

# Normal Factor Graphs and Holographic Transformations

Ali Al-Bashabsheh and Yongyi Mao

University of Ottawa

School of Information Technology and Engineering

## Abstract

This paper stands at the intersection of two distinct lines of research. One line is “holographic algorithms,” a powerful approach introduced by Valiant for solving various counting problems in computer science; the other is “normal factor graphs,” an elegant framework proposed by Forney for representing codes defined on graphs. We introduce the notion of holographic transformations for normal factor graphs, and establish a very general theorem, called the generalized Holant theorem, which relates a normal factor graph to its holographic transformation. We show that the generalized Holant theorem on the one hand underlies the principle of holographic algorithms, and on the other hand reduces to a general duality theorem for normal factor graphs, a special case of which was first proved by Forney. In the course of our development, we formalize a new semantics for normal factor graphs, which highlights various linear algebraic properties that potentially enable the use of normal factor graphs as a linear algebraic tool.

## 1 Introduction

The formulation of codes on graphs and the invention of iterative decoding algorithms for graphical codes have undoubtedly revolutionized coding theory. In this research area, the introduction of normal graphs [1] and their duality properties are arguably one of the most elegant and profound results.

In his celebrated paper [1], Forney introduced the notion of *normal realizations*, represented by *normal graphs*, as generalized state realizations of codes. In a normal graph, each vertex represents a local (group) code constraint, and each edge represents a variable that is involved in either one or two of the constraints. A variable involved in two constraints, called a *state* variable, is represented by a regular edge, namely, an edge connecting two vertices; the two vertices correspond to the two constraints involving the variable. A variable involved in only one constraint, called a *symbol* variable, is represented by a “dangling edge,”<sup>1</sup> namely, an edge incident on only one vertex; the vertex corresponds to the single constraint involving the variable. The *global behavior* represented by the normal graph is the set of all symbol-state configurations satisfying all local constraints, and the realized code is the set of all symbol configurations that participate in at least one symbol-state configuration in the global behavior. Forney showed in [1] that codes realized by Tanner graphs [2] or Wiberg-type graphs [3, 4], in which a variable may in general be involved in an unrestricted number of constraints, can be converted to normal graphs by properly replicating some variables.

---

<sup>1</sup>In Forney [1], such edges are called “half-edges”.

Normal realizations of codes have a fundamental duality property. By introducing a simple local “dualization” procedure that converts each local code in a normal realization to its dual code and inserts additional “sign inverters,” Forney proved a *normal graph duality theorem* [1], which shows that the dualized normal graph realizes the dual code.

The notion of a normal graph may be extended to the notion of a *normal factor graph*, or *Forney-style factor graph* [5, 6], which uses an identical graphical representation, but in which the graph vertices no longer represent constraints. Instead, in a normal factor graph, each vertex represents a local *function* involving precisely the variables represented by the edges incident to the vertex. Treated as a factor graph [7] with particular variable-degree restrictions, and interpreted using the standard semantics of factor graphs, a normal factor graph represents a multivariate function that factors as the product of all of the local functions that are represented by the graph vertices. When each local function in the normal factor graph is the indicator function of a local code constraint, the represented function is the indicator function of the global behavior. This makes normal factor graph representations of codes equivalent to normal graphs, and allows the translation of the normal graph duality theorem to an equivalent theorem for normal factor graphs that represent codes.

Among the first to recognize the profound value of normal graphs, Koetter [8] applied normal graphs (or normal factor graphs for codes) and the duality theorem in a study of trellis formations [9], which gives a necessary and sufficient condition of mergeability and a polynomial-time algorithm for deciding whether a trellis formation contains mergeable vertices. In addition, Koetter *et al.* [10] extended the applications of normal graphs from channel codes to network codes, and used the fundamental duality theorem to establish a reversibility theorem in network coding.

In a seemingly distant research area of complexity theory, Valiant proved the tractability of families of combinatorial problems for which no polynomial-time solvers were known previously in a ground-breaking work [11]. In this paper, Valiant develops a very powerful family of algorithms, which he calls *holographic algorithms*, to solve such problems. Holographic algorithms are based on the concept of “holographic reduction,” and are governed by a fundamental theorem that Valiant calls the *Holant theorem*.

Although in his original work [11] Valiant dealt only with transforming a product of functions to a specific form, the Holant theorem establishes a principle for transforming an arbitrary product of functions to another product of functions such that the sum over the configuration space is unchanged. Consequently, when computing the sum of a product of functions, the Holant theorem provides a family of transformations that convert the product to a different one, for which the sum may be efficiently computable.

Since many problems in coding and information theory require the computation of sums of products (such as in decoding error correction codes and in computing certain capacities), holographic algorithms become a potentially powerful tool for the information theory community. Indeed, in [12], Schwartz and Bruck showed that certain constrained-coding capacity problems may be solved in polynomial time using holographic algorithms.

This paper stands at the intersection of the above-mentioned two lines of research, bridging the two areas by unifying Valiant’s holographic reduction and Forney’s normal-graph dualization with the notion of “holographic transformations,” a term that we coin in this paper. The focus of this paper will be on general normal factor graphs, in which the vertices may represent arbitrary functions rather than just indicator functions.

We first introduce a new semantics for normal factor graphs, which we call the “exterior-function semantics.” In the exterior-function semantics, instead of letting the graph represent a product of the local functions, we let it represent a sum of products of local functions, which we call the “exterior function.” In this setting, a normal factor graph may be viewed as an expression, or realization, of its exterior function in terms of a “sum of products.” In fact, the notion of “sum of products” is the key connection between Forney’s normal graph duality theorem and Valiant’s Holant theorem: in the case of Valiant [11], this “sum of products” is the number of configurations that needs to be computed, and in the case of Forney [1] (when using normal factor graphs rather than normal graphs to represent codes), this “sum of products” is a code indicator function (possibly up to a scale factor). In this new framework, a holographic transformation is defined as a transformation of a normal factor graph that changes all local functions subject to certain conditions, and converts the normal factor graph to a structurally identical one.

The main result of this paper is what we call the *generalized Holant theorem*, relating a normal factor graph and its holographic transformations in terms of their realized functions. On the one hand, we show that the Holant theorem in [11] is a special case of the generalized Holant theorem. On the other hand, we prove a general duality theorem for normal factor graphs as a corollary of the generalized Holant theorem. This duality theorem reduces to Forney’s original normal graph duality theorem for normal factor graphs that represent codes.

Another result of our development is a new understanding of normal factor graphs from a linear algebraic perspective. More specifically, the exterior-function semantics associates a normal factor graph with a “sum-of-products” form, and a sum-of-products form may be regarded as a linear algebraic expression such as a vector dot product, matrix product or tensor product. In contrast to conventional notations for algebraic expressions, in which re-ordering terms results in illegitimate or different mathematical expressions, sum-of-products forms and their normal factor graph representations eliminate the need to properly order terms (*i.e.*, the local functions). This allows a transparent development of our results. To us, normal factor graphs with the exterior-function semantics appear to be a very natural and intuitive language for linear algebra, and may potentially be useful in a wide variety of applications.

The holographic transformations introduced in this paper equip normal factor graphs with a rich family of linear transformations, potentially enabling normal factor graphs to serve as a more general analytic framework and computational tool. The power of these transformations, in addition to providing a fundamental duality theorem in coding theory, has also been hinted at by the great power of holographic algorithms (see, *e.g.*, [11, 12, 13, 14, 15, 16]).

This paper was initially motivated by our recognition of the connection between Forney’s normal graph duality theorem and Valiant’s Holant theorem. Interactions with the editors of this paper (G. D. Forney, Jr. and P. Vontobel) and communications with Forney on his concurrent development of “partition functions of normal factor graphs” [17] have also provided important inspirations for our development. Indeed, much of our development shares similarities with Forney’s approach in [17], which also proves the general normal factor graph duality theorem.

During the review process, we learned from the reviewers and the editors of this paper that the techniques used to establish our results are similar to those in some previous literature, including, for example, the notion of “loop calculus” in [18, 19, 20, 21], and the “opening/closing the box” approach in [22, 23, 5]. We shall give proper credit to these authors as we proceed in this paper.

The remainder of the paper is structured as follows. Section 2 formalizes the exterior-function semantics for normal factor graphs, discusses its linear algebraic interpretations, introduces the notion of holographic transformations, and establishes the generalized Holant theorem. Section 3 discusses applications of holographic transformations, including deriving the general normal factor graph duality theorem as a corollary to generalized Holant theorem, and explaining the principle of holographic reduction. The paper is briefly concluded in Section 4.

## 2 Normal Factor Graphs and Holographic Transformations

The term “normal factor graph” has been used in the literature with various meanings. In particular, normal factor graphs as defined in [1] can be easily confused with normal graphs, normal factor graphs for codes, or graphs in which variables are represented by variable vertices. Making a joint effort with Forney [17], we advocate in this paper a more rigorous use of the term “normal factor graph,” and introduce a new semantics, the “exterior-function” semantics,<sup>2</sup> that defines what a normal factor graph means. To us, this new semantics is quite appealing, since it allows clean development of various graph properties, and has an elegant linear algebraic perspective.

### 2.1 Normal Factor Graphs: The Exterior-Function Semantics

Formally, a *normal factor graph* (NFG) is a graph  $(V, E)$ , with vertex set  $V$  and edge set  $E$ , where the edge set  $E$  consists of two kinds of edges, a set  $E^{\text{int}}$  of ordinary edges, each connecting two vertices, and a set  $E^{\text{ext}}$  of “dangling edges,” each having one end attached to a vertex and the other end free.<sup>3</sup> Each edge  $e \in E$  represents a variable  $x_e$  taking values from some finite alphabet<sup>4</sup>  $\mathcal{X}_e$ ; sometimes we may alternatively say that the edge  $e$  represents the alphabet  $\mathcal{X}_e$  when we do not wish to specify the variable name. Each vertex  $v$  represents a complex-valued function<sup>5</sup>  $f_v$  on the cartesian product  $\mathcal{X}_{E(v)} := \prod_{e \in E(v)} \mathcal{X}_e$ , where  $E(v)$  is the set of all edges incident to  $v$ . If we denote the set  $\{f_v : v \in V\}$  of functions by  $f_V$ , then the NFG is specified by the tuple  $(V, E^{\text{int}}, E^{\text{ext}}, f_V)$ .

