix}^{-1}\right)^{\otimes 3} = \frac{1}{8}\begin{pmatrix}
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\\
-1+\omega & 1+\omega & 1-\omega & -1-\omega & 1-\omega & -1-\omega & -1+\omega & 1+\omega\\
-1+\omega & 1-\omega & 1+\omega & -1-\omega & 1-\omega & -1+\omega & -1-\omega & 1+\omega\\
-3\omega & -2-\omega & -2-\omega & \omega & 3\omega & 2+\omega & 2+\omega & -\omega\\
-1+\omega & 1-\omega & 1-\omega & -1+\omega & 1+\omega & -1-\omega & -1-\omega & 1+\omega\\
-3\omega & -2-\omega & 3\omega & 2+\omega & -2-\omega & \omega & 2+\omega & -\omega\\
-3\omega & 3\omega & -2-\omega & 2+\omega & -2-\omega & 2+\omega & \omega & -\omega\\
3+6\omega & 3 & 3 & -1-2\omega & 3 & -1-2\omega & -1-2\omega & -1
\end{pmatrix},$$

and therefore (adding up the last 7 rows)

$$(0,1,1,1,1,1,1,1)\left(\begin{bmatrix}1+\omega & 1\\1-\omega & 1\end{bmatrix}^{-1}\right)^{\otimes 3} = \frac{1}{4}(0,1,1,0,1,0,0,1) = (\Gamma_{b_1b_2b_3}).$$

It follows that $(0,1,1,1,1,1,1,1) = (\Gamma_{b_1b_2b_3})\boldsymbol{\beta}^{\otimes 3}$. In this way each logical value 0 or 1 in $(0,1,1,1,1,1,1,1)$ is expressed as a "superposition" or a "holographic mix" of pefect matching values in $(\Gamma_{b_1b_2b_3})$.

In this paper we go beyond matchgates, and extend the reach of holographic algorithms. In Section 5 we will show how to solve in polynomial time a certain coloring problem. Consider a 3-regular graph. A 2-coloring (Black/White) of the edges is called *valid* if at each vertex the incident edges are not monochromatic. It is an *even* (resp. *odd*) coloring if it is a valid 2-coloring with an even (resp. odd) number of Black edges. The total number of valid 2-colorings is the sum of even and odd colorings. For planar graphs this total number can be computed in P by holographic algorithms based on matchgates. This is non-trivial, involving a similar "superposition" of pefect matchings as above. For general graphs, we will show in Section 7 that this problem is #P-complete. However, instead of taking a sum, to compute the difference of the numbers of even and odd colorings, even for planar graphs, the required signatures based on matchgates provably *do not* exist. This fact not withstanding, we solve this counting problem for the difference, even for general (non-planar) graphs, by holographic algorithms based on a new primitive: Fibonacci gates.

Our main contribution in this paper is to extend the framework of holographic algorithms by introducing Fibonacci gates. They play the role of matchgates in Valiant's theory, and constitute another class of P-time computable primitives. We develop a corresponding theory of holographic algorithms based on these Fibonacci gates. To understand what these Fibonacci gates can do in a computational setting we must understand its signature theory. We derive a complete characterization of all realizable signatures of Fibonacci gates under any choice of bases in $\mathbf{GL}_2$. We then apply this theory to some non-trivial counting problems to obtain P-time algorithms. We note that the signatures used in these holographic algorithms are not realizable by matchgates under any basis transformation, and therefore we have properly extended the reach of holographic algorithms.

Next we consider the signature theory under more general bases. Here we prove a general bases collapse theorem, showing that, for holographic algorithms based on Fibonacci gates, any holographic algorithm using a two-vector basis in arbitrary dimension can be simulated by a two-vector basis in dimension 2. These results parallel those obtained for matchgate signatures [3, 4].

Finally we prove some hardness results. We prove a minor variation of the problem we solved in polynomial time using holographic algorithms based on Fibonacci gates is #P-hard. Our framework of counting problems can discuss signatures which (unlike matchgates and Fibonacci gates) do not necessarily correspond to P-time computable primitives. We use a technique called interpolation to prove this hardness. In particular we make use of a technical lemma due to Vadhan [14]. An interesting feature of these reductions is that they provide some natural examples where the Karp-type mapping (or many-one) reductions seem to be weaker than the oracle query Cook-type reductions.

This paper is organized as follows: In Section 2 we define the framework of counting problems based on signatures. In Section 3 we introduce Fibonacci gates, and prove that they form a P-time computable primitive. In Section 4 we develop the signature theory for these Fibonacci gates. In Section 5, we consider some counting problems solvable in polynomial time using the theory just developed. In Section 6 we further develop the signature theory and prove a general bases collapse theorem. In Section 7 we prove hardness results using interpolation reductions.

2 Counting Problems and Holographic Reductions

Many counting problems can be formulated in the following framework.

A *signature grid* $\Omega = (H, \mathcal{F})$ is a tuple, where $H = (V, E)$ is a graph, and $\mathcal{F}$ are functions assigned to vertices of H. We consider every edge $e \in E$ as a Boolean variable, and every vertex $v \in V$ is assigned a function F_v from $\mathcal{F}$, where $F_v : \{0,1\}^{d(v)} \to \mathbf{F}$, for some field $\mathbf{F}$ and $d(v)$ is the degree of v. Given an assignment σ of all the edges, we have a valuation at each vertex v, which is the value of F_v on σ restricted to the edges incident to v. The value of Ω (or we simply say the value of H) under σ is the product of F_v on σ over all $v \in V$, and the value of Ω is the sum over all assignments σ. The counting problem on Ω is to compute this value.

Formally let $E = \{1, 2, \ldots, m\}$, the edges connected to vertex v are denoted by $i_1^v, i_2^v, \ldots, i_{d(v)}^v$. Then we define

$$\text{Holant}_\Omega = \sum_{x_1 x_2 \cdots x_m \in \{0,1\}^m} \prod_{v \in V} F_v(x_{i_1^v}, x_{i_2^v}, \ldots, x_{i_{d(v)}^v}).$$

We also write Holant_H when there is no confusion. 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 $(\mathbf{F}^2)^{\otimes d(v)}$. This is called a *signature*.

Many important counting problems can be viewed as computing Holant_Ω for appropriate signatures at each vertex. Consider, e.g., an assignment σ to edges incident to v, and let $F_v(\sigma) = 1$ if $\text{wt}(\sigma) = 1$, and $F_v(\sigma) = 0$ otherwise. Then, Holant_Ω is counting the number of perfect matchings in H. Suppose instead $F_v(\sigma) = 1$ if $\text{wt}(\sigma) \leq 1$, and $F_v(\sigma) = 0$ otherwise, then we are counting *all* matchings of the graph H. Many counting problems not directly defined in terms of graphs can also be formulated as holant problems. For example, for the #SAT problem, we can draw a bipartite graph for the formula, where a variable node v_x is connected to a clause node v_C iff that variable x appears in the clause C. Then, to count #SAT, for each variable x, we assign its signature at v_x to be the equality function, and for each clause C, the signature at v_C is exactly its truth table.

