Electronic Colloquium on Computational Complexity, Report No. 12 (2009)

Balanced Hashing, Color Coding and Approximate Counting

(Extended Abstract)

Noga Alon [*] Shai Gutner [†]

Abstract

Color Coding is an algorithmic technique for deciding efficiently if a given input graph contains a path of a given length (or another small subgraph of constant tree-width). Applications of the method in computational biology motivate the study of similar algorithms for counting the number of copies of a given subgraph. While it is unlikely that exact counting of this type can be performed efficiently, as the problem is $\#W[1]$-complete even for paths, approximate counting is possible, and leads to the investigation of an intriguing variant of families of perfect hash functions. A family of functions from $[n]$ to $[k]$ is an (ϵ, k)-balanced family of hash functions, if there exists a positive T so that for every $K \subset [n]$ of size $|K| = k$, the number of functions in the family that are one-to-one on K is between $(1-\epsilon)T$ and $(1+\epsilon)T$. The family is perfectly k-balanced if it is $(0, k)$-balanced.

We show that every such perfectly k-balanced family is of size at least $c(k)n^{\lfloor k/2 \rfloor}$, and that for every $\epsilon > \frac{1}{poly(k)}$ there are explicit constructions of (ϵ, k)-balanced families of hash functions from $[n]$ to $[k]$ of size $e^{(1+o(1))k} \log n$. This is tight up to the $o(1)$-term in the exponent, and supplies deterministic polynomial time algorithms for approximately counting the number of paths or cycles of a specified length k (or copies of any graph H with k vertices and bounded tree-width) in a given input graph of size n, up to relative error ϵ, for all $k \leq O(\log n)$.

Keywords: Approximate counting of subgraphs, color-coding, derandomization, expanders, perfect hashing, k-wise independence.

[*] Schools of Mathematics and Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel and IAS, Princeton, NJ, 08540, USA. Research supported in part by the Israel Science Foundation, by a USA-Israel BSF grant, by NSF grant CCF 0832797 and by the Ambrose Monell Foundation. Email: nogaa@tau.ac.il

[†] School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel. Email: gutner@tau.ac.il

ISSN 1433-8092

1 Introduction

1.1 Motivation and background

Color Coding is an algorithmic technique for deciding efficiently if a given input graph contains a path or a cycle of a given length, or any other prescribed subgraph of bounded tree-width. Focusing, for simplicity, on paths, the method supplies a deterministic algorithm for deciding, in time $2^{O(k)}|E|\log|V|$, whether or not a given input (directed or undirected) graph $G = (V, E)$ contains a (simple) path on k vertices. The basic approach, introduced in [8], is very simple. One first gives a randomized algorithm, and then converts it into a deterministic one. The randomized algorithm works by first coloring the vertices of G randomly by k colors. Call a path on k vertices (a k-path, for short) colorful if its vertices get all the distinct k colors. It is not difficult to check in time $O(k2^k|E|)$, using dynamic programming, if there is a colorful path. As the probability of a k-path to become colorful in a random coloring is $k!/k^k > e^{-k}$, repeating the above procedure some Ce^k times provides a randomized algorithm in which the probability not to find a path in case one exists is smaller than e^{-C}. The crucial point in the derandomization of this algorithm is the observation that known constructions of families of hash functions given by [22] following [14], supply an explicit family of $2^{O(k)}\log|V|$ colorings of the vertices of G by k colors, so that the members of every set of k vertices get distinct colors in at least one of the colorings. Thus one can simply run the dynamic programming algorithm for each of these colorings, getting a deterministic algorithm for the problem.

The above technique has found several recent applications in computational biology (see [23], [24], [25], [17]), where it has been applied for detecting signaling pathways in protein interaction networks. These applications suggest the problem of counting, or approximating the number of k-paths (or other graphs of bounded tree-width) in a given graph. As using dynamic programming it is easy to count precisely the number of colorful k-paths in a given graph with colored vertices, the existence of efficient randomized approximation algorithms for counting follows quite easily by following the same approach; this is done in [2].

In order to derandomize the randomized counting (or approximate counting) procedures, one needs a strengthening of the usual notion of hash functions. This is given in the following definition. A family of functions from $[n]$ to $[\ell]$ is an (ϵ, k)-balanced family of hash functions, if for every $S \subset [n]$, $|S| = k$, the number of functions that are one-to-one on S is between $(1-\epsilon)T$ and $(1+\epsilon)T$ for some constant $T > 0$. The family is perfectly k-balanced if it is $(0, k)$-balanced, that is, it is (ϵ, k)-balanced for $\epsilon = 0$.

Note that with a perfectly k-balanced family one can count precisely the number of k-paths in a graph on n vertices: we simply count, by dynamic programming, the number of colorful k-paths for each of the functions (considered as a coloring of the vertices), sum the results and divide by T. Similarly, an (ϵ, k)-balanced family will enable us to approximate the number of paths up to a relative error of ϵ. This suggests the study of the smallest possible size of such families, and the problem of constructing explicitly such families.

1.2 Related work

The problem of counting paths and cycles in graphs has been considered by various researchers. In [9] the authors describe an $O(|V|^\omega)$ algorithm for counting the number of cycles of size at most 7, where $\omega < 2.38$ is the exponent in fast matrix multiplication. The method of this paper does not extend to longer paths, and indeed Flum and Grohe [13] proved that the problem of counting *exactly* the number of paths and cycles of length k in both directed and undirected graphs, considered as a problem parameterized by k, is $\#W[1]$-complete. This implies that it is unlikely that there is an $f(k) \cdot n^c$-time algorithm for counting the precise number of paths or cycles of length k in a graph of size n for any computable function $f : \mathbb{N} \to \mathbb{N}$ and constant c. The best known algorithms for computing exactly the number of k-paths in an n vertex graph run in time $n^{k/2+O(1)}$, see [11], [26].

However, the problem of approximating these numbers is more tractable. Arvind and Raman (see [10]) obtained a *randomized* fixed-parameter tractable algorithm to approximately count the number of copies of k-paths (or any fixed subgraph with bounded tree-width) within a large graph. A similar approximation appears in [2].