When treated as a factor graph under the conventional semantics [7], the NFG  $\mathcal{G} = (V, E^{\text{int}}, E^{\text{ext}}, f_V)$  represents the product  $\prod_{v \in V} f_v(x_{E(v)})$  of all functions in  $f_V$ . This product, expressing a function on  $\mathcal{X}_E := \prod_{e \in E} \mathcal{X}_e$ , will be called the *interior function*<sup>6</sup> realized by the NFG. Here we have used the standard “variable set” notation  $x_{E(v)}$  to denote the set of variables  $\{x_e : e \in E(v)\}$ .

Now we introduce a new semantics for NFGs: instead of letting an NFG represent its interior function, we let it represent the interior function summed over all variables represented by regular edges. We call

---

<sup>2</sup>The same semantics is also presented in the concurrent development of Forney [17], in which what we call “exterior functions” are called “partition functions.”

<sup>3</sup>More formally, such dangling edges are hyperedges of degree 1, and thus strictly speaking a NFG is a hypergraph rather than a graph.

<sup>4</sup>It is possible to generalize the results of this paper to the cases where the alphabets are infinite or uncountable, although this involves some subtle technicalities.

<sup>5</sup>All functions in this paper are complex-valued. However, all results can be generalized to  $\mathbb{F}$ -valued functions, where  $\mathbb{F}$  is an arbitrary field.

<sup>6</sup>In the conventional factor graph literature, an interior function is referred to as a “global function.”

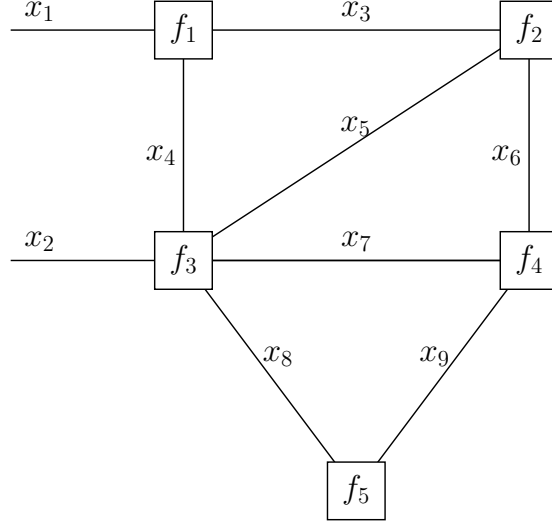


Figure 1: A normal factor graph (NFG)  $\mathcal{G}$ .

this function the *exterior function* realized by the NFG. More precisely, the exterior function realized by the NFG  $\mathcal{G} = (V, E^{\text{int}}, E^{\text{ext}}, f_V)$  is the function

$$Z_{\mathcal{G}}(x_{E^{\text{ext}}}) := \sum_{x_{E^{\text{int}}}} \prod_{v \in V} f_v(x_{E(v)}). \quad (1)$$

Thus the exterior function involves only the external variables, represented by the dangling edges. Letting the NFG  $\mathcal{G}$  express the function  $Z_{\mathcal{G}}$  as defined in (1) is what we call the *exterior-function semantics*, which we will use throughout this paper.

In this setting, one may view an NFG as an *expression*, or *realization*, of a function (the exterior function) that is given in “sum-of-products” form, as in (1), where each variable is involved in either one function or two functions, and the summation is over all variables that are involved in two functions. Since a variable involved in two functions (represented by a regular edge) is “invisible” in the exterior function, we call it an *internal variable*. In contrast, a variable involved only in one function (represented by a dangling edge) remains visible in the exterior function, and is called an *external variable*.

An example of an NFG is given in Figure 1, which realizes the exterior function

$$Z_{\mathcal{G}}(x_1, x_2) = \sum_{x_3, \dots, x_9} f_1(x_1, x_3, x_4) f_2(x_3, x_5, x_6) f_3(x_2, x_4, x_5, x_7, x_8) f_4(x_6, x_7, x_9) f_5(x_8, x_9).$$

At first glance, NFGs and this semantics may appear to impose a restriction on which sum-of-products forms are representable. We note, however, that *any* sum-of-products form can be straightforwardly converted to one that directly corresponds to an NFG. This requires only that we properly replicate variables, using a “normalization” procedure similar to that of Forney in [1] for converting a factor graph to a normal graph. (The Appendix gives a detailed account of this procedure.) For this reason, using NFGs to represent sum-of-products forms entails no loss of expressive power.

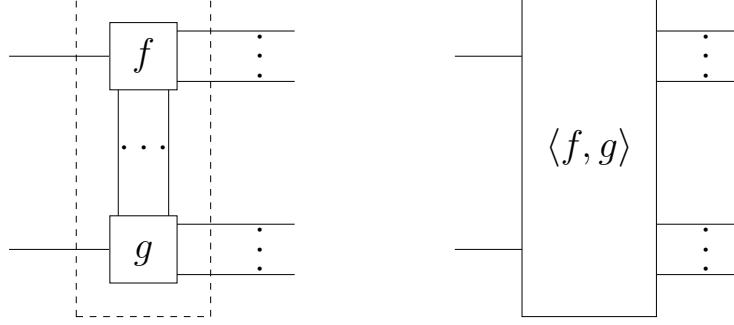


Figure 2: Vertex Grouping/Splitting Procedure. Left to right: Vertex Grouping; right to left: Vertex Splitting.

For notational convenience, we may denote the sum-of-products form in (1) by<sup>7</sup>

$$\langle f_1(x_{E(1)}), f_2(x_{E(2)}), \dots, f_{|V|}(x_{E(|V|)}) \rangle,$$

if  $V$  is identified with the set  $\{1, 2, \dots, |V|\}$ . Due to the commutativity of both multiplication and summation and the distributive law relating the two operations, it is easy to see that any ordering of the arguments of  $\langle \cdot, \cdot, \dots, \cdot \rangle$  expresses the same function. Consequently, we may write the sum-of-products form more compactly as  $\langle f_v(x_{E(v)}) : v \in V \rangle$ , or even as  $\langle f_v : v \in V \rangle$ , if no ambiguity results.

## 2.2 Exterior-Function-Preserving Procedures

The exterior-function semantics of NFGs allows us to identify immediately several elementary graph manipulation procedures that preserve the exterior function.

**Vertex Grouping/Splitting Procedure** In a Vertex Grouping Procedure, two vertices representing functions  $f$  and  $g$  are grouped together, and the group is replaced by a vertex representing the function  $\langle f, g \rangle$ . In a Vertex Splitting Procedure, a vertex representing a function that can be expressed by the sum-of-products form  $\langle f, g \rangle$  is replaced by an NFG representing  $\langle f, g \rangle$ . Figure 2 shows this pair of procedures.

**Lemma 1** *Applying a Vertex Grouping or Vertex Splitting Procedure to an NFG preserves the realized exterior function.*

*Proof:* This lemma holds because these procedures simply correspond to conversions between sum-of-products forms  $\langle f, g, h_1, \dots, h_m \rangle$  and  $\langle \langle f, g \rangle, h_1, h_2, \dots, h_m \rangle$ , which evidently express the same function.  $\square$

We note that this pair of procedures were first introduced by Loeliger in [5, 6], who refers to Vertex Grouping as “closing the box,” and to Vertex Splitting as “opening the box.” However, in [5, 6] these procedures are used to interpret an NFG (in the original semantics) as a flexible hierarchical model, and to explain message-passing algorithms, rather than in the context of the exterior-function semantics.

<sup>7</sup>In the case of two arguments, say functions  $f$  and  $g$ , the sum-of-products form  $\langle f, g \rangle$  should not be confused with the Hermitian inner product of  $f$  and  $g$ , whose definition requires complex conjugation of one of the two arguments.

Note that the Vertex Grouping Procedure may be applied recursively to an arbitrary number of vertices, say  $f_1, f_2, \dots, f_m$ , so that these functions are replaced by a single function realized by  $\langle f_1, f_2, \dots, f_m \rangle$ . The resulting NFG still realizes the same exterior function. Similarly, the reverse Vertex Splitting Procedure also preserves the exterior function. For these reasons, when we draw a dashed box (as in Figure 2, left) to group some vertices, we may freely interpret the NFG as the equivalent NFG in which the box is replaced by a single vertex representing the function realized by the box.

**Equality Insertion/Deletion Procedure** For any finite alphabet  $\mathcal{X}$ , let  $\delta_=_$  denote the  $\{0,1\}$ -valued function on  $\mathcal{X} \times \mathcal{X}$  which evaluates to 1 if and only if the two arguments of the function are equal. That is,  $\delta_=_$  is an “equality indicator function.” In an Equality Insertion Procedure, a  $\delta_=_$  function is inserted into an edge; in an Equality Deletion Procedure, a  $\delta_=_$  is deleted and the two edges originally connected to the function are joined. Figure 3 shows this pair of procedures.



Figure 3: Equality Insertion/Deletion Procedure. Left to right: Equality Insertion; right to left: Equality Deletion. The edges may be regular or dangling; the vertex labelled with “=” represents the function  $\delta_=_$ .