Now we define the notion of an $\mathcal{F}$-gate $\Gamma = (H, \mathcal{F})$, where $H = (V, E, D)$ is a graph with some dangling edges D. (See Figure 2 for one example.) 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, however we allow more than one dangling edges for a node. Consider a graph with dangling edges $H = (V, E, D)$, where 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 $\Gamma = (H, \mathcal{F})$,

$$\Gamma(y_1, y_2, \ldots, y_n) = \sum_{x_1 x_2 \cdots x_m \in \{0,1\}^m} H(x_1 x_2 \cdots x_m y_1 y_2 \cdots y_n),$$

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 also 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.

If we have $\mathcal{F}$-gates $\Gamma_1 = (H_1, \mathcal{F}_1)$ and $\Gamma_2 = (H_2, \mathcal{F}_2)$, we can form a new $\mathcal{F}$-gate $\Gamma = (H, \mathcal{F})$ by merging some of their dangling edges to form regular edges. (See figure 3 for one example.) Suppose $|D_1| = k + p$, $|D_2| = k + q$ and the first k dangling edges of D_1 are merged with the first k dangling edges of D_2 correspondingly. Then

$$\Gamma(y_1 y_2 \ldots y_p z_1 z_2 \ldots z_q) = \sum_{x_1 x_2 \ldots x_k \in \{0,1\}^k} \Gamma_1(x_1 x_2 \cdots x_k y_1 y_2 \ldots y_p) \Gamma_2(x_1 x_2 \ldots x_k z_1 z_2 \ldots z_q).$$

We remark that a single node with a number of dangling edges can be viewed as the simplest $\mathcal{F}$-gate; also the whole signature grid $\Omega = (H, \mathcal{F})$ can be viewed as an $\mathcal{F}$-gate with zero dangling edges, and its Holant_Ω is exactly its signature (here it is only a single value).

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 Hamming weight i.

A counting problem is now generally defined to be the computation of Holant_Ω for some singature grids Ω from some family of graphs and signatures on its vertices.

We will mostly consider bipartite graphs $H = (V_1, V_2, E)$ here. For any general graph, we can make it bipartite by adding an additional vertex on each edge. The signature for each new vertex is the equality function $(1, 0, 0, 1)$ on 2 inputs, or in symmetric notation $[1, 0, 1]$.

We use $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to denote all the counting problems, expressed as holant problems on bipartite graphs $H = (V_1, V_2, E)$, where the graph H is from the graph family $\mathcal{H}$, and each signature for a vertex in V_1 or V_2 is from $\mathcal{G}$ or $\mathcal{R}$, respectively. If $\mathcal{H}$ consists of all bipartite graphs, we will simply use $\#\mathcal{G}|\mathcal{R}$ to denote the holant problem. 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) [5].

Much of the power of Holographic Algorithms is derived from custom-made cancelations in tensor spaces, as illustrated in Section 1. Let $T = [\mathbf{n}, \mathbf{p}]$, where $\mathbf{n}$ and $\mathbf{p}$ are two linearly independent vectors in the vector space $\mathbf{F}^{2^k}$ of dimension 2^k over $\mathbf{F}$. Such a basis is called a basis of size k. Then we can define a holographic reduction as follows. Suppose $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ and $\#\mathcal{H} : \mathcal{G}'|\mathcal{R}'$ are two holant problems defined for the same family of graphs. Suppose there exists a basis T and a polynomial time computable mapping σ, such that: for every generator $G \in \mathcal{G}$ of arity g, $\sigma(G)$ is an $\mathcal{F}$-gate with gk dangling edges (not necessary a single node) whose signature $G' \in \mathcal{G}'$, for every recognizer $R \in \mathcal{R}$ of arity r, $\sigma(R)$ is an $\mathcal{F}$-gate with rk dangling edges whose signature $R' \in \mathcal{R}'$, and $G' = T^{\otimes g} G$ and $R = R' T^{\otimes r}$. Then we say that there is a holographic reduction from $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to $\#\mathcal{H} : \mathcal{G}'|\mathcal{R}'$. (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 in a holographic reduction a signature grid Ω is mapped to a signature grid Ω', then*

$$\text{Holant}_{\Omega} = \text{Holant}_{\Omega'}.$$

The proof of this theorem is omitted here. It follows from general principles of contravariant and covariant tensors and their contractions [5]. In particular, if there is a holographic reduction from $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to $\#\mathcal{H} : \mathcal{G}'|\mathcal{R}'$, and there is a P-time algorithm for $\#\mathcal{H} : \mathcal{G}'|\mathcal{R}'$, then there is a P-time algorithm for $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$. Similarly, if the first holant problem is #P-hard, then so is the second.

In holographic algorithms proposed by Valiant [16], we reduce a given problem to the planar perfect matching problem, which has the FKT algorithm. Using that, many interesting problems are proved to be polynomial time solvable [16, 17, 2]. But in the framework of holographic reductions, we can reduce a given problem to any other holant problem, which is polynomial time solvable. In the next section, we will introduce another family of polynomial time solvable holant problems.

3 Fibonacci Gates

In this section, we introduce a new set of signatures called Fibonacci gates. Then we give a polynomial time algorithm for holant problems on these signatures.

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 $F(\sigma) = f_{\text{wt}(\sigma)}$, 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: We only need to consider connected graphs as inputs. If $H_1, H_2, \ldots, H_l$ are all the connected components of a graph H, obviously $\text{Holant}_H = \prod_{i=1}^{l} \text{Holant}_{H_i}$, so the holant problem is polynomial time computable iff it is polynomial time computable on all signature grids with Fibonacci gates where the underlying graph is connected.

Suppose H has n nodes and m edges. First we cut all the edges in the graph 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 polynomial time. 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. These two operations are illustrated in Fig. 4 and Fig. 5.

In the first case, the two dangling edges belong to two components before their merging (Fig. 4). For notational simplicity, we will consider exactly the case in Fig. 4, where gate F has 4 dangling edges y_1, y_2, y_3, z and gate G has 3 dangling edges y_4, y_5, z', and after merging the dangling edge z with z', we have a new gate H with 5 dangling edges y_1, y_2, y_3, y_4, y_5. We already know that 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 $\{y_1, y_2, y_3\}$ or from $\{y_4, y_5\}$, the value of H is clearly not changed. Suppose one input is from $\{y_1, y_2, y_3\}$ and the other is from $\{y_4, y_5\}$. Again by symmetry of F and G we may assume these two inputs are y_1 and y_4. Thus we will fix an arbitrary assignment for y_2, y_3 and y_5, and we want to show $H(0, y_2, y_3, 1, y_5) = H(1, y_2, y_3, 0, y_5)$.