In an earlier paper [4] we considered *deterministic* approximation counting algorithms for this problem. To this end, we introduced the notion of (ϵ, k)-balanced families of hash functions and used them to exhibit a deterministic polynomial time algorithm for approximating the number of paths of length k up to any $k \leq O(\frac{\log n}{\log \log \log n})$ in a graph with n vertices. This was done by constructing explicitly (ϵ, k)-balanced families from $[n]$ to $[k]$, where the size of the family is $2^{O(k \log \log k)} \log n$ and the time for construction is $2^{O(k \log \log k)} n \log n$. The main open problem raised in [4] is to find such a construction of size $2^{O(k)} \log n$ (in time $2^{O(k)} n^{O(1)}$), which is optimal, even for standard (non-balanced) families of hash functions, and will supply polynomial time deterministic approximation algorithms for counting the number of paths of length k in a given graph of size n, for all $k \leq O(\log n)$. This problem is settled in the present paper.

1.3 The new results

The results of Flum and Grohe mentioned above suggest that there is no perfectly k-balanced family of hash functions from $[n]$ to $[k]$ of size $f(k)n^{O(1)}$. We prove a stronger result, showing that every perfectly k-balanced family of hash functions from $[n]$ to $[\ell]$ is of size at least $c(k, \ell) n^{\lfloor k/2 \rfloor}$, where $c(k, \ell)$ is a positive constant depending only on k and ℓ. We also observe that this is not far from being tight, as for every $n > k$ there is a perfectly k-balanced family of functions from $[n]$ to $[k]$ of size $\binom{n}{k-1}$. This shows that the Color Coding approach cannot supply an algorithm for counting k-paths in an n vertex graph in time $o(n^{\lfloor k/2 \rfloor})$.

Our main positive result is an explicit construction, for every $\frac{1}{poly(k)} < \epsilon \leq 1$, of an (ϵ, k)-balanced family of hash functions from $[n]$ to $[k]$ of size $e^{k+O(\log^3 k)} \log n$. The running time of the procedure that provides the construction is $e^{k+O(\log^3 k)} n \log n$. Note that the size of the family is optimal up to the error term $O(\log^3 k)$ in the exponent, as there is a known lower bound of $\Omega(e^k \log n / \sqrt{k})$ for the size of any family of hash functions from $[n]$ to $[k]$, (even if it is not balanced and the only requirement is that every set of size $[k]$ is mapped in a one-to-one fashion at least once).

This supplies deterministic approximation algorithms for counting the number of simple k-paths in a graph $G = (V, E)$ up to a relative error of $\epsilon = \frac{1}{poly(k)}$ in time $2^{O(k)} |E| \log |V|$. Similar results hold for counting approximately the number of copies of any graph of size k with constant tree-width. Note that this is polynomial for all $k \leq O(\log n)$, and it is unlikely that one can do better, as this would imply the existence of a $2^{o(n)}$-time algorithm for the Hamilton path problem, contradicting the Exponential Time Hypothesis of [18, 19].

1.4 Methods and organization

Our lower bound for the size of perfectly balanced families are proved by Linear Algebra tools, combining the basic approach of [1] in the proof of the lower bound for the size of sample spaces supporting k-wise independent random variables with two additional ideas.

The construction of (ϵ, k)-balanced families combines several ingredients. Two of them are rather standard and are based on nearly pairwise independent random variables and on the method of conditional expectations. The third one is more challenging, and combines the approach of [21] with an iterative construction based on properties of expanders. It is convenient to apply here (some version of) the expanders of [5], though other expanders could have been used as well.

Since our main motivation is the application for the subgraph approximate counting problem using Color Coding, there is no reason to provide explicit constructions of (ϵ, k)-balanced families of functions which are more efficient than the time of writing these functions down, as anyway our counting algorithm will have to go through these functions. We thus describe the constructions in this way, without trying to describe separately which parts of them admit more efficient descriptions. It is worth noting, however, that the part of our construction which applies the method of conditional expectations indeed requires the time stated in its description.

The rest of this extended abstract is organized as follows. In section 2 we describe the main ingredients of the construction: balanced families of hash functions and balanced splitters, a modified version of a notion introduced in [21]. Section 3 contains the results concerning perfectly balanced families of hash functions. The explicit construction of expanders presented in section 4 is used in section 5 for constructing small sample spaces supporting a certain relaxed version of nearly k-wise independent random variables. This is used to obtain a construction of what we call balanced (n, k, ℓ)-splitters, which is later applied in section 6 as a crucial ingredient in the construction of balanced families of hash functions. The constructions, together with the color coding technique, are used for designing algorithms for approximately counting the number of copies of subgraphs of bounded tree-width in given graphs. We conclude with some remarks and open problems. Due to space limitations, some of the proofs are given in the appendix.

2 The ingredients of the construction

In this section we formally define the notions of balanced families of hash functions and balanced splitters. For a positive integer n, denote by $[n]$ the set $\{1, \ldots, n\}$. For any k, $1 \leq k \leq n$, the family of k-sized subsets of $[n]$ is denoted by $\binom{[n]}{k}$. As usual, $k \bmod \ell$ denotes the unique integer $0 \leq r < \ell$ so that $k = q\ell + r$, for some integer q.

Definition 2.1. *Suppose that $1 \leq k \leq \ell \leq n$ and $\epsilon \geq 0$. A family of functions from $[n]$ to $[\ell]$ is an (ϵ, k)-balanced family of hash functions if there exists a constant $T > 0$, such that for every $S \in \binom{[n]}{k}$, the number of functions that are one-to-one on S is between $(1-\epsilon)T$ and $(1+\epsilon)T$. The family is perfectly k-balanced if it is $(0, k)$-balanced.*

The following definition is motivated by a related notion defined and used in [21].