**Lemma 2** *Applying an Equality Insertion or Deletion Procedure to an NFG preserves the realized exterior function.*

*Proof:* This result is a corollary of Lemma 1, and simply follows from the fact that if the  $\delta_=_$  function is inserted into an edge incident to a function  $f$ , then we may group  $f$  with  $\delta_=_$  and replace the sum-of-products form  $\langle f, \delta_=_ \rangle$  with the function it expresses, namely  $f$ .  $\square$

**Dual Vertex Insertion/Deletion Procedure** Let  $\mathcal{X}$  and  $\mathcal{Y}$  be two finite alphabets and  $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$  and  $\hat{\Phi} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$  be two functions. Then we say that  $\Phi$  and  $\hat{\Phi}$  are *dual* with respect to the alphabet  $\mathcal{Y}$ , and call  $\mathcal{Y}$  the *coupling* alphabet, if  $\langle \Phi(x, y), \hat{\Phi}(x', y) \rangle = \delta_=(x, x')$  for every  $x, x' \in \mathcal{X}$ . In the case when  $\mathcal{X}$  and  $\mathcal{Y}$  have the same cardinality, we also call the vertices representing  $\Phi$  and  $\hat{\Phi}$  *transformers*, a term which will be justified in Section 2.3. In a Dual Vertex Insertion Procedure, we insert into an edge representing alphabet  $\mathcal{X}$  a dual pair of functions  $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$  and  $\hat{\Phi} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$ , with  $\mathcal{Y}$  being the coupling alphabet, and let the edge connecting the two functions represent  $\mathcal{Y}$ . A Dual Vertex Deletion Procedure is the reverse of a Dual Vertex Insertion Procedure, in which we delete a pair of dual functions and the edge connecting them, and then join the ends of the two cut edges.

**Lemma 3** *Applying a Dual Vertex Insertion or Deletion Procedure to an NFG preserves the realized exterior function.*

*Proof:* The Dual Vertex Insertion Procedure is equivalent to first inserting a  $\delta_=_$  function in the edge and then splitting the  $\delta_=_$  function into a pair of dual functions. The lemma then follows from Lemmas 1 and 2.  $\square$



Figure 4: Dual Vertex Insertion/Deletion Procedure. Left to right: Dual Vertex Insertion; right to left: Dual Vertex Deletion. The edges may be regular or dangling; the oppositely oriented triangular vertices represent a dual pair of functions.

### 2.3 A Linear Algebraic Perspective

Before we proceed to introduce holographic transformations, we pause to interpret NFGs from a linear algebraic perspective.

We will denote the set of all complex-valued functions on a finite alphabet  $\mathcal{X}$  by  $\mathbb{C}^{\mathcal{X}}$ . It is well known that  $\mathbb{C}^{\mathcal{X}}$  is isomorphic to the vector space  $\mathbb{C}^{|\mathcal{X}|}$ : after imposing an order on  $\mathcal{X}$ , one can arrange the values of any function  $f \in \mathbb{C}^{\mathcal{X}}$  as a vector in  $\mathbb{C}^{|\mathcal{X}|}$  according to that order. Similarly, depending on the structure of  $\mathcal{X}$ , the function  $f$  may also be viewed as a matrix, or as its higher-dimensional generalization, namely a tensor; if  $\mathcal{X}$  is the cartesian product  $\mathcal{X}_1 \times \mathcal{X}_2$  of some alphabets  $\mathcal{X}_1$  and  $\mathcal{X}_2$ , then  $f$  may be regarded as a matrix; if  $\mathcal{X}$  is a multifold cartesian product of alphabets, then  $f$  may be viewed as a multi-dimensional array, or as a tensor. On the other hand, conventional linear algebraic objects like vectors, matrices and tensors may be alternatively regarded as multivariate functions. In particular, a tensor with  $n$  indices may be identified with a multivariate function involving  $n$  variables. From this perspective, any sum-of-products form corresponding to an NFG may be viewed as a linear algebraic expression.

In the simplest case, consider a sum-of-products form  $\langle f(x_I), g(x_J) \rangle$  involving exactly two functions  $f : \mathcal{X}_I \rightarrow \mathbb{C}$  and  $g : \mathcal{X}_J \rightarrow \mathbb{C}$ , where  $I$  and  $J$  are two finite index sets, possibly intersecting, and where for every  $i \in I \cup J$ ,  $\mathcal{X}_i$  is an arbitrary finite alphabet. It is straightforward to verify the following propositions:

- If  $I = J \neq \emptyset$ , then  $\langle f, g \rangle$  is the (non-Hermitian) vector inner product, or dot product,  $f \cdot g$ , where  $f$  and  $g$  are regarded as  $|\mathcal{X}_I|$ -dimensional vectors and “ $\cdot$ ” is a dot product.
- If  $I \supset J \neq \emptyset$ , then  $\langle f, g \rangle$  is the matrix-vector product  $f \cdot g$ , where  $f$  is regarded as a  $|\mathcal{X}_{I \setminus J}| \times |\mathcal{X}_J|$  matrix,  $g$  is regarded as a  $|\mathcal{X}_J|$ -dimensional vector, and “ $\cdot$ ” is a matrix-vector product.
- If  $I \setminus J$ ,  $J \setminus I$  and  $I \cap J$  are all non-empty, then  $\langle f, g \rangle$  is the matrix-matrix product  $f \cdot g$ , where  $f$  is regarded as a  $|\mathcal{X}_{I \setminus J}| \times |\mathcal{X}_{I \cap J}|$  matrix,  $g$  is regarded as  $|\mathcal{X}_{I \cap J}| \times |\mathcal{X}_{J \setminus I}|$  matrix, and “ $\cdot$ ” is a matrix-matrix product.
- If  $I$  and  $J$  are disjoint and both non-empty, then  $\langle f, g \rangle$  is the vector outer product, matrix Kronecker product, or tensor product,  $f \cdot g$ , where  $f$  and  $g$  are regarded as two vectors, two matrices, or two tensors, respectively, and “ $\cdot$ ” is the corresponding product operation.

In summary, this simple sum-of-products form, namely  $\langle f, g \rangle$ , unifies various notions of “product” in linear algebra. This unification illustrates the convenience of understanding linear algebraic objects such



as vectors, matrices and tensors as multivariate functions, since in this perspective one never needs to be concerned with whether a vector is a row or column vector, whether a matrix is transposed, and so forth.

A general sum-of-products form which involves multiple functions and which can be represented by an NFG may be viewed as a linear algebraic expression involving various such linear algebraic objects and various such notions of product. The fact that the sum-of-products form  $\langle \cdot, \cdot, \dots, \cdot \rangle$  does not depend on how its arguments are ordered contrasts with the standard order-dependent notations in linear algebra.

In this perspective, Lemma 1 follows from the order-independent nature of sum-of-products forms, and Lemma 2 follows from the fact that  $\delta_{=}$  is essentially an identity matrix.

In linear algebra, vectors, matrices and tensors may be viewed alternatively as linear maps, which are characterized by the spaces they act on and the product operation used in defining the maps. Similar perspectives can be made explicit in the NFG context. For example, a complex-valued bivariate function  $f(x, y)$  defined on  $\mathcal{X} \times \mathcal{Y}$  may be viewed as two maps: when participating in the sum-of-products form  $\langle f(x, y), g(y) \rangle$  with a function  $g : \mathcal{Y} \rightarrow \mathbb{C}$ ,  $f$  can be viewed as a linear map from the vector space  $\mathbb{C}^{\mathcal{Y}}$  to the vector space  $\mathbb{C}^{\mathcal{X}}$ ; when participating in the sum-of-products form  $\langle f(x, y), h(x) \rangle$  with a function  $h : \mathcal{X} \rightarrow \mathbb{C}$ ,  $f$  can be viewed as a linear map from the vector space  $\mathbb{C}^{\mathcal{X}}$  to the vector space  $\mathbb{C}^{\mathcal{Y}}$ .

This aspect allows us to interpret dual functions in two different ways. Suppose that  $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$  and  $\hat{\Phi} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{C}$  are a pair of dual functions. On one hand, we may view  $\Phi$  as a map from  $\mathbb{C}^{\mathcal{X}}$  to  $\mathbb{C}^{\mathcal{Y}}$  and  $\hat{\Phi}$  as a map from  $\mathbb{C}^{\mathcal{Y}}$  back to  $\mathbb{C}^{\mathcal{X}}$ . In this view, the composition map  $\hat{\Phi} \circ \Phi$  is the identity map from  $\mathbb{C}^{\mathcal{X}}$  to  $\mathbb{C}^{\mathcal{X}}$ , and  $\hat{\Phi}$  is essentially the inverse or pseudo-inverse of  $\Phi$ . On the other hand, we may view  $\Phi$  as a map from  $\mathbb{C}^{\mathcal{X}}$  to  $\mathbb{C}^{\mathcal{Y}}$  and  $\hat{\Phi}$  also as a map from  $\mathbb{C}^{\mathcal{X}}$  to  $\mathbb{C}^{\mathcal{Y}}$ . In this view, the dot product of two vectors  $f$  and  $g$  in  $\mathbb{C}^{\mathcal{X}}$  is preserved after they are mapped, respectively, to vectors  $\langle f, \Phi \rangle$  and  $\langle g, \hat{\Phi} \rangle$  in  $\mathbb{C}^{\mathcal{Y}}$ . This view is justified by  $\langle f, g \rangle = \langle f, \Phi, \hat{\Phi}, g \rangle = \langle \langle f, \Phi \rangle, \langle \hat{\Phi}, g \rangle \rangle$ .