As y_2, y_3 and y_5 are fixed, we can suppress them and denote $F_{y_1 z} = F(y_1, y_2, y_3, z)$, $G_{y_4 z} = G(y_4, y_5, z)$, and $H_{y_1 y_4} = H(y_1, y_2, y_3, y_4, y_5)$. Then by the definition of Holant, $H_{ab} = F_{a0}G_{b0} + F_{a1}G_{b1}$, for $a, b \in \{0, 1\}$. In particular, $H_{01} = F_{00}G_{10} + F_{01}G_{11}$, and $H_{10} = F_{10}G_{00} + F_{11}G_{01}$.

Because F and G are Fibonacci functions, $F_{11} = F_{01} + F_{00}$ and $G_{11} = G_{01} + G_{00}$. We have

$$H_{01} = F_{00}G_{10} + F_{01}G_{01} + F_{01}G_{00}, \quad \text{and} \quad H_{10} = F_{10}G_{00} + F_{01}G_{01} + F_{00}G_{01}.$$

By symmetry of F and G, $H_{01} = H_{10}$.

Now we show that $H(y_1, y_2, y_3, y_4, y_5)$ 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_4. For any fixed value of all the other inputs y_2, y_3, y_5, 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 $H_{00} + H_{01} = H_{11}$.

The above proof can clearly be generalized to any number of dangling edges in F and G.

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 following: $h_0 = f_0 g_0 + f_1 g_1$ 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 (Fig. 5). 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. ∎

Now we can use Fibonacci signatures and this algorithm to solve new problems.

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 $F^{\mathrm{T}} = T^{\otimes n} G$ (resp. $R = F T^{\otimes n}$).*

4 Realizability

In this section, we characterize all holant problems which can be solved by holographic algorithms with Fibonacci gates. Here we only consider bases of size 1, but the result is actually universal. In Section 6, we will prove a bases collapse theorem, which shows that any holographic algorithm with Fibonacci gates using bases of size k can be simulated on bases of size 1.

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 $R = [x_0, x_1, \ldots, x_n]$, we have

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

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

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

Theorem 4.1. *A symmetric signature $[x_0, x_1, \ldots, x_n]$ (for generator or recognizer) is realizable as a Fibonacci gate on some basis of size 1 iff there exist three constants a, b and c, such that $b^2 - 4ac \neq 0$, and for all k, where $0 \leq k \leq n - 2$,*

$$ax_k + bx_{k+1} + cx_{k+2} = 0. \tag{3}$$

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 $A = A'$, $B = B'$, $n_0 = 1$, $n_1 = 0$, $p_0 = \frac{\beta\phi - \alpha\bar{\phi}}{\phi - \bar{\phi}}$ and $p_1 = \frac{\alpha - \beta}{\phi - \bar{\phi}}$. 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 $A = \frac{x_0}{(\bar{\phi} - \phi)^n}$, $B = \frac{x_n}{(\phi - \bar{\phi})^n}$, $n_0 = \bar{\phi}$, $n_1 = -1$, $p_0 = \phi$ 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 a same basis of size 1 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 $T = \begin{bmatrix} 1 & \frac{\beta\phi - \alpha\bar{\phi}}{\phi - \bar{\phi}} \\ 0 & \frac{\alpha - \beta}{\phi - \bar{\phi}} \end{bmatrix}$ 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 Some Problems

In a formal sense, we already have a complete characterization of the power and expressibility of holographic algorithms with Fibonacci gates. In this section, we show some concrete problems formulated as holant problems which can be solved by our new holographic algorithms. All the problems, even restricted to planar structure are not solvable by original holographic algorithms with matchgates. Furthermore, some variations of the problem are $\#P$-complete, which will be proved in Section 7.

5.1 A Coloring Problem

Given a 3-regular graph, A 2-coloring (Black/White) of the edges is called *valid* if at each vertex the incident edges are not monochromatic. It is an *even* (resp. *odd*) coloring if it is a valid 2-coloring with an even (resp. odd) number of Black edges.
Input: A 3-regular graph.
Output: The number of even colorings minus the number of odd colorings.
Solution: For every edge we use the signature $[1, 0, -1]^{\mathtt{T}}$ as a generator, and for every vertex we use the signature $[0, 1, 1, 0]$ as a recognizer. By choosing $a = c = 1, b = -1$ in Theorem 4.2, we know that they are simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem.

Here we have some comments for this problem:

1. The same signatures $[1, 0, -1]$ and $[0, 1, 1, 0]$ are not simultaneous realizable as matchgates.
2. Computing the number of all valid colorings is #P-hard.
3. If we only consider planar graphs, then the number of all valid colorings *can* be computed in polynomial time by a holographic algorithm with matchgates. (Signatures $[1, 0, 1]$ and $[0, 1, 1, 0]$ are simultaneous realizable as matchgates.)

5.2 A Satisfiability Problem

We consider Rtw-CNF formulas, where "read twice" means that each variable appears in exact 2 clauses. Here we allow that the assignments of the same variable may differ from each other in two clauses. When one variable is assigned differently in two clauses, we call it a lying variable.
Input: A Rtw-CNF formula.
Output: The number of satisfying assignments with an even number of lying variables minus the number of satisfying assignments with an odd number of lying variables.
Solution: For each variable, if it appears both positive or both negative in two clauses, we use the

signature $[1,-1,1]^{\mathrm{T}}$ as a generator; if one is positive and one is negative, we use the signature $[-1,1,-1]^{\mathrm{T}}$ as a generator. For every clause with k variables, we use the signature $[0,1,1,\ldots,1]$ (k 1's) as a recognizer. By choosing $a=0, b=1, c=-1$ in Theorem 4.2, we know that they are all simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem.

5.3 Parity Problems

Here we consider a family of holant problems $\Gamma = (G, \mathcal{P})$, where each function in $\mathcal{P}$ has the property that its value only depends on the parity of its input. Thus we have signatures of the form $[a,b,a,b,\ldots]$. We show that these problems can all be solved by holographic algorithms with Fibonacci gates.
Input: A signature grid $\Gamma = (G, \mathcal{P})$ with parity signatures.
Output: Holant_Γ.
Solution: Applying the standard method to make the problem bipartite, for every edge we use the signature $[1,0,1]^{\mathrm{T}}$ as a generator. For every vertex, we keep its signature and is viewed as a recognizer. By choosing $a=1, b=0, c=-1$ in Theorem 4.2, we know that they are all simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem. We note that the choice of Fibonacci gates are different from the previous problems as different parameters a, b and c are used.

6 Bases Collapse

In this section, we prove a basis collapse theorem for holographic algorithms with Fibonacci gates. These results parallel those obtained for matchgate signatures [3, 4]. But the proof techniques are quite different. In the proof for matchgates, we construct size 1 bases and matchgates from the given high dimensional bases and matchgates, then prove that they realize the same signatures. Here we make use of Theorem 4.2, which characterizes all the signatures simultaneously realizable on a size 1 basis. We prove the collapse theorem by showing that signatures realizable on high dimensional bases also satisfy the conditions of Theorem 4.2. Here the properties of the golden ratio ϕ and its conjugate $\bar{\phi}$ are crucial.