Definition 2.2. *Suppose that $1 \leq \ell < k \leq n$ and $\epsilon \geq 0$, and let H be a family of functions from $[n]$ to $[\ell]$. For a set $S \in \binom{[n]}{k}$, let $split_H(S)$ denote the number of functions $h \in H$ so that for every j, $1 \leq j \leq k \bmod \ell$, $|h^{-1}(j) \cap S| = \lceil k/\ell \rceil$, and for all $k \bmod \ell < j \leq \ell$, $|h^{-1}(j) \cap S| = \lfloor k/\ell \rfloor$. The family H is an ϵ-balanced (n, k, ℓ)-splitter if there exists a constant $T > 0$, such that for every $S \in \binom{[n]}{k}$, $(1-\epsilon)T \leq split_H(S) \leq (1+\epsilon)T$.*

Note that if ℓ divides k, then in the above definition $split_H(S)$ is the number of functions that split S into equal parts. The splitters of [21] differ from the ones defined here, just as usual families of hash functions differ from balanced families; in [21] it is only required that for every set S there will be some function in H splitting it evenly, while in our splitters each S should be divided evenly by roughly the same number of functions. The construction of balanced splitters is thus much harder than the one of splitters in [21], and is in fact the most challenging part in the explicit description of balanced families of hash functions.

Each function f in our explicit construction of balanced families of hash functions is the composition of members from three families. The first one is an (ϵ_1, k)-balanced family of hash functions from $[n]$ to $[q]$, where $q = \Theta(\frac{k^2}{\epsilon})$. The second one is an ϵ_2-balanced (q, k, ℓ)-splitter from $[q]$ to $[\ell]$, where $\ell = \Theta(\log k)$, and the last one is an $(\epsilon_3, k/\ell)$-balanced family of hash functions from $[q]$ to $[k/\ell]$ (for simplicity assume for now that ℓ divides k). In order to define f we actually need ℓ members of the third family, with each of them being applied to the elements mapped by the members of the second family to a single $j \in [\ell]$.

3 Perfectly Balanced Families

Let $n > \ell \geq k > 0$ be positive integers. Recall that a family $\mathcal{F}$ of functions from $[n]$ to $[\ell]$ is perfectly k-balanced, if there exists a number $T > 0$ so that for every set $K \subset [n]$ of size $|K| = k$, $|\{f \in \mathcal{F} : |f(K)| = k\}| = T$. In this section we show that the size of each such family must be at least $c(k, \ell)n^{\lfloor k/2 \rfloor}$, where $c(k, \ell)$ is a positive constant depending only on k and ℓ.

Theorem 3.1. *Let $\mathcal{F}$ be a perfectly k-balanced family of functions from $[n]$ to $[\ell]$, where $n > \ell \geq k$.*
(i) If $k = 2r$ is even then

$$|\mathcal{F}| \geq \frac{\binom{n}{r}}{\binom{\ell}{r}\binom{\ell-r}{r}}.$$

(ii) If $k = 2r + 1$ is odd then

$$|\mathcal{F}| \geq \frac{\binom{n-1}{r}}{\binom{\ell-1}{r}\binom{\ell-r-1}{r}}.$$

(iii) If $\ell = k = 2$ then $|\mathcal{F}| \geq n - 1$, and equality can hold if and only if there is a Hadamard matrix of order n. Otherwise, the smallest possible size of $\mathcal{F}$ is precisely n.

Proof. We describe the proof of part (i). The proofs of parts (ii) and (iii), which are similar, but require some additional ideas, are given in the appendix.
(i) Let $\mathcal{F}$ be a perfectly $2r$-balanced family of functions from $[n]$ to $[\ell]$.

For each $R \subset [n]$ of size $|R| = r$, define two vectors u_R and w_R, each of length $|\mathcal{F}|\binom{\ell}{r}\binom{\ell-r}{r}$, whose coordinates are indexed by the set of all ordered triples (f, S_1, S_2), with

$$f \in \mathcal{F},\ S_1, S_2 \subset [\ell],\ |S_1| = |S_2| = r, \text{ and } S_1 \cap S_2 = \emptyset.$$

These vectors are defined as follows:

$$u_R(f, S_1, S_2) = 1 \text{ if } f(R) = S_1, \text{ and } u_R(f, S_1, S_2) = 0 \text{ otherwise.}$$

$$w_R(f, S_1, S_2) = 1 \text{ if } f(R) = S_2, \text{ and } w_R(f, S_1, S_2) = 0 \text{ otherwise.}$$

Note that the inner product of two such vectors u_{R_1} and w_{R_2} is zero if $R_1 \cap R_2 \neq \emptyset$. Indeed, in this case $f(R_1)$ must have a nonempty intersection with $f(R_2)$ for all $f \in \mathcal{F}$, and thus there is no coordinate (f, S_1, S_2) as above in which both v_{R_1} and w_{R_2} do not vanish. Similarly, if $R_1 \cap R_2 = \emptyset$, the inner product of u_{R_1} and w_{R_2} is precisely the number of functions $f \in \mathcal{F}$ which are one-to-one on $R_1 \cup R_2$. Indeed, for each such f there is a unique pair of disjoint sets S_1, S_2, each of size r, so that $f(R_1) = S_1$ and $f(R_2) = S_2$, while if f maps two elements of $R_1 \cup R_2$ to the same image, there is no such pair. Since $\mathcal{F}$ is a perfectly balanced $2r$-family, there exists a positive T so that for every disjoint R_1, R_2 as above, the inner product of u_{R_1} with w_{R_2} is T.

Let U be the $\binom{n}{r}$ by $|\mathcal{F}|\binom{\ell}{r}\binom{\ell-r}{r}$ matrix whose rows are all vectors u_R with $R \subset [n], |R| = r$, and let W be the matrix whose rows are all vectors w_R. By the above discussion, the product $U \cdot W^t = T \cdot DIS_{n,r}$, where $DIS_{n,r}$ is the disjointness matrix whose rows and columns are indexed by the r-subsets of $[n]$, defined by $DIS_{n,r}(R_1, R_2) = 1$ if $R_1 \cap R_2 = \emptyset$ and $DIS_{n,r}(R_1, R_2) = 0$ otherwise. It is well known (see, e.g., [20]) that the matrix $DIS_{n,r}$ is nonsingular (over the reals) for all $n \geq 2r$, and as this is the case here and T is nonzero, it follows that the rank of U is at least that of $U \cdot W^t$ which is $\binom{n}{r}$. As this rank is at most the number of columns of U, we conclude that