Finally, we justify the term “transformer” that was introduced in Section 2.2. Suppose that the functions  $\Phi$  and  $\hat{\Phi}$  on  $\mathcal{X} \times \mathcal{Y}$  are dual with respect to alphabet  $\mathcal{Y}$ , and that  $\mathcal{X}$  and  $\mathcal{Y}$  have the same cardinality. Then we may identify  $\Phi$  with its square-matrix representation in which the rows are indexed by  $\mathcal{X}$  and the columns are indexed by  $\mathcal{Y}$ ; similarly, we may identify  $\hat{\Phi}$  with its square-matrix representation in which the rows are indexed by  $\mathcal{Y}$  and the columns are indexed by  $\mathcal{X}$ . The fact that  $\Phi$  and  $\hat{\Phi}$  are dual with respect to  $\mathcal{Y}$  implies that the matrix product  $\Phi \cdot \hat{\Phi}$  is the identity matrix. Thus the matrix  $\hat{\Phi}$  is the unique inverse of the matrix  $\Phi$ , and *vice versa*. Therefore, the functions  $\Phi$  and  $\hat{\Phi}$  may be regarded as a pair of *transformations* (or transformation kernels) that are inverse to each other.

## 2.4 Holographic Transformations and Generalized Holant Theorem

Now we are ready to define holographic transformations.

Suppose that  $I$  is a finite index set and that for each  $i \in I$ , there are two finite alphabets  $\mathcal{X}_i$  and  $\mathcal{Y}_i$  having the same cardinality. We will call a function  $\Phi : \mathcal{X}_I \times \mathcal{Y}_I \rightarrow \mathbb{C}$  a *separable transformation* if  $\Phi$  is a transformation from  $\mathbb{C}^{\mathcal{X}_I}$  to  $\mathbb{C}^{\mathcal{Y}_I}$  (namely, there exists a unique function  $\hat{\Phi} : \mathcal{X}_I \times \mathcal{Y}_I \rightarrow \mathbb{C}$  such that  $\langle \Phi(x, y), \hat{\Phi}(x', y) \rangle = \delta_{=(x, x')}$  for all  $x, x' \in \mathcal{X}_I$ ), and there exists a collection of functions  $\{\Phi_i \in \mathbb{C}^{\mathcal{X}_i \times \mathcal{Y}_i} : i \in I\}$  such that  $\Phi = \prod_{i \in I} \Phi_i$ . Noting that  $\prod_{i \in I} \Phi_i$  may be identified with the sum-of-products form  $\langle \Phi_i : i \in I \rangle$ , we see that transforming any function in  $f \in \mathbb{C}^{\mathcal{X}_I}$  by  $\Phi$  is equivalent to evaluating the sum-of-products form  $\langle f, \langle \Phi_i : i \in I \rangle \rangle$ , which can be performed via *separately* transforming  $f$  by each of the  $\Phi_i$ ’s in an arbitrary

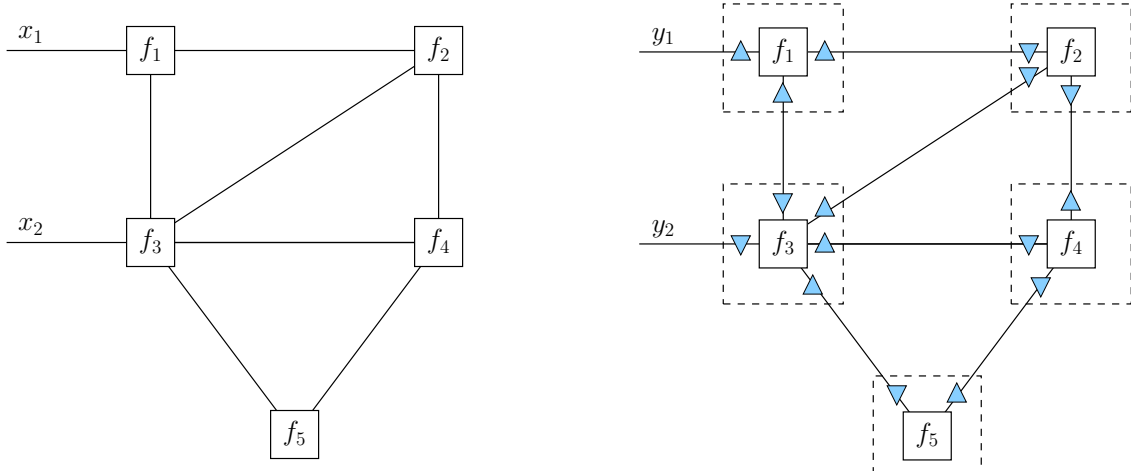


Figure 5: Holographic transformation: an NFG  $\mathcal{G}$  (left) and its holographic transformation  $\mathcal{G}^H$  (right). Each triangular vertex is a transformer, possibly different. Oppositely oriented transformers on an edge are the inverses of each other.

order; hence the term “separable.”

It is easy to verify that if  $\Phi := \prod_{i \in I} \Phi_i$  is a separable transformation, then its inverse transformation  $\widehat{\Phi} : \mathcal{X}_I \times \mathcal{Y}_I \rightarrow \mathbb{C}$  is  $\prod_{i \in I} \widehat{\Phi}_i$  where each  $\widehat{\Phi}_i : \mathcal{X}_i \times \mathcal{Y}_i \rightarrow \mathbb{C}$  is the inverse transformation of  $\Phi_i$ . It follows that if a transformation is separable, then so is its inverse.

**Holographic Transformation** Let  $\mathcal{G} = (V, E^{\text{int}}, E^{\text{ext}}, f_V)$  be an NFG where for each edge  $e \in E := E^{\text{ext}} \cup E^{\text{int}}$ ,  $\mathcal{X}_e$  is the alphabet of the represented variable. For each  $e \in E$ , let  $\mathcal{Y}_e$  be another alphabet having the same cardinality as  $\mathcal{X}_e$ . For every vertex  $v$  and every edge  $e \in E(v)$ , we associate a transformation  $\Phi_{v,e} : \mathcal{X}_e \times \mathcal{Y}_e \rightarrow \mathbb{C}$  such that if  $e$  is a regular edge connecting vertices  $u$  and  $v$ , then  $\Phi_{u,e}$  and  $\Phi_{v,e}$  are the inverse transformations of each other. Let  $\Phi_v := \prod_{e \in E(v)} \Phi_{v,e}$  for every vertex  $v$ . Locally transform each function  $f_v$  to the function  $F_v \in \mathbb{C}^{\mathcal{Y}_{E(v)}}$  via  $F_v := \langle f_v, \Phi_v \rangle$ , and collectively denote  $\{F_v : v \in V\}$  by  $F_V$ . We call the NFG  $\mathcal{G}^H := (V, E^{\text{int}}, E^{\text{ext}}, F_V)$  the holographic transformation of  $\mathcal{G}$  with respect to the collection of local separable transformations  $\{\Phi_v : v \in V\}$ .

A graphical example of holographic transformation is shown in Figure 5. We note that holographic transformations keep the topology of the NFG unchanged, and only transform each local function.

**Theorem 1 (Generalized Holant Theorem)** *In the setting above, the exterior function  $Z_{\mathcal{G}^H}$  of the NFG  $\mathcal{G}^H$  is related to the exterior function  $Z_{\mathcal{G}}$  of the original NFG  $\mathcal{G}$  by*

$$Z_{\mathcal{G}^H}(y_{E^{\text{ext}}}) = \langle Z_{\mathcal{G}}(x_{E^{\text{ext}}}), \langle \Phi_e(x_e, y_e) : e \in E^{\text{ext}} \rangle \rangle,$$

where for each  $e \in E^{\text{ext}}$ , we have written  $\Phi_e$  in place of  $\Phi_{v,e}$ .

*Proof:* The theorem simply follows from Lemma 3. Graphically, as shown in Figure 5, the holographic transformation is equivalent to first inserting into each edge an inverse pair of transformers and into each

dangling edge a transformer, and then transforming each local function by its surrounding transformers. Since each inverse pair of transformers cancels out, the only difference between the exterior function of  $\mathcal{G}$  and that of  $\mathcal{G}^H$  is due to the transformers that have been inserted in the dangling edges. This establishes the theorem.  $\square$

When  $E^{\text{ext}}$  is the empty set, the exterior functions of  $\mathcal{G}$  and  $\mathcal{G}^H$  reduce to scalars. In this case, the generalized Holant theorem reduces to the Holant theorem of [11].

We note also that in the literature of “loop calculus” [18, 19, 20, 21], which has been introduced for the study of belief propagation and of the partition functions of statistical mechanics models, a result equivalent to the Holant theorem has been proved, and a transformation equivalent to our holographic transformation (on NFGs without dangling edges) has been proposed under the name of “gauge transformation” (see, *e.g.*, [21]).

Although our generalization of the Holant theorem appears straightforward, we believe that there is a conceptual leap in this generalization. In particular, the original Holant theorem reveals only that there are redundant and structurally identical NFGs that may be used to represent the same scalar quantity as a sum of products; it makes no attempt to transform or reparameterize an exterior function that assumes a more general form. In the general setting of holographic transformations, when the original NFG  $\mathcal{G}$  is viewed as a realization of some function  $g := Z_{\mathcal{G}}$  on a collection of alphabets  $\{\mathcal{X}_e : e \in E\}$ , the holographically transformed NFG  $\mathcal{G}^H$  is viewed as a realization of a related function  $g^H := Z_{\mathcal{G}^H}$  on a different collection of alphabets  $\{\mathcal{Y}_e : e \in E\}$ . In particular, the function  $g^H$  may be regarded as a transform-domain representation of  $g$  via an “external change of basis,” namely, a change of basis for the vector space  $\mathbb{C}^{|\mathcal{X}_{E^{\text{ext}}}|}$ , that is characterized by the “external transformation”  $\langle \Phi_e : e \in E^{\text{ext}} \rangle$ , where this “external change of basis” involves, in its sum-of-products form, a “local change of basis” for each component vector space  $\mathbb{C}^{|\mathcal{X}_e|}$ ,  $e \in E$ .