Theorem 6.1. *Let $T = [\mathbf{n}, \mathbf{p}]$ be a basis of size k. 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 T. Then they are also simultaneously realizable on a basis of size 1.*

Proof: For any recognizer R_i, there exists some Fibonacci signature F_i such that $R_i = F_i T^{\otimes n}$.

$$\begin{aligned} R_i &= (A_i(1,\phi)^{\otimes nk} + B_i(1,\bar{\phi})^{\otimes nk})T^{\otimes n} \\ &= A_i(1,\phi)^{\otimes nk}T^{\otimes n} + B_i(1,\bar{\phi})^{\otimes nk}T^{\otimes n} \\ &= A_i(\langle (1,\phi)^{\otimes k}, \mathbf{n}\rangle, \langle (1,\phi)^{\otimes k}, \mathbf{p}\rangle)^{\otimes n} + B_i(\langle (1,\bar{\phi})^{\otimes k}, \mathbf{n}\rangle, \langle (1,\bar{\phi})^{\otimes k}, \mathbf{p}\rangle)^{\otimes n}, \end{aligned}$$

where we denote $\langle \cdot, \cdot \rangle$ for inner product. Let $\Phi = (1,\phi)^{\otimes k}$, $\bar{\Phi} = (1,\bar{\phi})^{\otimes k}$ be row vectors, and $R_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}]$. Then $x_h^{(i)} = A_i \langle \Phi, \mathbf{n}\rangle^{n_i-h}\langle \Phi, \mathbf{p}\rangle^h + B_i\langle \bar{\Phi}, \mathbf{n}\rangle^{n_i-h}\langle \bar{\Phi}, \mathbf{p}\rangle^h$. Choose $a = \langle \Phi, \mathbf{p}\rangle\langle \bar{\Phi}, \mathbf{p}\rangle = \mathbf{p}^{\mathrm{T}}\Phi^{\mathrm{T}}\bar{\Phi}\mathbf{p}$, $b = -\langle \Phi, \mathbf{p}\rangle\langle \bar{\Phi}, \mathbf{n}\rangle - \langle \Phi, \mathbf{n}\rangle\langle \bar{\Phi}, \mathbf{p}\rangle = -\mathbf{p}^{\mathrm{T}}\Phi^{\mathrm{T}}\bar{\Phi}\mathbf{n} - \mathbf{n}^{\mathrm{T}}\Phi^{\mathrm{T}}\bar{\Phi}\mathbf{p}$, and $c = \langle \Phi, \mathbf{n}\rangle\langle \bar{\Phi}, \mathbf{n}\rangle = \mathbf{n}^{\mathrm{T}}\Phi^{\mathrm{T}}\bar{\Phi}\mathbf{n}$, we can verify that $ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$ are satisfied for all recognizers.

For the generators, we have $F_j^{\mathrm{T}} = T^{\otimes m_j}G_j$. If $m_j = 1$, then G_j is of arity 1 and it is realizable on any size 1 basis. Now we consider the case $m_j = 2$. Then G_j is of arity 2 and we can assume that $G_j = [g_0, g_1, g_1, g_2]^{\mathrm{T}}$. Then we have

$$F_j^{\mathrm{T}} = (T \otimes T)G_j = g_0\mathbf{n}\otimes\mathbf{n} + g_1(\mathbf{n}\otimes\mathbf{p} + \mathbf{p}\otimes\mathbf{n}) + g_2\mathbf{p}\otimes\mathbf{p}.$$

Assume $F_j^{\mathrm{T}} = A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes 2k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes 2k}$, we have for any $x, y \in \{0,1\}^k$,

$$g_0 n_x n_y + g_1(n_x p_y + p_x n_y) + g_2 p_x p_y = A'_j \phi^{wt(x)+wt(y)} + B'_j \bar{\phi}^{wt(x)+wt(y)}. \quad (4)$$

By the sufficiency condition of Theorem 4.2, we only need to prove $cg_0 - bg_1 + ag_2 = 0$. Substituting a, b and c in the above equation, we have the following:

$$\begin{aligned}
& cg_0 - bg_1 + ag_2 \\
= & \sum_{x,y\in\{0,1\}^k} \left[g_0 n_x n_y \phi^{wt(x)} \bar{\phi}^{wt(y)} + g_1 (p_x n_y \phi^{wt(x)} \bar{\phi}^{wt(y)} + n_x p_y \phi^{wt(x)} \bar{\phi}^{wt(y)}) + g_2 p_x p_y \phi^{wt(x)} \bar{\phi}^{wt(y)} \right] \\
= & \sum_{x,y\in\{0,1\}^k} \phi^{wt(x)} \bar{\phi}^{wt(y)} (g_0 n_x n_y + g_1 (n_x p_y + p_x n_y) + g_2 p_x p_y) \\
= & \sum_{x,y\in\{0,1\}^k} \phi^{wt(x)} \bar{\phi}^{wt(y)} (A'_j \phi^{wt(x)+wt(y)} + B'_j \bar{\phi}^{wt(x)+wt(y)}) \\
= & A'_j \sum_{x,y\in\{0,1\}^k} \phi^{2wt(x)} (-1)^{wt(y)} + B'_j \sum_{x,y\in\{0,1\}^k} (-1)^{wt(x)} \bar{\phi}^{2wt(y)} \\
= & A'_j \sum_{x\in\{0,1\}^k} \phi^{2wt(x)} \sum_{y\in\{0,1\}^k} (-1)^{wt(y)} + B'_j \sum_{x\in\{0,1\}^k} (-1)^{wt(x)} \sum_{y\in\{0,1\}^k} \bar{\phi}^{2wt(y)} = 0,
\end{aligned}$$

proving the theorem for $m_j = 2$. Here we used the crucial relation $\phi\bar{\phi} = -1$.

Now we can assume that $m_j \geq 3$. Again we have $F_j^{\mathtt{T}} = T^{\otimes m_j} G_j$. Since $\text{rank}(T) = 2$, we can find some $\tilde{T} = \begin{bmatrix} \tilde{\mathbf{n}} \\ \tilde{\mathbf{p}} \end{bmatrix}$, such that $\tilde{T}T = I$. So $G_j = \tilde{T}^{\otimes m_j} F_j^{\mathtt{T}}$.

$$G_j = \tilde{T}^{\otimes m_j} F_j^{\mathtt{T}} = \begin{bmatrix} \tilde{\mathbf{n}} \\ \tilde{\mathbf{p}} \end{bmatrix}^{\otimes m_j} (A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes m_j k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes m_j k}) = A'_j \begin{bmatrix} \tilde{\mathbf{n}}\Phi^T \\ \tilde{\mathbf{p}}\Phi^T \end{bmatrix}^{\otimes m_j} + B'_j \begin{bmatrix} \tilde{\mathbf{n}}\bar{\Phi}^{\mathtt{T}} \\ \tilde{\mathbf{p}}\bar{\Phi}^{\mathtt{T}} \end{bmatrix}^{\otimes m_j}.$$

Denoting $\alpha = \tilde{\mathbf{n}}\Phi^{\mathtt{T}}$, $\beta = \tilde{\mathbf{p}}\Phi^{\mathtt{T}}$, $\gamma = \tilde{\mathbf{n}}\bar{\Phi}^{\mathtt{T}}$ and $\delta = \tilde{\mathbf{p}}\bar{\Phi}^{\mathtt{T}}$, we have $G_j = A'_j \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^{\otimes m_j} + B'_j \begin{bmatrix} \gamma \\ \delta \end{bmatrix}^{\otimes m_j}$.