$$|\mathcal{F}|\binom{\ell}{r}\binom{\ell-r}{r} \geq \binom{n}{r},$$

completing the proof of part (i). □

Remarks:
(i) A well known conjecture (c.f., e.g., [15]) asserts that for $n > 2$ there is a Hadamard matrix of order n iff n is divisible by 4. It is easy to see that if there is such a matrix that n is indeed divisible by 4. The converse is not known, but there are many infinite families of known Hadamard matrices, showing that the $(n-1)$-bound in part (iii) of the theorem is tight in many cases.
(ii) It is easy to see that for every $n > k$ there is a perfectly k-balanced family $\mathcal{F}$ of functions from $[n]$ to $[k]$ of size $|\mathcal{F}| = \binom{n}{k-1}$. Indeed, for each subset $R = \{r_1, r_2, \ldots, r_{k-1}\}$ of $[n]$, with $r_1 < r_2 < \ldots < r_{k-1}$ let f_R denote the function defined by $f_R(r_i) = i$ for all $1 \leq i \leq k-1$, and $f_R(j) = k$ for all $j \in [n] - R$. It is not difficult to check that the family of all these functions f_R is perfectly k-balanced (with $T = k$).
(iii) The lower bounds in Theorem 3.1 hold for weighted families as well, even if the weight $weight(f)$ of some of the functions f is negative, as long as there is a real $T \neq 0$ so that for every $K \subset [n]$, $|K| = k$, the total weight of functions which are one-to-one on K is exactly T. To see this, repeat the proof above, modifying the definition of the vectors u_R to be

$$u_R(f, S_1, S_2) = weight(f) \text{ if } f(R) = S_1, \text{ and } u_R(f, S_1, S_2) = 0 \text{ otherwise},$$

keeping the definition of the vectors w_R as before.

4 Expanders

In this section we describe a special case of the Cayley expanders of [5] that suffices for our purposes.

The following are standard definitions and observations concerning eigenvalues and expanders (c.f., e.g., [7],[16]).

Let $G = (V, E)$ be a d-regular graph and let $A = A_G = (a_{uv})_{u,v \in V}$ be its adjacency matrix. Since G is d-regular, the largest eigenvalue of A is d, corresponding to the all 1 eigenvector. Let

$\lambda = \lambda(G)$ denote the largest absolute value of an eigenvalue other than the first one. For two (not necessarily disjoint) subsets B and C of V, let $e(B, C)$ denote the number of ordered pairs (u, v), where $u \in B$, $v \in C$ and uv is an edge of G. The following useful bound is the Expander Mixing Lemma (c.f., e.g., [7], page 146).

Proposition 4.1. *Let G be a d-regular graph with n vertices and set $\lambda = \lambda(G)$. For every two sets of vertices B and C of G, where $|B| = bn$ and $|C| = cn$, we have*

$$|e(B, C) - bcdn| \leq \lambda\sqrt{bc}\, n.$$

We need the following explicit expanders, described, for example, in [6], following [5].

Theorem 4.2. *For every two positive integers d and k satisfying $4^k < 2^d$ there is an explicit construction of a 4^k-regular graph $G_{d,k}$ on 2^d vertices so that $\lambda(G_{d,k}) \leq d \cdot 2^k$.*

Note that this construction is applicable for a wide range of parameters, that is, the number of vertices of the expander can be any power of 2, whereas the degree can be any power of 4. For completeness, we include the construction in the appendix.

5 Partially independent variables

In this section we introduce a certain relaxation of almost k-wise independence and describe an appropriate explicit construction, which will give the main building block required in the construction of balanced families of hash functions of optimal size. For notational convenience, we give the following definitions related to the probabilities implied by a multinomial distribution.

Definition 5.1. *Suppose that $1 \leq \ell \leq k$ and $k = k_1 + k_2 + \cdots + k_\ell$, where $k_i \geq 0$ for every i. Define $m(k_1, \ldots, k_\ell)$ to be the following probability:*

$$\frac{\binom{k}{k_1, k_2, \ldots, k_\ell}}{\ell^k} = \frac{k!}{k_1! k_2! \cdots k_\ell! \ell^k}.$$

For random variables $X_1, \ldots, X_k$, let Y_i denote the number of variables X_j that are equal to i. Define $M(X_1, \ldots X_k; k_1, \ldots k_\ell)$ to be the event that $Y_i = k_i$ for every i, $1 \leq i \leq \ell$.

We now construct probability distributions which are uniform over a set of strings of length q in the alphabet $[\ell]$. In the standard notion of almost k-wise independence, it is required that in any k positions, each substring of length k appears with probability close to ℓ^{-k}. Here we are interested in a weaker condition. Our objective is to construct small probability spaces of the following type.

Definition 5.2. *A sequence $X_1, \ldots X_q$ of random variables that take values from $[\ell]$ is (ϵ, k)-partially-independent if for any $p \leq k$ positions $i_1 < \cdots < i_p$ and any ℓ values $k_1, \ldots, k_\ell$ such that $k_1 + \cdots + k_\ell = p$, we have*

$$|Pr[M(X_{i_1}, \ldots X_{i_p}; k_1, \ldots k_\ell)] - m(k_1, \ldots, k_\ell)| < \epsilon.$$

Observe that we require the property to hold for any p variables, where $1 \leq p \leq k$. This is needed since the fact that the property is satisfied for a value p does not imply that it holds for $p' < p$. Furthermore, requiring that it applies for every value $p \leq k$ is crucial for the correctness of our recursive construction. To demonstrate the definition, here is what it means for

$\ell = 2$. A sequence $X_1, \dots X_q$ of random Boolean variables (taking values from {0,1}) is (ϵ, k)-partially-independent if for any $p \leq k$ positions $i_1 < \cdots < i_p$ and any r, $0 \leq r \leq p$, we have $\left|Pr[X_{i_1} + \cdots + X_{i_p} = r] - \binom{p}{r} 2^{-p}\right| < \epsilon$.