### 3 Applications of Holographic Transformations

#### 3.1 Duality in Normal Factor Graphs

The first duality theorem for codes on graphs was the normal graph duality theorem that was introduced by Forney in [1]. In the setting of [1], the graphs considered, rather than being NFGs, are “normal graphs,” where edges incident on one or two vertices represent “symbol” variables and “state” variables, respectively, and where each vertex represents a local group-code constraint. The global behavior represented by the graph is the set of all symbol-state configurations that satisfy all local constraints, and the graph itself represents a group code that consists of all symbol configurations that participate in at least one symbol-state configuration in the global behavior. In [1], Forney introduced a local “dualization” procedure for normal graphs, which converts each local code constraint to its dual code constraint and inserts a “sign inverter” into every edge connecting two vertices. The normal graph duality theorem then states that the dualized graph represents the dual code.

The normal graph duality theorem of [1] may be formulated as an equivalent theorem, which we call the “code normal factor graph duality theorem,” using the language of normal factor graphs. More specifically, we may use an NFG to represent a state realization of a code  $C$ , where each vertex represents the indicator

function of a local code constraint, and the exterior function is, up to scale,<sup>8</sup> the indicator function of the code  $C$ . The dualization procedure may be reformulated on the NFG as converting the indicator function of each local code to the indicator function of the dual of the local code and inserting an indicator function  $\delta_+$  into each edge, where  $\delta_+$  evaluates to 1 if and only if the two arguments of the function are additive inverses of each other. Then the code normal factor graph duality theorem states that the exterior function realized by the dual NFG is up to scale the indicator function of the dual code  $C^\perp$ .

In the framework of factor graphs, Mao and Kschischang [24] introduced the notions of multivariate convolution and convolutional factor graphs, and proved a duality theorem between a multiplicative factor graph and its dual convolutional factor graph. The duality theorem of [24] (Theorem 11), which we call the “MK theorem,” states that a dual pair of factor graphs represent a Fourier transform pair. Since the indicator function of a code and that of its dual code are a Fourier transform pair up to scale, the code normal graph duality theorem and hence the normal graph duality theorem follow from the MK theorem as corollaries.

In a concurrent development [17], Forney has established a general normal factor graph duality theorem, where the vertices of an NFG can represent arbitrary functions and the dualization procedure is defined as converting each local function to its Fourier transform and inserting a  $\delta_+$  function into each edge. The general normal factor graph duality theorem states that a dual pair of NFG’s represent a Fourier transform pair up to scale. This theorem reduces to the code normal factor graph duality theorem (and hence to the normal graph duality theorem) if each graph vertex is the indicator function of a local code.

In this subsection, we will show that the general normal factor graph duality theorem follows directly from the generalized Holant theorem.

Let  $\mathcal{G} = (V, E^{\text{int}}, E^{\text{ext}}, f_V)$  be an arbitrary NFG, where each variable alphabet  $\mathcal{X}_e, e \in E = E^{\text{ext}} \cup E^{\text{int}}$ , is a finite abelian group written additively. It is well-known that every finite abelian group  $(\mathcal{X}, +)$  has a character group  $\mathcal{X}^\wedge$ , consisting of precisely the set of all homomorphisms, called characters, of  $\mathcal{X}$  mapping  $\mathcal{X}$  into the multiplicative group of the unit circle in the complex plane. The character group  $\mathcal{X}^\wedge$  of  $\mathcal{X}$  has the following properties [25].

- The group operation  $+$  in  $\mathcal{X}^\wedge$  is defined by  $(\hat{x}_1 + \hat{x}_2)(x) = \hat{x}_1(x)\hat{x}_2(x)$  for any two characters  $\hat{x}_1, \hat{x}_2 \in \mathcal{X}^\wedge$  and any  $x \in \mathcal{X}$ .
- $(\mathcal{X}^\wedge)^\wedge$  is isomorphic to  $\mathcal{X}$ . This result, known as Pontryagin duality [25], allows each element of  $\mathcal{X}$  to be treated as a character of  $\mathcal{X}^\wedge$ .
- $\mathcal{X}^\wedge$  is isomorphic to  $\mathcal{X}$ .
- For each  $x \in \mathcal{X}$  and  $\hat{x} \in \mathcal{X}^\wedge$ ,  $x(\hat{x}) = \hat{x}(x)$ . We will denote<sup>9</sup> both  $x(\hat{x})$  and  $\hat{x}(x)$  by  $\kappa_{\mathcal{X}}(x, \hat{x})$  and, for later use, denote  $\kappa_{\mathcal{X}}(x, -\hat{x})/|\mathcal{X}|$  by  $\hat{\kappa}_{\mathcal{X}}(x, \hat{x})$ . Keeping in mind that  $\kappa_{\mathcal{X}}$  and  $\hat{\kappa}_{\mathcal{X}}$  are both defined with respect to the alphabet  $\mathcal{X}$ , we may sometimes suppress such dependency in our notation. It is easy to see that  $\kappa_{\mathcal{X}}$  and  $\hat{\kappa}_{\mathcal{X}}$  are a dual pair of functions (with respect to either  $\mathcal{X}$  or  $\mathcal{X}^\wedge$ ). Since  $\mathcal{X}$  and  $\mathcal{X}^\wedge$

---

<sup>8</sup>The scaling constant is the number of symbol-state configurations in the global behavior that correspond to each codeword; this number is the same for every codeword since the global behavior is an abelian group.

<sup>9</sup>It is customary in the literature to denote both  $x(\hat{x})$  and  $\hat{x}(x)$  by the pairing  $\langle x, \hat{x} \rangle$ . But we choose not to use this notation since it collides with our notation for “sum-of-products” forms.

have the same size, they in fact define a pair of transformations, namely, the Fourier transform and its inverse, as we state next.

- For any function  $f \in \mathbb{C}^{\mathcal{X}}$ , its Fourier transform  $\mathcal{F}[f]$  is a complex-valued function on  $\mathcal{X}^\wedge$  defined by  $\mathcal{F}[f] := \langle f, \kappa \rangle$ . It follows that for any function  $f \in \mathbb{C}^{\mathcal{X}^\wedge}$ , its inverse Fourier transform  $\mathcal{F}^{-1}[f]$  is a complex-valued function on  $\mathcal{X}$ ,  $\mathcal{F}^{-1}[f] = \langle f, \hat{\kappa} \rangle$ . We note that the inverse Fourier transform operator  $\mathcal{F}^{-1}$  may also be applied to a function  $f \in \mathbb{C}^{\mathcal{X}}$  and result in a function on  $\mathcal{X}^\wedge$ , where in the sum-of-products form the summation is over the  $\mathcal{X}$ -valued variable.
- If  $\mathcal{X}$  is the direct product  $\mathcal{X}_1 \times \mathcal{X}_2$  of finite abelian groups  $\mathcal{X}_1$  and  $\mathcal{X}_2$ , then  $\mathcal{X}^\wedge$  is the direct product  $\mathcal{X}_1^\wedge \times \mathcal{X}_2^\wedge$  of the character groups  $\mathcal{X}_1^\wedge$  and  $\mathcal{X}_2^\wedge$ . In this case, for any  $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$  and any  $(\hat{x}_1, \hat{x}_2) \in \mathcal{X}_1^\wedge \times \mathcal{X}_2^\wedge$ ,

$$\kappa((x_1, x_2), (\hat{x}_1, \hat{x}_2)) = \kappa(x_1, \hat{x}_1) \kappa(x_2, \hat{x}_2). \quad (2)$$

Equation (2) shows that the Fourier transform is a separable transformation.

Returning to the NFG  $\mathcal{G}$ , we next obtain its “dual” by applying Forney’s dualization procedure.

**NFG Dualization Procedure** Replace each  $\mathcal{X}_e$ -valued variable  $x_e$  by an  $\mathcal{X}_e^\wedge$ -valued the variable  $\hat{x}_e$ . Replace each function  $f_v, v \in V$ , by its Fourier transform  $\mathcal{F}[f_v]$  and insert into each internal edge a vertex representing the function  $\delta_+(\cdot)$ . Again, the function  $\delta_+$  is an indicator function which evaluates to 1 if and only if its two variables are the additive inverses of each other. We will denote the resulting NFG by  $\hat{\mathcal{G}}$ , and refer to it as the *dual* NFG of  $\mathcal{G}$ .