Here if $A'_j = 0$ or $B'_j = 0$, then G_j can be decomposed into an arity 1 tensor and is realizable on any size 1 basis. So we can assume that $A'_j B'_j \neq 0$.

Substituting this back to $F_j^{\mathtt{T}} = T^{\otimes m_j} G_j$, we have

$$A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes m_j k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes m_j k} = A'_j (\alpha\mathbf{n} + \beta\mathbf{p})^{\otimes m_j} + B'_j (\gamma\mathbf{n} + \delta\mathbf{p})^{\otimes n}.$$

From Lemma 8.1, we know that $\alpha\mathbf{n} + \beta\mathbf{p} = \omega_1 \Phi^{\mathtt{T}}$, $\gamma\mathbf{n} + \delta\mathbf{p} = \omega_2 \bar{\Phi}^{\mathtt{T}}$, where ω_1 and ω_2 are roots of unity $\omega_1^n = \omega_2^n = 1$, or $\alpha\mathbf{n} + \beta\mathbf{p} = \omega'_1 \bar{\Phi}^{\mathtt{T}}$, $\gamma\mathbf{n} + \delta\mathbf{p} = \omega'_2 \Phi^{\mathtt{T}}$, where $(\omega'_1)^n = \frac{B'_j}{A'_j}$ and $(\omega'_2)^n = \frac{A'_j}{B'_j}$.

In the following proof we will assume the first case, the second case is similar. Since $G_j = [y_1^{(j)}, y_2^{(j)}, \ldots, y_{m_j}^{(j)}]$, we have $y_h^{(j)} = A'_j \alpha^{m_j - h} \beta^h + B'_j \gamma^{m_j - h} \delta^h$. Now we verify that $c y_h^{(j)} - b y_{h+1}^{(j)} + a y_{h+2}^{(j)} = 0$.

$$c y_h^{(j)} - b y_{h+1}^{(j)} + a y_{h+2}^{(j)} = \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} y_h^{(j)} + (\mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} + \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p}) y_{h+1}^{(j)} + \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p} y_{h+2}^{(j)}.$$

We substitute $y_h^{(j)}$. Here we only do the calculation on the A'_j part here. The B'_j part is similar.

$$\begin{aligned}
& \text{The coefficient of } A'_j \\
= & \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} \alpha^{m_j - h} \beta^h + (\mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} + \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p}) \alpha^{m_j - h - 1} \beta^{h+1} + \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p} \alpha^{m_j - h - 2} \beta^{h+2} \\
= & \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} \alpha^{m_j - h} \beta^h + \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p} \alpha^{m_j - h - 1} \beta^{h+1} + \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{n} \alpha^{m_j - h - 1} \beta^{h+1} + \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \mathbf{p} \alpha^{m_j - h - 2} \beta^{h+2} \\
= & \alpha^{m_j - h - 1} \beta^h \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} (\alpha\mathbf{n} + \beta\mathbf{p}) + \alpha^{m_j - h - 2} \beta^{h+1} \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} (\alpha\mathbf{n} + \beta\mathbf{p}) \\
= & \alpha^{m_j - h - 1} \beta^h \omega_1 \mathbf{n}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \Phi^{\mathtt{T}} + \alpha^{m_j - h - 2} \beta^{h+1} \omega_1 \mathbf{p}^{\mathtt{T}} \Phi^{\mathtt{T}} \bar{\Phi} \Phi^{\mathtt{T}} = 0.
\end{aligned}$$

The last equation uses the fact that $\bar{\Phi}\Phi^T = 0$. So by Theorem 4.2, we know that all the 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 a basis of size 1. This completes the proof. ∎

7 Hardness of Some Problems

In this section, we prove that counting all valid colorings in the coloring problem introduced in Section 5 is #P-complete. In the holant language, we prove the following theorem.

Theorem 7.1. *The holant problem* $\#(3\text{-}regular\text{-}graphs, \{[0, 1, 1, 0]\})$ *is #P-complete.*

Note that each edge in a 3-regular graph is assigned a truth value, which corresponds to a Black/White edge coloring, and at every vertex it is subjected to the Not-All-Equal gate $[0, 1, 1, 0]$.

Our starting point is #NAE-3SAT. It is #P-complete by the dichotomy theorem [7] on Constrained Satisfiability Counting Problems, which says that #CSP(Γ) is polynomial time computable, if all Boolean relations in Γ are affine, otherwise it is #P-complete.

Our first step is to represent this problem #NAE-3SAT as a holant problem over the signature set $\{[0, 1, 1, 0], [1, 0, 0, 1]\}$. Given an instance of #NAE-3SAT we represent it by a graph with variable nodes and NAE-clause nodes as follows: For each NAE-clause node, we use the NAE signature $[0, 1, 1, 0]$. For a variable node, if it occurs just once, we use the gadget in Fig. 6 to realize the equality gate $[1, 1]$. If a variable occurs twice, we can represent it by an edge, and append it with the gadget in Fig. 8 if it is once positive and once negative. If a variable occurs in $d \geq 3$ clauses as literals, we use the gadget in Fig. 7, and each negated literal occurrence is further appended the gadget in Fig. 8. Note that the Holant of this matchgrid evaluates to exactly the answer #NAE-3SAT. It follows that

Lemma 7.1. *The holant problem* $\#(3\text{-}regular\text{-}graph, \{[0, 1, 1, 0], [1, 0, 0, 1]\})$ *is #P-complete.*

The next step in the proof uses a technique called interpolations. Given a matchgrid Ω for the above holant problem, we will construct a sequence of matchgrids Ω_i for the holant problem #(3-regular-graph, $\{[0, 1, 1, 0]\}$). And we show that from a polynomial number of queries to $\text{Holtant}_{\Omega_i}$, where $i = 0, 1, \ldots,$ we can compute Holtant_{Ω} in polynomial time. This will prove Theorem 7.1.

Let n be the number of nodes in Ω with the signature $[1, 0, 0, 1]$. Let x_j be the number of truth assignments to all edges of Ω where there are exactly j nodes among those n nodes which receive an assignment $(0, 0, 0)$ or $(1, 1, 1)$ on its three incident edges, and on all other nodes in Ω (which have a signature $[0, 1, 1, 0]$) the truth assignment satisfies the local ternary NAE condition. Clearly the answer we seek is $\text{Holtant}_{\Omega} = x_n$.