Theorem 5.3. *For any $\ell \leq k \leq q$ and $0 < \epsilon \leq 1$, a sample space of size $\left(\frac{qk^\ell}{\epsilon}\right)^{O(\log q)}$ that supports q variables that take values from $[\ell]$ and are (ϵ, k)-partially-independent can be constructed in time $\left(\frac{qk^\ell}{\epsilon}\right)^{O(\log q)}$.*

Proof. Assume, without loss of generality, that q is a power 2. Otherwise, q can be simply rounded to the next power of 2. Assume also that $\epsilon \leq \frac{1}{k^\ell}$. If this is not the case, then ϵ can be replaced by $\frac{\epsilon}{k^\ell}$. We recursively construct sample spaces that support an increasing number of variables. For every $t = 0, 1, \dots, \log_2 q$, we shall construct a sample space C_t that supports 2^t variables that take values from $[\ell]$ and are $\left(\frac{4^t \epsilon}{q^2}, k\right)$-partially-independent. The sample space C_t will consists of strings of length 2^t over the alphabet $[\ell]$.

We start with $t = 0$. To support one variable, it is possible to simply define a sample space that consists of the ℓ strings of length 1, and there will be no error at all in this case. For our purpose, the size of each sample space should be a power of 2, so let N_0 be the result of rounding the value $4(\frac{20q^2k^\ell}{\epsilon})^4$ to the next higher power of 2. The sample space consists of N_0 strings, where each string of length 1 appears either $\lfloor \frac{N_0}{\ell} \rfloor$ or $\lceil \frac{N_0}{\ell} \rceil$ times. Obviously $N_0 \leq 8(\frac{20q^2k^\ell}{\epsilon})^4$ and we have one variable which is certainly $\left(1, \frac{\epsilon}{q^2}\right)$-partially-independent.

Let D be the result of rounding the value $\left(\frac{20q^2k^\ell}{\epsilon}\right)^4$ to the next higher power of 4. Suppose that in step t, a sample space of size $N_t \leq 8D^{t+1}$ that supports 2^t variables that are $\left(\frac{4^t\epsilon}{q^2}, k\right)$-partially-independent has been constructed. We now describe step $t+1$. Let G be the D-regular expander with N_t vertices described in section 4 (note that $D < N_t$). It follows from Theorem 4.2 that

$$\frac{\lambda(G)}{D} \leq \frac{\log_2 N_t}{\sqrt{D}} \leq \frac{3 + (t+1)\log_2 D}{\sqrt{D}} \leq \frac{(\log_2 D)^2}{\sqrt{D}} \leq \frac{20}{D^{1/4}} \leq \frac{\epsilon}{q^2 k^\ell}.$$

To every vertex of the graph G we assign one of the N_t strings of length 2^t from C_t that were constructed in step t. For every ordered pair (u, v) such that uv is an edge of G, the concatenation of the string assigned to u followed by the string assigned to v is added to the sample space C_{t+1}. The resulting sample space is of size $N_{t+1} = DN_t$.

Suppose that in step t, a sample space C_t of size N_t that supports 2^t variables that are (γ, k)-partially-independent has been constructed, where $\gamma = \frac{4^t\epsilon}{q^2}$. We now prove that the approximation error is increased in step $t+1$ by a multiplicative factor of at most 4, that is, the sample space C_{t+1} supports 2^{t+1} variables that are $(4\gamma, k)$-partially-independent. Suppose that $p \leq k$ and take any p positions $1 \leq i_1 < \cdots < i_p \leq 2^{t+1}$ and any ℓ values $k_1, \dots, k_\ell$ such that $k_1 + \cdots + k_\ell = p$. We further assume that among the p positions selected, exactly p' positions are in the first half of the string. Therefore $Pr[M(X_{i_1}, \dots X_{i_p}; k_1, \dots k_\ell)]$ is equal to

$$\sum_{k'_1 + \cdots + k'_\ell = p'} Pr[M(X_{i_1}, \dots X_{i_{p'}}; k'_1, \dots, k'_\ell) \cap M(X_{i_{p'+1}}, \dots X_{i_p}; k_1 - k'_1, \dots, k_\ell - k'_\ell)].$$

We would like $Pr[M(X_{i_1},\dots X_{i_p};k_1,\dots k_\ell)]$ to be close to:

$$m(k_1,\dots k_\ell) = \sum_{k'_1+\cdots+k'_\ell=p'} m(k'_1,\dots,k'_\ell)m(k_1-k'_1,\dots,k_\ell-k'_\ell).$$

Note that the number of terms in the two summations above is at most k^ℓ and that obviously $\sum_{k'_1+\cdots+k'_\ell=p'} m(k'_1,\dots,k'_\ell)\le 1$. Since C_t is (γ,k)-partially-independent, it follows from Proposition 4.1 that the estimation error is as follows:

$$\begin{aligned}
&\left|Pr[M(X_{i_1},\dots X_{i_p};k_1,\dots k_\ell)]-m(k_1,\dots k_\ell)\right| \le \\
&\quad\sum_{k'_1+\cdots+k'_\ell=p'}\left[(m(k'_1,\dots,k'_\ell)+\gamma)(m(k_1-k'_1,\dots,k_\ell-k'_\ell)+\gamma)+\frac{\lambda(G)}{D}\right]- \\
&\quad\sum_{k'_1+\cdots+k'_\ell=p'} m(k'_1,\dots,k'_\ell)m(k_1-k'_1,\dots,k_\ell-k'_\ell) = \\
&\quad\sum_{k'_1+\cdots+k'_\ell=p'} \gamma[m(k'_1,\dots,k'_\ell)+m(k_1-k'_1,\dots,k_\ell-k'_\ell)]+\gamma^2+\frac{\lambda(G)}{D}\le \\
&\quad 2\gamma+k^\ell\left(\gamma^2+\frac{\lambda(G)}{D}\right)\le 4\gamma,
\end{aligned}$$

where the last inequality follows from the inequalities $\gamma\le\epsilon\le\frac{1}{k^\ell}$ and $\frac{\lambda(G)}{D}\le\frac{\epsilon}{q^2k^\ell}\le\frac{\gamma}{k^\ell}$. After step $\log_2 q$, the sample space constructed is (ϵ,k)-partially-independent, as needed.

□

6 Balanced Families and Approximate Counting

The following inequality is Robbins' formula [12] (a tight version of Stirling's formula).