**Theorem 2 (General Normal Factor Graph Duality Theorem)** *The exterior function  $Z_{\hat{\mathcal{G}}}$  realized by the dual NFG  $\hat{\mathcal{G}}$  and the exterior function  $Z_{\mathcal{G}}$  realized by the original NFG  $\mathcal{G}$  are related by*

$$Z_{\hat{\mathcal{G}}} = |\mathcal{X}_{E^{\text{int}}}| \cdot \mathcal{F}[Z_{\mathcal{G}}]. \quad (3)$$

*Proof:* This theorem can be viewed as a corollary of the generalized Holant theorem, and can be simply proved graphically (Figure 6): Construct another NFG  $\mathcal{G}'$  by inserting a  $\delta_-$  function into each regular edge  $\mathcal{G}$ ; by Lemma 2,  $Z_{\mathcal{G}'} = Z_{\mathcal{G}}$ . Obtain the NFG  $\mathcal{G}'^H$  from  $\mathcal{G}'$  by inverse Fourier transforming every  $\delta_-$  in  $\mathcal{G}'$  and Fourier transforming every other function. This corresponds to inserting the dual functions  $\kappa_{\mathcal{X}_e}$  and  $\hat{\kappa}_{\mathcal{X}_e}$  into each regular edge  $e$  (with  $\hat{\kappa}_{\mathcal{X}_e}$  adjacent to the function  $\delta_-$ ) and inserting the function  $\kappa_{\mathcal{X}_e}$  into each dangling edge  $e$ . Since the inserted transformers in each regular edge are the inverses of each other, this verifies that  $\mathcal{G}'^H$  is a holographic transformation of  $\mathcal{G}'$ . By the generalized Holant theorem,

$$Z_{\mathcal{G}'^H} = \langle Z_{\mathcal{G}'}, \langle \kappa_e : e \in E^{\text{ext}} \rangle \rangle = \langle Z_{\mathcal{G}}, \langle \kappa_e : e \in E^{\text{ext}} \rangle \rangle = \mathcal{F}[Z_{\mathcal{G}}].$$

Invoking a well-known result that for  $\delta_-$  defined on  $\mathcal{X} \times \mathcal{X}$ ,  $\mathcal{F}^{-1}[\delta_-] = \frac{1}{|\mathcal{X}|} \delta_+$ , we see that  $\mathcal{G}'^H$  and  $\hat{\mathcal{G}}$  are in fact identical except that in  $\mathcal{G}'^H$ , each  $\delta_+$  inserted in edge  $e$  is scaled by  $\frac{1}{|\mathcal{X}_e|}$ . The theorem is then proved by collecting all the scaling factors.  $\square$

We note that Theorem 2 is the most general NFG duality theorem. If each function in an NFG is an indicator function of a local code, then the exterior function realized by the NFG is up to scale the indicator

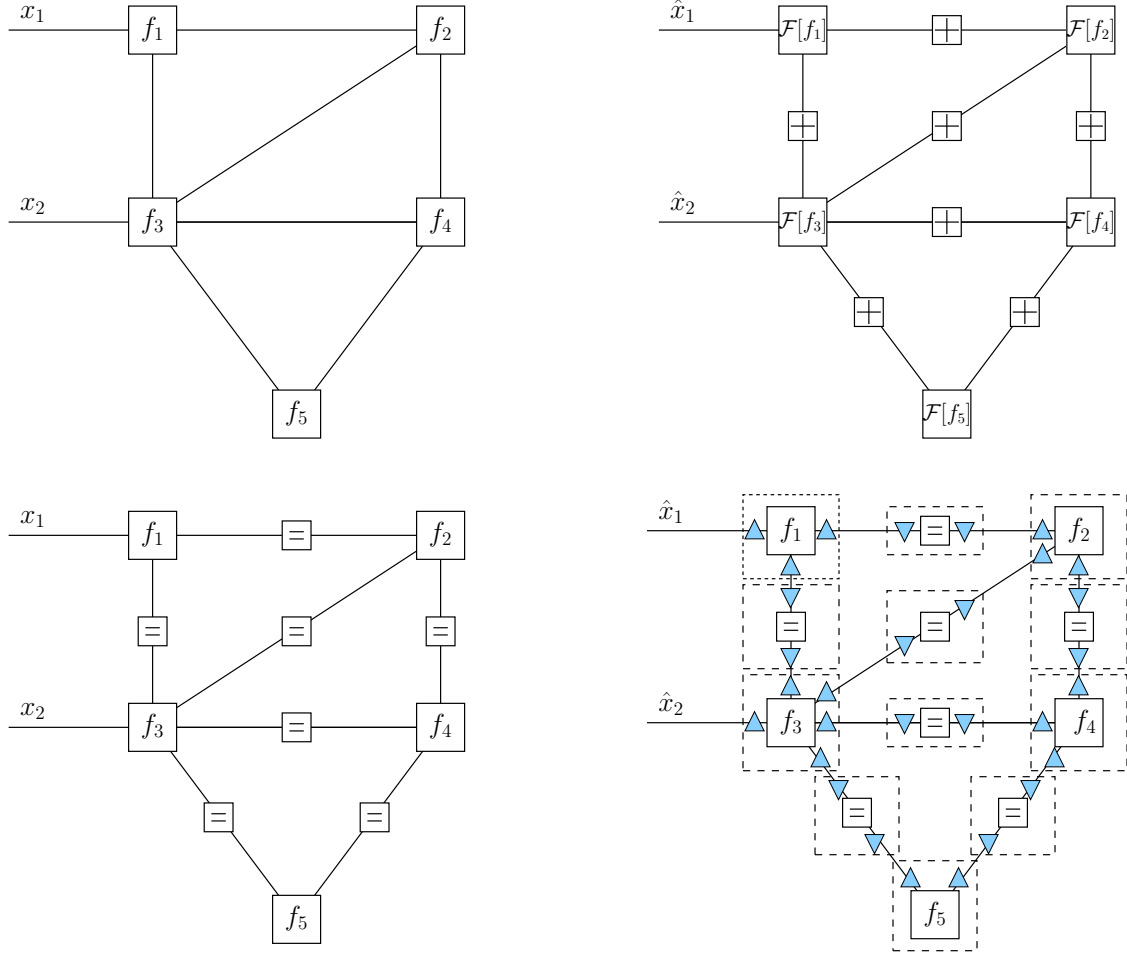


Figure 6: NFG's  $\mathcal{G}$  (top left),  $\hat{\mathcal{G}}$  (top right),  $\mathcal{G}'$  (bottom left) and  $\mathcal{G}'^H$  (bottom right).

function of a group code. Such an NFG, which may be called a “code normal factor graph” (“code NFG”) may then be used to represent the group code. This makes a code NFG equivalent to a normal graph. Since the indicator function of a code and that of its dual are (up to scale) a Fourier transform pair, the general normal factor graph duality theorem then reduces to the code normal graph duality theorem (and the normal graph duality theorem), which state that a dual pair of code NFGs (resp. a dual pair of normal graphs) represent a pair of dual codes.

It is worth noting that the general normal factor graph duality theorem also follows from the MK theorem, via the application of the “projection-slice theorem” of the Fourier transform, or the “sampling/averaging duality theorem” of [24, Theorem 8]. However, the proof using the generalized Holant theorem seems more transparent.

### 3.2 Holographic Reduction

A holographic reduction may be regarded as a particular kind of holographic transformation applied to an NFG without dangling edges, by which a counting problem may be reduced to an equivalent “PerfMatch problem”. Using this technique, Valiant constructed polynomial-time solvers for various families of “counting” problems previously unknown to be in P [11]. We now summarize the main results of Valiant [11] and explain how holographic reduction works.

**The PerfMatch Problem** Suppose that  $H = (V, E, w)$  is a weighted graph with vertex set  $V$ , edge set  $E$ , and weighting function  $w$  which assigns to each edge  $e \in E$  a complex weight  $w(e)$ . The quantity PerfMatch  $\pi(H)$  of  $H$  is defined as

$$\pi(H) := \sum_{M \in Q(H)} \prod_{e \in M} w(e),$$

where  $Q(H)$  is the collection of all perfect matchings<sup>10</sup> of  $H$ . It is known that the PerfMatch problem, namely, solving for  $\pi(H)$ , can be performed in polynomial time using the FKT algorithm [26, 27] if  $H$  is a *planar* graph.

A principle underlying holographic reduction is a graph-theoretic property which expresses the PerfMatch of a weighted graph as a sum-of-products form, in which each involved function is defined based on a local component of the graph. Such a local component is referred to as a “matchgate,” and each involved function is referred to as the “signature” of such a matchgate. More precisely, a matchgate is a weighted graph  $H$  (which will be used as a local component of a larger graph) with a subset  $W$  of its vertices specified as its “external vertices” (which will be used to connect to other matchgates to form a larger graph). Suppose that  $(H_1, W_1), (H_2, W_2), \dots, (H_m, W_m)$  are a collection of matchgates. We may build a larger graph  $H$  by connecting these matchgates via their external vertices where the only restriction is that each external vertex of a matchgate  $(H_i, W_i)$  connects to exactly one external vertex of a different matchgate  $(H_j, W_j)$ . The edges that connect the matchgates will be assigned weight 1. Figure 7 (a) shows two kinds of matchgates, which are used to construct a larger weighted graph as shown in Figure 7 (c) in such a way.

---

<sup>10</sup>In graph theory, a perfect matching of a graph is a set of non-adjacent edges such that every vertex of the graph is the endpoint of an edge in the set.

Now let  $\mathcal{E}$  denote the set of edges in the larger graph  $H$  that connect (the external vertices of) the matchgates, and let  $\mathcal{E}(i)$  denote the subset of the edges in  $\mathcal{E}$  incident to the external vertices of the matchgate  $(H_i, W_i)$ . Associate with each  $e \in \mathcal{E}$  a  $\{0, 1\}$ -valued variable  $x_e$ . The signature  $\mu_i$  of the matchgate  $(H_i, W_i)$  is a function of the variable set  $x_{\mathcal{E}(i)}$  defined as follows: Every configuration  $x_{\mathcal{E}(i)}$  is made to correspond to a subgraph of  $H_i$  induced by deleting a subset of its external vertices; more precisely, an external vertex is deleted if and only if it is the endpoint of an edge  $e \in \mathcal{E}(i)$  whose  $x_e = 1$  in the configuration  $x_{\mathcal{E}(i)}$ ; the PerfMatch of the subgraph induced this way is then defined to be the value  $\mu_i(x_{\mathcal{E}(i)})$ . Then it is possible to express the PerfMatch of the larger graph  $H$  as a sum-of-products form involving the signatures of the matchgates as follows.

$$\pi(H) = \sum_{x_{\mathcal{E}}} \prod_{i=1}^m \mu_i(x_{\mathcal{E}(i)}). \quad (4)$$

It is easy to verify that the sum-of-products form in (4) has an NFG representation, since each variable  $x_e, e \in \mathcal{E}$ , is involved in precisely two functions (noting that every edge  $e \in \mathcal{E}$  connects two matchgates). In this case, the NFG has no dangling edges, and the realized exterior function reduces to a scalar, i.e., the PerfMatch of the larger graph  $H$ .