Consider the sequence of $\{[0, 1, 1, 0]\}$-gate N_i in Fig. 9. The sequence N_i is recursively constructed. N_0 is simply a single node with the signature $[0, 1, 1, 0]$ and 3 dangling edges. For $i \geq 1$, we construct N_i by inscribing a copy of N_{i-1} inside of a triangle as shown in Fig. 9. Every node in the $\{[0, 1, 1, 0]\}$-gate N_i is assigned the signature $[0, 1, 1, 0]$ as we must. Obviously N_i has a symmetric signature as can be seen recursively by a geometric rotation or reflection directly on the picture. If we denote the signature of N_i by $[a_i, b_i, c_i, d_i]$, there is in fact a further symmetry. We claim that $a_i = d_i$ and $b_i = c_i$, which we prove recursively. Clearly N_0 has this property. Assume this holds for N_{i-1}. For N_i, if we flip all 0 and 1 bits, we get a 1-1 map on all satisfiable truth assignments counted in a_i or d_i and all satisfiable truth assignments counted in b_i or c_i. Thus the signature of N_i is $[a_i, b_i, b_i, a_i]$.

Now we construct our sequence of matchgrids Ω_i: For each of those n nodes in Ω with a signature $[0, 1, 1, 0]$, we replace it by a copy of N_i. It is easy to see that $\text{Holtant}_{\Omega_i} = \sum_{j=0}^{n} x_j a_i^j b_i^{n-j}$. Thus from the values $\text{Holtant}_{\Omega_i}$ for $i = 0, 1, \ldots, n$, we get a linear system in $x_0, x_1, \ldots, x_n$. Provided the coefficient matrix is non-singular, we can compute all these x_j, and $\text{Holtant}_{\Omega} = x_n$ is what we want.

To show that it is non-singular, we derive a recurrence relation for (a_i, b_i). To compute a_{i+1}, fix all three inputs $x_{i+1}, y_{i+1}, z_{i+1}$ of the gate N_{i+1} to 1, and then fix one of the eight possible assignments to x_i, y_i, z_i, the inputs of N_i, then multiply the number of satisfiable truth assignments for the remaining six edges in N_{i+1} with the value of N_i on this input (it is either a_i or b_i), then sum over all eight assignments, we get $a_{i+1} = 20a_i + 60b_i$. (This takes some careful accounting.) And similarly we have $b_{i+1} = 20a_i + 75b_i$. By this relation and Lemma 8.2 by Vadhan [14], we know that the coefficient matrix of the above linear system is non-singular (Vandemonde). This finishes the hardness proof.

Acknowledgments

References

[1] J-Y. Cai and Vinay Choudhary. Some Results on Matchgates and Holographic Algorithms. In Proceedings of ICALP 2006, Part I. Lecture Notes in Computer Science vol. 4051. pp 703-714. Also available at ECCC TR06-048, 2006.

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

[3] J-Y. Cai and Pinyan Lu. Bases Collapse in Holographic Algorithms. In the proceedings of IEEE Conference on Computational Complexity 2007, pp 292-304.

[4] J-Y. Cai and Pinyan Lu. Holographic Algorithms: The Power of Dimensionality Resolved. ICALP 2007: 631-642. Also available at Electronic Colloquium on Computational Complexity Report TR07-020.

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

[6] Matthew Cook. Networks of Relations. PhD thesis, 2005.

[7] Nadia Creignou, Miki Hermann: Complexity of Generalized Satisfiability Counting Problems. Inf. Comput. 125(1): 1-12 (1996),

[8] M. Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. J. Stat. Phys. 48 (1987) 121-134; erratum, 59 (1990) 1087-1088

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

[10] P. W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, (F. Harary, ed.), Academic Press, London, 43-110 (1967).

[11] E. Knill. Fermionic Linear Optics and Matchgates.
At http://arxiv.org/abs/quant-ph/0108033

[12] K. Murota. Matrices and Matroids for Systems Analysis, Springer, Berlin, 2000.

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

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

[15] L. G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal of Computing*, 31(4): 1229-1254 (2002).

[16] L. G. Valiant. Holographic Algorithms (Extended Abstract). In *Proc. 45th IEEE Symposium on Foundations of Computer Science*, 2004, 306–315. A more detailed version appeared in ECCC Report TR05-099.

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

8 Appendix

Lemma 8.1. *Let* $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ *be four vectors and* $\mathbf{c}, \mathbf{d}$ *are linear independent with each other. If for some* $n \geq 3$ *we have* $\mathbf{a}^{\otimes n} + \mathbf{b}^{\otimes n} = \mathbf{c}^{\otimes n} + \mathbf{d}^{\otimes n}$*. Then* $\mathbf{a} = \omega_1 \mathbf{c}$ *and* $\mathbf{b} = \omega_2 \mathbf{d}$ *or* $\mathbf{a} = \omega_1 \mathbf{d}$ *and* $\mathbf{b} = \omega_2 \mathbf{c}$ *for some* $\omega_1^n = \omega_2^n = 1$*.*

Proof: First we prove that $\mathbf{a}, \mathbf{b} \in span\{\mathbf{c}, \mathbf{d}\}$. Since $\mathbf{c}, \mathbf{d}$ are linear independent with each other, so if the vector space is of dimension 2, this is obliviously true. Now we assume that the dimension is larger than 2. Now we assume for contradiction that $\mathbf{bc}, \mathbf{d}$ are linear independent. Then we can design a linear transform S to map $\mathbf{b}$ to $\mathbf{e_3}$ as well as $\mathbf{c}$ to $\mathbf{e_1}$ and $\mathbf{d}$ to $\mathbf{e_2}$. also $\mathbf{a}$ maps to $\sum c_i \mathbf{e_i}$. We apply the operator $S^{\otimes n}$to the equation in the assumption and we get

$$(\sum c_i \mathbf{e_i})^{\otimes n} + \mathbf{e_3}^{\otimes n} = \mathbf{e_1}^{\otimes n} + \mathbf{e_2}^{\otimes n}.$$

Comparing the coefficient of $\mathbf{e_3}^{\otimes n}$, we know that $c_3 \neq 0$. Comparing the coefficient of $\mathbf{e_3}^{\otimes n-1}\mathbf{e_1}$, we have $c_3^{n-1}c_1 = 0$, which implies $c_1 = 0$, Then comparing the coefficient of $\mathbf{e_1}^{\otimes n}$, we have $0 = 1$, a contradiction.