Claim 6.1. *For every integer $n\ge 1$,*

$$\sqrt{2\pi}n^{n+1/2}e^{-n+1/(12n+1)} < n! < \sqrt{2\pi}n^{n+1/2}e^{-n+1/(12n)}.$$

This supplies the following simple lower bound for the multinomial distribution (recall Definition 5.1).

Lemma 6.2. *If $k\ge\ell>0$, then*

$$m(\underbrace{\lceil k/\ell\rceil,\dots,\lceil k/\ell\rceil}_{k \bmod \ell},\underbrace{\lfloor k/\ell\rfloor,\dots,\lfloor k/\ell\rfloor}_{\ell-(k \bmod \ell)}) > (15k/\ell)^{-\ell/2}.$$

Proof (sketch). Assume first that ℓ divides k. Using Robbins' formula, we get:

$$m(\underbrace{k/\ell,\dots,k/\ell}_{\ell}) = \frac{k!}{(k/\ell)!^\ell\ell^k} > (2\pi k/\ell)^{-\ell/2}e^{-\ell^2/12k}\ge(2\pi e^{1/6}k/\ell)^{-\ell/2} > (7.5k/\ell)^{-\ell/2}.$$

The result for general k and ℓ follows similarly. □

The previous Lemma shows that the events we would like to estimate have a relatively high probability, enabling us to give the following construction.

Theorem 6.3. *For any $k \geq \ell$ and $0 < \epsilon \leq 1$, an ϵ-balanced (q, k, ℓ)-splitter of size $\left(\frac{qk^\ell}{\epsilon}\right)^{O(\log q)}$ can be constructed in time $\left(\frac{qk^\ell}{\epsilon}\right)^{O(\log q)}$.*

Proof. As implied by Theorem 5.3, we use an explicit probability space of size $\left(\frac{qk^\ell}{\gamma}\right)^{O(\log q)}$ that supports q random variables that take values from $[\ell]$ and are (γ, k)-partially-independent, where $\gamma = (15k/\ell)^{-\ell/2}\epsilon$. We attach one of the random variables to each element of $[q]$. If follows from Lemma 6.2 that the splitter achieves the required approximation. □

We can now describe our main construction of balanced families of hash functions, using the ingredients mentioned at the end of section 2. Recall that there are three ingredients in this construction. Two of them are relatively simple, and are given in the next two propositions.

Proposition 6.4. *For any $0 < \epsilon \leq 1$, an (ϵ, k)-balanced family of hash functions from $[n]$ to $[q]$, where $q = \lceil \frac{2k^2}{\epsilon} \rceil$, of size $\frac{k^{O(1)} \log n}{\epsilon^{O(1)}}$ can be constructed in time $\frac{k^{O(1)} n \log n}{\epsilon^{O(1)}}$.*

Proposition 6.5. *For any $0 < \epsilon \leq 1$, an (ϵ, g)-balanced family of hash functions from $[m]$ to $[g]$ of size $O(\frac{e^g \sqrt{g} \log m}{\epsilon^2})$ can be constructed in time $\binom{m}{g} \frac{e^g g^{O(1)} m \log m}{\epsilon^2}$.*

The first proposition is proved using a standard construction of nearly pairwise independent random variables. Here n is the number of variables, they attain values in $[q]$, and the number of functions is the size of the sample space. Since every two variables are equal with probability close to $1/q$, for every fixed set S of k variables, the values of the random variables in S are pairwise distinct in at least a fraction of $(1 - \epsilon)$ of the functions.

The second proposition is proved using the method of conditional expectations. The details appear in [4].

The main part of the construction is the balanced (q, k, ℓ)-splitter described in Theorem 6.3. The three ingredients are combined as follows. Each function f of our final family is described by a member f_1 of an $(\epsilon/6, k)$-balanced family of Proposition 6.4, a member f_2 of the ϵ_2-balanced splitter of Theorem 6.3 with $\epsilon_2 = \frac{\epsilon}{6}, q = \lceil \frac{2k^2}{\epsilon_2} \rceil$ and $\ell = \lceil \log k \rceil$, and ℓ members $\phi_1, \ldots, \phi_\ell$ of the $(\frac{\epsilon}{6\ell}, g)$-balanced family of Proposition 6.5 with $m = q$ and $g = k/\ell$. (For simplicity we assume here that ℓ divides k.) To compute the value of f on some $x \in [n]$, we first apply f_1 to x, getting a value y in $[q]$, then we apply f_2 to y, getting as a result some $i \in [\ell]$, and finally we apply ϕ_i to y, where the final result is $(i-1)k/\ell + \phi_i(y)$. A k-set $S \subset [n]$ can be mapped in a one-to-one manner by such an f only if it is mapped in a one-to-one manner by f_1, and then only if it is split evenly into ℓ parts by f_2, and then only if its elements mapped to each of the ℓ parts are mapped in a one-to-one manner by each of the functions ϕ_i. Since all the ingredients in the construction are sufficiently balanced, this gives the required balanced family. The detailed computation, which yields the following theorem, is postponed to the full version of the paper.

Theorem 6.6. *For $\frac{1}{poly(k)} < \epsilon \leq 1$, an (ϵ, k)-balanced family of hash functions from $[n]$ to $[k]$ of size $e^{k+O(\log^3 k)} \log n$ can be constructed in time $e^{k+O(\log^3 k)} n \log n$.*

Using Color-Coding we can now approximate the number of paths and cycles (or other fixed graphs of bounded tree-width) in a given input graph. Let $G = (V, E)$ be a directed or undirected graph. The algorithms use the construction of (ϵ, k)-balanced families of hash functions from V to $[k]$. Each such function defines a coloring of the vertices of the graph. Recall that a path is colorful if each vertex on it is colored by a distinct color. Using dynamic programming one can count efficiently the exact number of colorful paths in each of these colorings. The properties of the balanced family of hash functions then provide the following deterministic polynomial time algorithms for approximately counting the number of paths or cycles of size k in a given input graph of size n for all $k \leq \log n$. Similar results apply for approximate counting of prescribed subgraphs of size k and bounded tree-width.