Figure 7 shows an example of how to build an NFG that realizes the PerfMatch of a graph using the signatures of its matchgates. In the figure, (a) shows two matchgates, where the signature of each matchgate by itself can be viewed as an NFG vertex in (b). When we use the matchgates in (a) to build the larger graph  $H$  in (c), then equality (4) suggests that the PerfMatch of  $H$  is realized by the NFG in (d). That is, the NFG topology is identical to the topology by which the matchgates form the larger graph  $H$ . Visually, the relationship between the NFG and the graph  $H$  is apparent: The picture in Figure 7(d) is the NFG if we ignore the details inside the boxes, and is the graph  $H$  if we ignore the boxes.

**Solving Counting Problems via Holographic Reduction** Many counting problems are described in terms of a large collection of variables and a large collection of constraints each involving a subset of the variables. The objective of such problems is to compute the total number of global variable configurations satisfying all the constraints. In this context, the idea of holographic reduction is to transform the problem of interest to a PerfMatch problem. We now outline this approach.

1. Express the problem as the computation of the exterior function realized by a *planar* NFG  $\mathcal{G}$  without dangling edges. When this is possible, each vertex of  $\mathcal{G}$  represents an indicator (i.e.,  $\{0, 1\}$ -valued) function.
2. For each variable  $x_e$  in  $\mathcal{G}$ , find a pair of inverse transformations  $\Phi_e$  and  $\widehat{\Phi}_e$ , and construct a holographic transformation  $\mathcal{G}^H$  of  $\mathcal{G}$  such that each function vertex in  $\mathcal{G}^H$  represents the signature of a matchgate.
3. Create a weighted graph  $H$  by replacing each vertex of  $\mathcal{G}^H$  with the corresponding matchgate drawing, and assign weight one to each edge of  $\mathcal{G}^H$ . This process essentially turns the holographically transformed NFG as in Figure 7(d) into a weighted graph as in Figure 7(c).

By the Holant theorem, the exterior function realized by  $\mathcal{G}$  is the same as that realized by  $\mathcal{G}^H$ . Since  $\mathcal{G}$  and  $\mathcal{G}^H$  do not have dangling edges, the realized exterior function is in fact a scalar; by (4), this scalar is the PerfMatch of  $H$ . It is easy to verify that  $\mathcal{G}$  being a planar graph implies that  $H$  is a planar graph



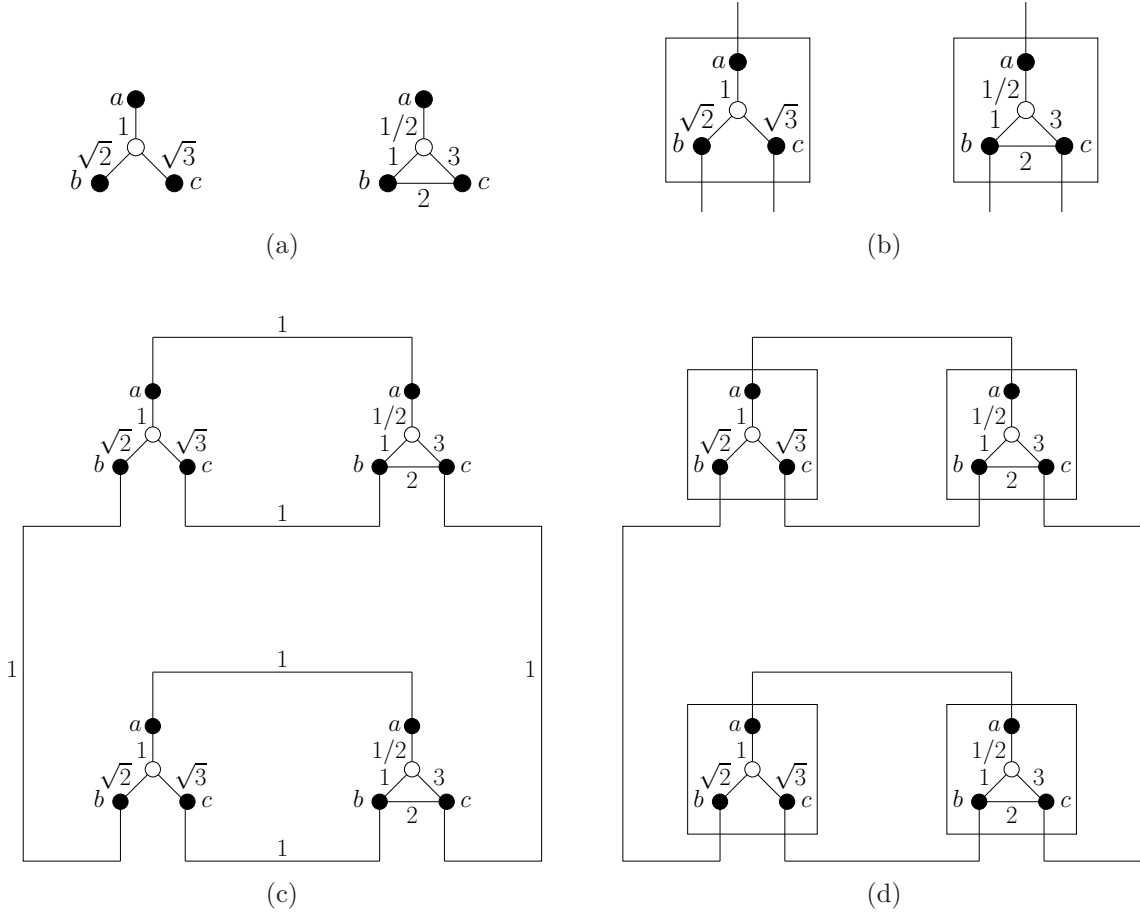


Figure 7: (a) Two matchgates, where solid circles represent the external vertices; (b) the signatures of the matchgates represented as NFG function vertices (the boxes); (c) a larger graph  $H$  constructed from the matchgates; (d) the NFG realization of the PerfMatch of  $H$ , where each box is a function vertex.

(provided that each matchgate is also a planar graph). Thus solving the PerfMatch problem for  $H$  via the FKT algorithm solves the original counting problem in polynomial time.

In [11], finding the “right” transformations  $\{\Phi_e : e \in E\}$  that transform each local function in the original NFG to the signature of some matchgate appeared to be an art. Later Cai and Lu [13] presented a systematic approach to determine whether such a transformation exists. Remarkably, Cai *et al.* [14] have extended the approach of holographic reduction beyond transformations to the PerfMatch problem by introducing the concept of “Fibonacci gates.”

## 4 Concluding Remarks

Sums of products are fundamental in physics, computer science, coding theory, information theory, and indeed all of science and applied mathematics. In this paper, we have introduced what we call the “exterior-function semantics” for normal factor graphs, which establishes a one-to-one correspondence between sum-of-products forms satisfying certain nonrestrictive “normal” constraints and their associated normal factor graphs. Within this framework, we have introduced a very general notion of holographic transformations of normal factor graphs, and have stated and proved a general and powerful theorem (which we call the generalized Holant theorem) that relates the exterior function of a normal factor graph to that of its holographic transformation. As corollaries of this theorem, we obtain Valiant’s original Holant theorem, as well as a very simple proof of a general normal factor graph duality theorem, of which Forney’s original normal graph duality theorem is a special case. This connection between two seemingly distant fields seems to us remarkable.

Although the use of internal variables in graphical models is by no means new, the exterior-function semantics introduced in this paper seems to us elegant, intuitive, and potentially of wide application. The linear algebraic perspectives that we have mentioned briefly in this paper may have much more general use. Indeed, as Pascal Vontobel has observed [28], the use of “trace diagrams” in mathematical physics (see, e.g., [29, 30]) appears to have much in common with our graphical techniques. We suspect that the areas of potential applications are vast.

## Acknowledgment

We would like to thank Frank Kschischang for introducing to us holographic algorithms and for earlier discussions on related subjects. We also want to thank the anonymous reviewers for their helpful suggestions. We are particularly indebted to Pascal Vontobel and David Forney for their extensive and very detailed comments on previous drafts of this paper, which have helped significantly to improve the presentation of this paper.

## Appendix: Converting Arbitrary Sum-of-Products Forms to NFGs

In the framework of factor graphs [7], the product of any collection of multivariate functions may be represented by a factor graph. By specifying a subset of the variables in the factor graph to be “internal” (namely,

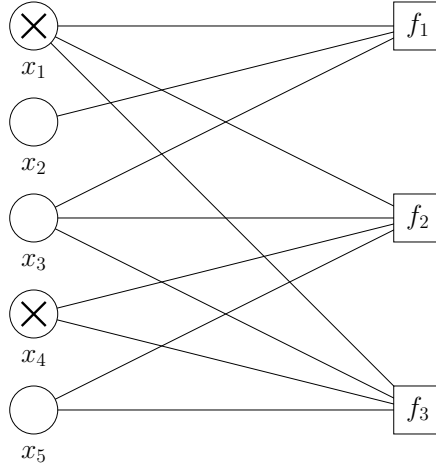


Figure 8: The “marked” factor graph representing the sum-of-products form  $\sum_{x_1, x_4} f_1(x_1, x_2, x_3) f_2(x_1, x_3, x_4, x_5) f_3(x_1, x_3, x_4, x_5)$ .

to be summed over), it is then possible to represent *any* sum-of-products form using a factor graph with additional marks on some variable vertices.

Figure 8 is an example of a sum-of-products form represented by such a “marked” factor graph, where the variable vertices marked with “ $\times$ ” represent internal variables; the variable vertices without such marks are external variables, namely, those remaining in the argument of the represented function. Such a “marked” factor graph then represents the product of all local functions with all internal variables summed over; the function resulting from the summation then clearly involves only the external variables, analogous to the exterior function of an NFG.