So now we may assume $\mathbf{a}$ and $\mathbf{b}$ are both linearly dependent on $\mathbf{c}$ and $\mathbf{d}$. In terms of S, we may assume

$$S\mathbf{a} = a_1\mathbf{e_1} + a_2\mathbf{e_2},$$

$$S\mathbf{b} = b_1\mathbf{e_1} + b_2\mathbf{e_2}.$$

And we have

$$(a_1\mathbf{e_1} + a_2\mathbf{e_2})^{\otimes n} + (b_1\mathbf{e_1} + b_2\mathbf{e_2})^{\otimes n} = \mathbf{e_1}^{\otimes n} + \mathbf{e_2}^{\otimes n}.$$

So we have

$$\begin{cases} a_1^n + b_1^n = 1, \\ a_2^n + b_2^n = 1, \\ a_1^k a_2^{n-k} + b_1^k b_2^{n-k} = 0. \quad k = 1, 2, \cdots, n-1 \end{cases}$$

If $a_1 = 0$ then we know $b_1 b_2 = 0$ from the third equation. But for the first equation, we know $b_1^n = 1$, so $b_1 \neq 0$. So we have $b_2 = 0$ and then $a_2^n = 1$. As a result we have $S\mathbf{a} = a_2\mathbf{e_2} = a_2 S\mathbf{d}$ and $S\mathbf{b} = b_1\mathbf{e_1} = a_2 S\mathbf{c}$. So $\mathbf{a} = a_2\mathbf{d}$ and $\mathbf{b} = b_1\mathbf{c}$. This is one of the desired cases for the lemma. Similarly we can show that $a_2 = 0$ will imply the other desired case for the lemma.

The remaining cases is $a_1 a_2 \neq 0$. We prove that this is impossible. Assume for contradiction that $a_1 a_2 \neq 0$, then $b_1 b_2 \neq 0$ from the third equation. Since $n \geq 3$, we can choose $k = 1, 2$ in the third equation. Combining the two equations, we will have $a_1/a_2 = b_1/b_2$ and then $a_1^n = -b_1^n$, but this is a contradiction with the first equation. This contradiction completes the proof. ∎

Lemma 8.2 (Vadhan)**.** *Let* A, B, C, D, x_0 *and* y_0 *be rational numbers. Define the sequences* (x_i, y_i) *recursively by* $x_{i+1} = Ax_i + By_i$ *and* $y_{i+1} = Cx_i + Dy_i$*. Then the sequence* $\{z_i = x_i/y_i\}$ *never repeats as long as all of the following conditions hold:*

$$\begin{cases} AD - BC & \neq & 0 \\ D^2 - 2AD + A^2 + 4BC & \neq & 0 \\ D + A & \neq & 0 \\ D^2 + AD + A^2 + BC & \neq & 0 \\ D^2 + A^2 + 2BC & \neq & 0 \\ D^2 - AD + A^2 + 3BC & \neq & 0 \\ By_0^2 - Cx_0^2 - (A-D)x_0y_0 & \neq & 0 \end{cases}.$$

Figures

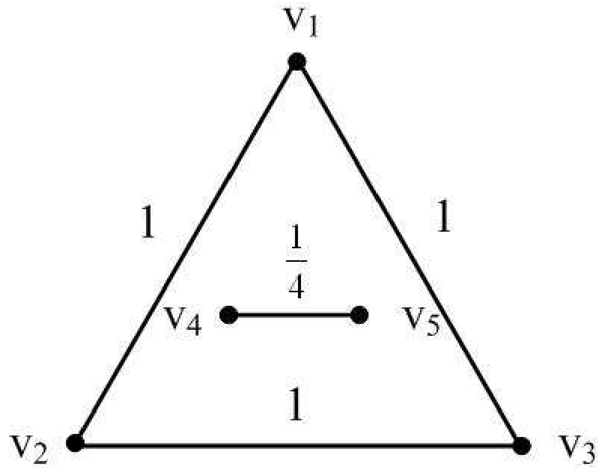

Figure 1: A matchgate with signature $\frac{1}{4}(0,1,1,0,1,0,0,1)$.

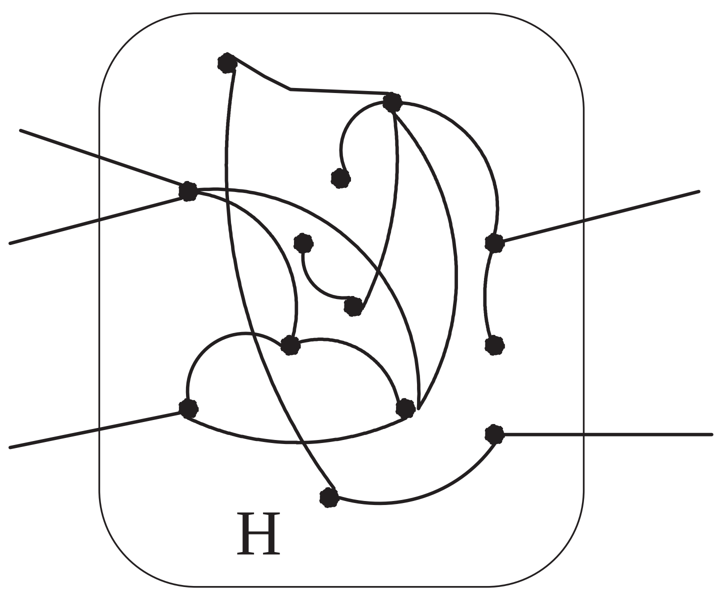

Figure 2: An example of an $\mathcal{F}$-gate with five dangling edges.

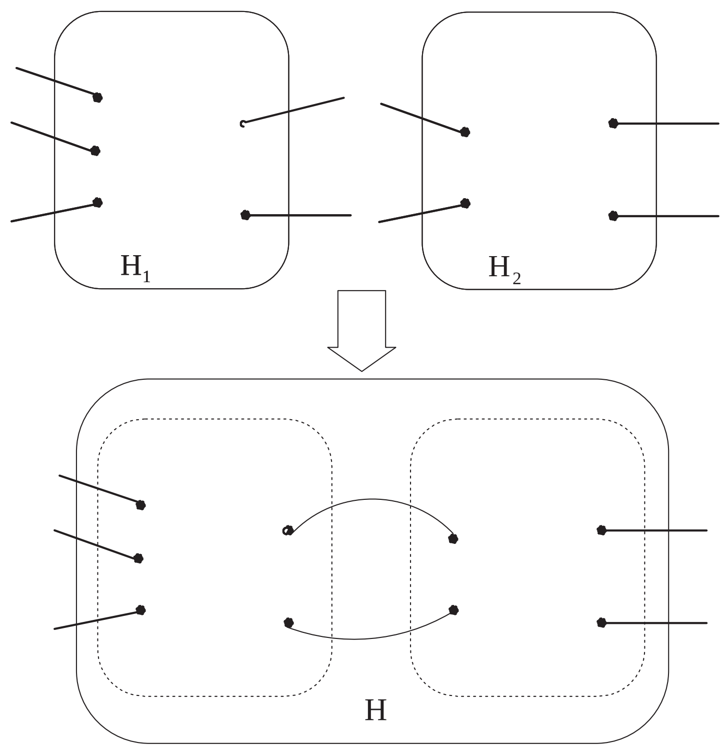

Figure 3: Two $\mathcal{F}$-gates compose into one.