Theorem 6.7. *For any* $\frac{1}{poly(k)} < \epsilon \leq 1$*, the number of simple (directed or undirected) paths of k vertices in a (directed or undirected) graph* $G = (V, E)$ *can be approximated deterministically up to relative error* ϵ *in time* $2^{O(k)}|E|\log|V|$*.*

Theorem 6.8. *For any* $\frac{1}{poly(k)} < \epsilon \leq 1$*, the number of simple (directed or undirected) cycles of size* k *in a (directed or undirected) graph* $G = (V, E)$ *can be approximated deterministically up to relative error* ϵ *in time* $2^{O(k)}|E||V|\log|V|$*.*

7 Concluding Remarks

- The notion of balanced families of hash functions seems natural and useful, and it will be interesting to find additional applications of it.

- An easy combination of Proposition 6.4 and Theorem 6.6 supplies, for any $\epsilon \geq \frac{1}{k^{\ell}}$, explicit ϵ-balanced (n, k, ℓ)-splitters of size at most $e^{O(\ell \log^2 k)} \log n$. In particular, for $\ell = 2$ the size is $e^{O(\log^2 k)} \log n$. A simple probabilistic argument shows, however, that for any fixed $\epsilon > 0$ there are ϵ-balanced $(n, k, 2)$-splitters of size $O(k\sqrt{k} \log n)$, and although this is not crucial for our application here, it will be interesting to find an explicit construction of such splitters of size polynomial in k and $\log n$.

- Our results settle the problem of approximately counting the number of paths and cycles of length $k = \Theta(\log n)$ in an n-vertex graph in deterministic polynomial time. As mentioned in the introduction, it is probably impossible to extend the result for larger values of k, since even a polynomial time algorithm for deciding whether there exists one simple path of length k where $\log n = o(k)$ would imply a sub-exponential time algorithm for the Hamiltonian cycle problem. This follows easily by padding a graph on k vertices by $n - k = 2^{o(k)}$ isolated ones, thus converting the above decision algorithm to one that decides in time $2^{o(k)}$ whether a graph on k vertices is Hamiltonian, contradicting the Exponential Time Hypothesis (ETH) [18, 19].

- Our method here, combined with the Color Coding technique, easily yields results for additional approximate counting problems for graphs. In particular, given a weighted graph G on n vertices, we can approximate deterministically, in polynomial time, the number of minimum (or maximum) weight paths or cycles (or copies of any prescribed subgraph of bounded tree width) on k vertices in G up to any fixed desired relative accuracy, for all $k \leq O(\log n)$.

References

[1] Noga Alon, Lászlö Babai, and Alon Itai. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. *Journal of Algorithms* 7, 567–583, 1986.

[2] Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and Süleyman Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. In *ISMB*, pages 241–249, 2008.

[3] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[4] Noga Alon and Shai Gutner. Balanced families of perfect hash functions and their applications. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 435–446. Springer, 2007.

[5] Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994.

[6] Noga Alon, Oded Schwartz, and Asaf Shapira. An elementary construction of constant-degree expanders. *Combin. Probab. Comput.*, 17(3):319–327, 2008.

[7] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., Hoboken, NJ, third edition, 2008.

[8] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, July 1995.

[9] Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, March 1997.

[10] Vikraman Arvind and Venkatesh Raman. Approximation algorithms for some parameterized counting problems. In Prosenjit Bose and Pat Morin, editors, *ISAAC*, volume 2518 of *Lecture Notes in Computer Science*, pages 453–464. Springer, 2002.

[11] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. The fast intersection transform with applications to counting paths. To appear.

[12] William Feller. *An Introduction to Probability Theory and its Applications. Vol. I.* Third edition. Wiley, New York, 1968.

[13] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, August 2004.

[14] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31(3):538–544, July 1984.

[15] Marshall Hall, Jr. *Combinatorial Theory*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Inc., New York, second edition, 1986. A Wiley-Interscience Publication.

[16] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561, 2006.

[17] Falk Hüffner, Sebastian Wernicke, and Thomas Zichner. Algorithm engineering for color-coding to facilitate signaling pathway detection. In David Sankoff, Lusheng Wang, and Francis Chin, editors, *APBC*, volume 5 of *Advances in Bioinformatics and Computational Biology*, pages 277–286. Imperial College Press, 2007.

[18] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci*, 62(2):367–375, 2001.

[19] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci*, 63(4):512–530, 2001.

[20] Stasys Jukna. *Extremal combinatorics.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2001.

[21] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science*, pages 182–191, 1995.

[22] Jeanette P. Schmidt and Alan Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM Journal on Computing*, 19(5):775–786, October 1990.

[23] Jacob Scott, Trey Ideker, Richard M. Karp, and Roded Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology*, 13(2):133–144, 2006.

[24] Roded Sharan and Trey Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, 2006.

[25] Tomer Shlomi, Daniel Segal, Eytan Ruppin, and Roded Sharan. QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, 7:199, 2006.

[26] R. Williams. Counting weighted k-subgraphs exactly. To appear.

8 Appendix

8.1 Perfectly balanced families

Proof of Theorem 3.1, parts (ii) and (iii).

(ii) The proof is similar to that of part (i), with a few modifications. Here are the details. Let $\mathcal{F}$ be a perfectly $2r+1$-balanced family of functions from $[n]$ to $[\ell]$.

For each $R \subset [n-1]$ of size $|R| = r$ define two vectors u_R and w_R, each of length $|\mathcal{F}|\binom{\ell-1}{r}\binom{\ell-r-1}{r}$, whose coordinates are indexed by the set of all ordered triples (f, S_1, S_2), satisfying

$$f \in \mathcal{F},\ S_1, S_2 \subset [\ell] - \{f(n)\},\ |S_1| = |S_2| = r, \text{ and } S_1 \cap S_2 = \emptyset.$$

These vectors are defined as before:

$$u_R(f, S_1, S_2) = 1 \text{ if } f(R) = S_1, \text{ and } u_R(f, S_1, S_2) = 0 \text{ otherwise.}$$

$$w_R(f, S_1, S_2) = 1 \text{ if } f(R) = S_2, \text{ and } w_R(f, S_1, S_2) = 0 \text{ otherwise.}$$