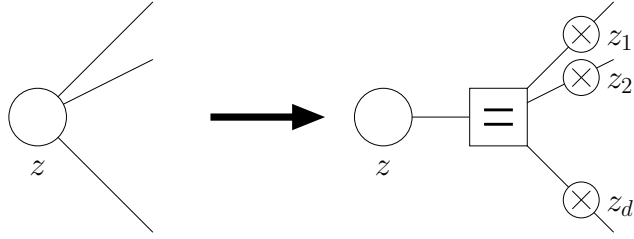
Since a factor graph can have an unrestricted topology and an arbitrary subset of its variable vertices may be marked, it is possible to represent any sum-of-products form using a “marked” factor graph.

Without loss of generality, we will assume that there are no degree-1 internal variable vertices in a “marked” factor graph, since otherwise it is always possible to modify the local function connecting to the variable by summing over that variable.

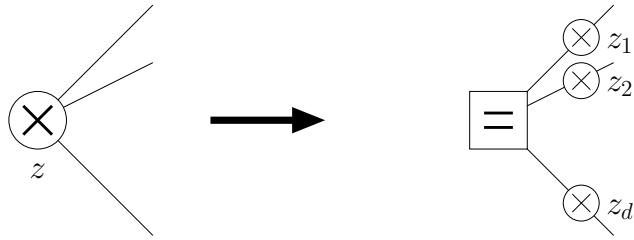
The following procedure, operating on a “marked” factor graph representations, “normalizes” any sum-of-products form.

**Variable Replication Procedure.** Suppose that a variable  $z$  in a factor graph has degree  $d$ . Then we may create  $d$  replicas  $\{z_1, z_2, \dots, z_d\}$  of variable  $z$ , isolate  $z$  from its edges, and attach each of the  $d$  replicas to one of these edges. Remove  $z$  if  $z$  is an internal variable in the original factor graph. Connect all the replicas of  $z$  and  $z$  itself, if it is kept, to a new function vertex representing  $\delta_=(\cdot)$ . Finally, mark all replicas of  $z$  internal (i.e., with “ $\times$ ”). Figure 9 is a graphical illustration of this procedure.

A procedure similar to the Vertex Replication Procedure above was first presented in [1]. It is easy to verify that when applying the Variable Replication Procedure to any variable vertex, the sum-of-products



(a) Replicating an external variable



(b) Replicating an internal variable

Figure 9: Variable Replication Procedure.

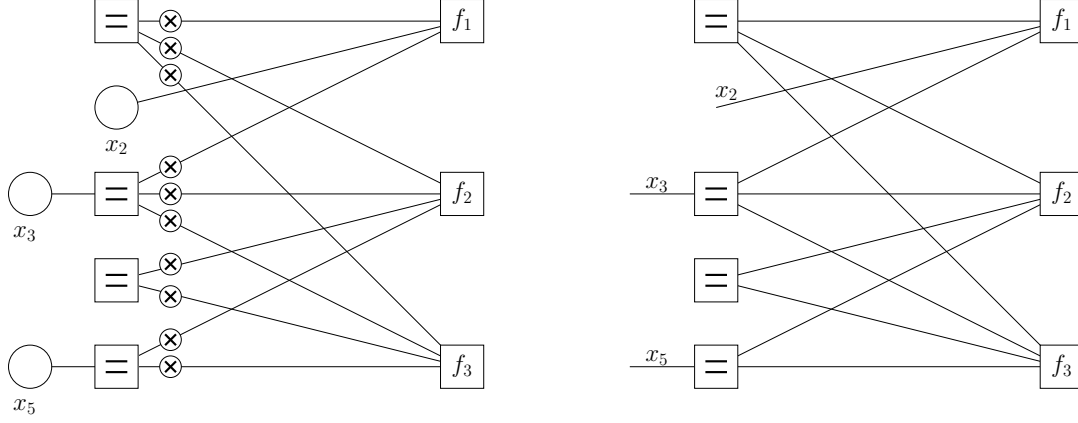


Figure 10: The sum-of-products form resulting from normalizing the “marked” factor graph of Figure 8. Left: the sum-of-products form represented as a “marked” factor graph; right: the sum-of-product form represented as an NFG.

form corresponding to the resulting “marked” factor graph expresses the same function as the original sum-of-products form does.

On a “marked” factor graph, we may apply this procedure to every variable that does not satisfy the “normal” degree restriction (namely that an internal variable vertex have degree two and an external variable vertex have degree one). It is straightforward to verify that in the resulting “marked” factor graph, the normal degree restriction is necessarily satisfied by all variables. We can then represent the resulting sum-of-products form using the NFG notation, representing internal variables as edges and external variables as dangling edges.

Figure 10 shows the sum-of-products form resulting from normalizing the “marked” factor graph in Figure 8.

## References

- [1] G. D. Forney, Jr., “Codes on graphs: Normal realizations,” *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 520–548, 2001.
- [2] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533 – 547, 1981.
- [3] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Univ. Linköping, Linköping, Sweden, 1996.
- [4] N. Wiberg, H.-A. Loeliger, and R. Kötter, “Codes and iterative decoding on general graphs,” *Euro. Trans. Telecomm.*, vol. 6, pp. 513–525, Sept./Oct. 1995.
- [5] H.-A. Loeliger, “An introduction to factor graphs,” *IEEE Sig. Proc. Mag.*, vol. 21, no. 1, pp. 24–41, 2004.

- [6] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.
- [7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [8] R. Koetter, “On the representation of codes in Forney graphs,” in *Codes, Graphs, and Systems*, R. E. Blahut and R. Koetter, Eds. Kluwer Academic Publishers, Feb. 2002, pp. 425–450.
- [9] R. Koetter and A. Vardy, “Factor graphs: Constructions, classification, and bounds,” in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998.
- [10] R. Koetter, M. Effros, T. Ho, and M. Médard, “Network codes as codes on graphs,” in *Proc. 38th Annual Conf. on Information Sciences and Systems*, Princeton, NJ, USA, March 2004, pp. 1–6.
- [11] L. G. Valiant, “Holographic algorithms (extended abstract),” in *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science*, Rome, Italy, Oct. 2004, pp. 306–315.
- [12] M. Schwartz and J. Bruck, “Constrained codes as networks of relations,” *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 2179–2195, 2008.
- [13] J. Y. Cai and P. Lu, “Holographic algorithms: From art to science,” in *Proc. of the 39th Annual ACM Symp. on Theory of Computing*, San Diego, California, June 2007, pp. 401–410.
- [14] J. Y. Cai, P. Lu, and M. Xia, “Holographic algorithms by Fibonacci gates and holographic reductions for hardness,” in *IEEE 49th Annual IEEE Symp. on Foundations of Computer Science*, Philadelphia, PA, Oct. 2008, pp. 644–653.
- [15] L. G. Valiant, “Some observations on holographic algorithms,” To appear in *Proc. 9th Latin American Theoretical Informatics Symp.*, 2010, available at: <http://people.seas.harvard.edu/~valiant/>.
- [16] J. Cai, P. Lu, and M. Xia, “Holographic algorithms with matchgates capture precisely tractable planar #CSP,” *CoRR*, vol. abs/1008.0683, 2010.
- [17] G. D. Forney, Jr., “Codes on graphs: Duality and MacWilliams identities,” Submitted to *IEEE Trans. Inform. Theory*, 2009, available at arXiv:0911.5508.
- [18] M. Chertkov and V. Y. Chernyak, “Loop series for discrete statistical models on graphs,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, p. P06009, 2006.
- [19] V. Y. Chernyak and M. Chertkov, “Loop calculus and belief propagation for q-ary alphabet: Loop tower,” in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, June 2007, pp. 316–320.
- [20] M. Chertkov and V. Y. Chernyak, “Loop calculus in statistical physics and information science,” *Physical Review E*, vol. 73, no. 6, 2006.
- [21] V. Y. Chernyak and M. Chertkov, “Planar graphical models which are easy,” 2009, available at arXiv:0902.0320v4.

- [22] P. O. Vontobel and H.-A. Loeliger, “On factor graphs and electrical networks,” *Mathematical Systems Theory in Biology, Communication, Computation, and Finance*, vol. J. Rosenthal and D. S. Gilliam, eds., IMA Volumes in Math. and Appl., Springer Verlag, 2003.
- [23] P. O. Vontobel, “Kalman filters, factor graphs, and electrical networks,” *Post-Diploma Project, ETH Zurich*, 2002.
- [24] Y. Mao and F. R. Kschischang, “On factor graphs and the Fourier transform,” *IEEE Trans. Inform. Theory*, vol. 51, no. 5, pp. 1635–1649, 2005.
- [25] G. D. Forney, Jr., “Transforms and groups,” in *Codes, Curves and Signals: Common Threads in Communications*, A. Vardy, Ed. Norwood, MA: Kluwer, 1998, ch. 7.
- [26] P. W. Kasteleyn, “The statistics of dimers on a lattice,” *Physica*, vol. 27, no. 12, pp. 1209–1225, 1961.
- [27] H. N. V. Temperley and M. E. Fisher, “Dimer problem in statistical mechanics— An exact result,” *Philosophical Magazine*, vol. 6, no. 68, pp. 1061–1063, 1961.
- [28] G. D. Forney, Jr. and P. O. Vontobel, “Private communications,” 2010.
- [29] S. Morse and E. Peterson, “Trace diagrams, matrix minors, and determinant identities,” 2009, available at arXiv: 0903.1373v2.
- [30] E. Peterson, “On a diagrammatic proof of the Cayley-Hamilton theorem,” 2009, available at arXiv:0907.2364v1.