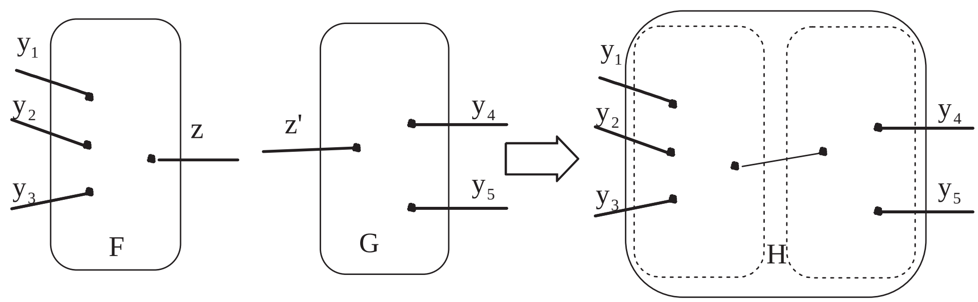

Figure 4: First operation.

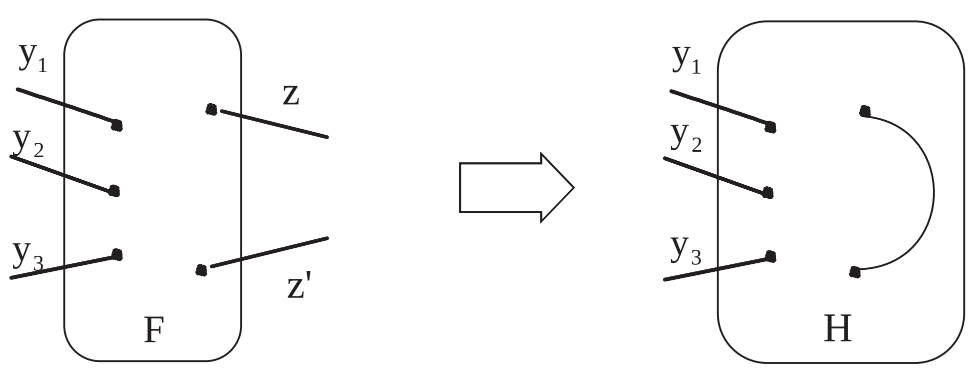

Figure 5: Second operation.

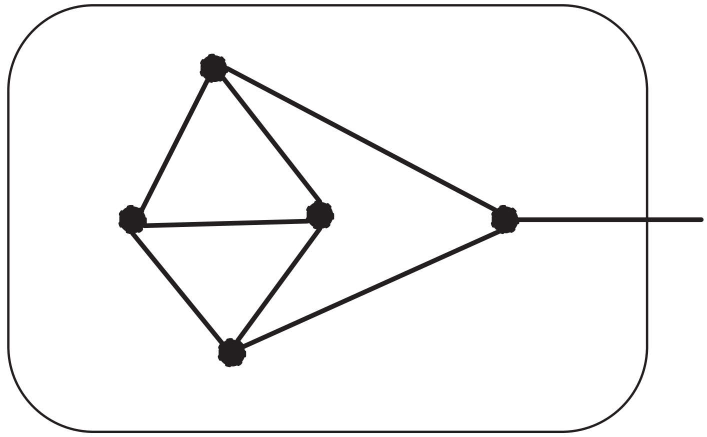

Figure 6: The signature for every node in this $\{[0,1,1,0],[1,0,0,1]\}$-gate is $[1,0,0,1]$. And the signature of the gate is $[1,1]$.

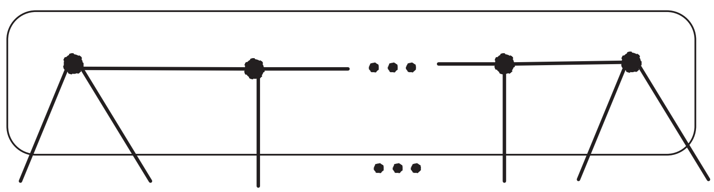

Figure 7: This gate realizes the equality signature $[1,0,0,\ldots,0,1]$ of more than 3 variables. The signature for every node in this is $[1,0,0,1]$. There are $d-2$ nodes in the gate if we want to realize the equality signature for d variables.

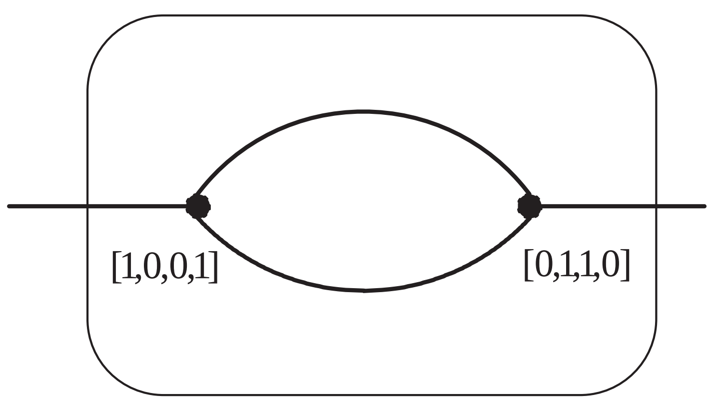

Figure 8: An $\{[0,1,1,0],[1,0,0,1]\}$-gate with signature $[0,1,0]$.

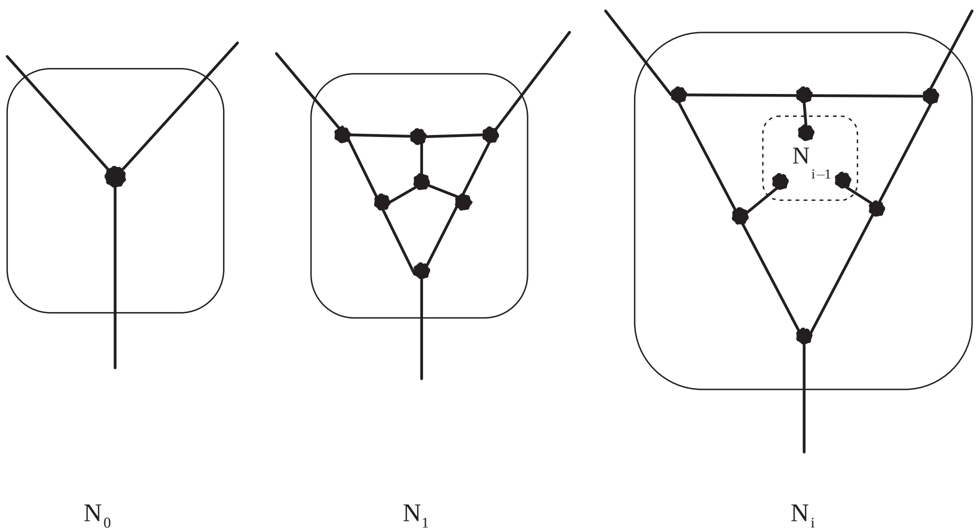

Figure 9: The recursive construction of the $\{[0, 1, 1, 0]\}$-gates N_i. The signature for every node in these gates is $[0, 1, 1, 0]$.