It is clear that just as before, the inner product of two such vectors u_{R_1} and w_{R_2} is zero if $R_1 \cap R_2 \neq \emptyset$. Similarly, if $R_1 \cap R_2 = \emptyset$, the inner product of u_{R_1} and w_{R_2} is precisely the number of functions $f \in \mathcal{F}$ which are one-to-one on $R_1 \cup R_2 \cup \{n\}$. Indeed, for each such f there is a unique pair of disjoint subsets S_1, S_2 of $[\ell] - \{f(n)\}$, each of size r, so that $f(R_1) = S_1$ and $f(R_2) = S_2$, while if f does not map $R_1 \cup R_2 \cup \{n\}$ in a one-to-one manner, there is no such pair. As before, since $\mathcal{F}$ is a perfectly balanced $2r+1$-family, there exists a positive T so that for the matrices U and W whose rows are all vectors u_R and w_R, respectively, with $R \subset [n-1], |R| = r$, the product $U \cdot W^t = T \cdot DIS_{n-1,r}$. The desired result follows as before, since $DIS_{n-1,r}$ is nonsingular and yet its rank cannot exceed the number of columns of U. This completes the proof of part (ii).

(iii) Let $\mathcal{F}$ be a perfectly 2-balanced family of functions from $[n]$ to $[2]$. Note that by part (i), $|\mathcal{F}| \geq n/2$, but here one can improve the constant factor and obtain a tight bound. To do so, define, for each $i \in [n]$, a vector u_i of length $|\mathcal{F}|$, whose coordinates are indexed by the elements of $\mathcal{F}$, where here $u_i(f) = (-1)^{f(i)-1}$. It is easy to check that the inner product of u_i and u_j is $|\mathcal{F}|$ if $i = j$, and is $|\mathcal{F}| - 2T$ if $i \neq j$, where here $T > 0$ is the number of functions $f \in \mathcal{F}$ that map i and j to distinct elements. (This number is the same for all $i \neq j$, as $\mathcal{F}$ is perfectly 2-balanced.) We conclude that all diagonal elements of the gram matrix of the n vectors u_i are $|\mathcal{F}|$, while all other elements are $|\mathcal{F}| - 2T$. It is easy to check that this matrix is nonsingular unless the sum of its elements in each row is zero, in which case it has rank $n-1$. In fact, all eigenvalues of this matrix are $2T$, with multiplicity $n-1$, and the sum of all entries in a row, with multiplicity 1. (In case this sum is also $2T$, then the matrix is $2T$ times the identity matrix, and all eigenvalues are equal). We conclude that the length of the vectors, $|\mathcal{F}|$ is always at least $n-1$. Equality can hold only if the sum of elements in a row of the gram matrix is 0. In this case, $|\mathcal{F}| = n-1$ and $n-1-2T = -1$, that is, the inner product of each two of our n vectors is -1. For each $1 \leq i \leq n$, let $\overline{u_i}$ denote the vector obtained from u_i by adding to it a coordinate in which its value is 1. Then the vectors $\overline{u_i}$ are n pairwise orthogonal vectors of length n with $\{-1, 1\}$ entries, that is, they form the rows of a Hadamard matrix of order n. Thus, if there is no Hadamard matrix of order n then any family of perfectly 2-balanced functions from $[n]$ to $[2]$ has at least n functions. The family $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ in which $f_i(i) = 1$ and $f_i(j) = 2$ for all $j \neq i$ shows that this is tight, completing the proof of the theorem. □

8.2 Expanders

In this subsection we present the construction of the expanders described in Section 4. Note that these are not bounded-degree graphs, and their degrees grow with the number of vertices, but they suffice for our purpose. This is a special case of a construction suggested in [5], which is based on one of the codes described in [3].

Let $bin : GF(2^k) \mapsto \{0,1\}^k$ be a one-to-one mapping satisfying $bin(0) = 0^k$ and $bin(x+y) = bin(x) \oplus bin(y)$, where $\alpha \oplus \beta$ means the bit-by-bit xor of the binary strings α and β. (The standard representation of $GF(2^k)$ as a vector space satisfies the above conditions.) Given $x, y \in GF(2^k)$, let $< x, y >$ denote the bit $(bin(x), bin(y))_2$, where $(\alpha, \beta)_2$ is the inner-product mod 2 of the binary vectors α and β. For a fixed d and $x, y \in GF(2^k)$, the binary vector u_{xy} is defined as $< x, y >< x^2, y > \cdots < x^d, y >$. For every $d, k \geq 1$, we define a 4^k-regular graph $G_{d,k}$ with 2^d vertices, as follows. The vertex set is $\{0,1\}^d$ and every vertex v is adjacent to $v \oplus u_{xy}$ for all $x, y \in GF(2^k)$.

Proof of Theorem 4.2. Denote $F = GF(2^k)$, $D = \{0,1\}^d$, and let A be the $2^d \times 2^d$ adjacency matrix of $G_{d,k}$. For every $a = a_1 a_2 \cdots a_d \in D$, let v_a be the vector whose bth entry, where $b \in D$, satisfies $v_a(b) = (-1)^{(a,b)_2}$. Let $p_a(x)$ be the polynomial $\sum_{i=1}^{d} a_i x^i$ and denote $\lambda_a = \sum_{x,y \in F} (-1)^{<p_a(x), y>}$.

We now prove that v_a is an eigenvector of A over $\mathbb{R}$ with eigenvalue λ_a.

$$(Av_a)(b) = \sum_{c \in D} A_{bc} v_a(c) = \sum_{x,y \in F} v_a(b \oplus u_{xy}) = v_a(b) \sum_{x,y \in F} v_a(u_{xy}) = v_a(b) \sum_{x,y \in F} (-1)^{(a, u_{xy})_2} = \lambda_a v_a(b).$$

It is easy to verify that the 2^d vectors $\{v_a\}_{a \in D}$ are orthogonal, and therefore we found all the eigenvalues of A. It remains to bound the absolute value of λ_a. For a fixed $x \in F$, the term $\sum_{y \in F} (-1)^{<p_a(x), y>}$ is equal to 2^k if $p_a(x) = 0$, and to zero in case $p_a(x) \neq 0$. If $a \neq 0^d$, then $p_a(x)$ is a non-zero polynomial of degree at most d, and therefore has at most d roots. Thus, $|\lambda_a| \leq d \cdot 2^k$, as needed. □