TURING MACHINE ALGORITHMS AND STUDIES IN QUASI-RANDOMNESS

A Thesis
Presented to
The Academic Faculty

by

Subrahmanyam Kalyanasundaram

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Algorithms, Combinatorics, and Optimization

School of Computer Science
Georgia Institute of Technology
December 2011

TURING MACHINE ALGORITHMS AND STUDIES IN QUASI-RANDOMNESS

Approved by:

Professor Richard J. Lipton, Advisor
School of Computer Science
Georgia Institute of Technology

Professor Asaf Shapira, Advisor
School of Mathematics and School of Computer Science
Georgia Institute of Technology

Professor Dana Randall
School of Computer Science
Georgia Institute of Technology

Professor Prasad Tetali
School of Mathematics and School of Computer Science
Georgia Institute of Technology

Professor H. Venkateswaran
School of Computer Science
Georgia Institute of Technology

Date Approved: 12 October 2011

ACKNOWLEDGEMENTS

I am very fortunate to have Dick Lipton as my advisor. His passion and infectious enthusiasm towards research and everything in general inspired me a lot. I have benefitted immensely from the kind and generous support, financial and otherwise, that he lent me through my time here at Georgia Tech. The knowledge that I had Dick's support in whatever I wanted to pursue made it very comfortable for me. I was always amazed at his vast breadth of knowledge on past and present literature and his never ending collection of entertaining stories and anecdotes. I am also very thankful that Dick involved me to a great extent in his highly successful and popular blog. I learned a lot proofreading his posts, and it was a privilege to see these posts a day or two before the rest of the world.

I feel I am equally lucky to have Asaf Shapira as my advisor too. I started working with Asaf towards the end of my stay here at Georgia Tech. His attitude towards research, clarity in understanding of complex material, analogies comparing proofs with software, quirky sense of humor all made it a pleasure both in collaboration and conversation. I really have benefitted from his wonderful vision for presenting proofs, and papers in general.

I wish to thank ACO and the CS Theory group at Georgia Tech, for providing me with the great program and an intellectual atmosphere to do research, as well as the financial support. Faculty members Milena Mihail, Dana Randall, Robin Thomas and H. Venkateswaran have all helped guide me by giving their valuable advice at different times. I would also like to thank the College of Computing for having the faith in me to teach a full class, and also particularly thank Venkat for helping me thoroughly prepare for the class. I am also thankful to Ken Regan for our collaboration and the interesting discussions that we have had during his many a visit to Georgia Tech.

I have to mention all the theory students who provided a fun setting for me at work. The work culture in Georgia Tech is extremely collaborative and open minded. In particular I wish to thank Florin Constantin, Deeparnab Chakrabarty, Atish Das Sarma, Farbod Shokrieh, Anand Louis, Pushkar Tripathi, Karthekeyan Chandrasekaran and Elena Grigorescu for being very good friends too.

Outside the theory group, I am happy that I had the friendship of Avishek Aiyar, Ashish Sinha, Ashwin Kumar Suresh, Balaji Ganapathy and Varun Varun. I have cherished, and will miss, the countless racquetball sessions with Avishek and the trips to the bridge club with Florin.

Finally, above all, I would like to thank my family, without whom I wouldn't have made this far. I am thankful to my father and mother for always being there with me, having believed in me, and letting me pursue my dreams. I would like to thank my sister Vandukkutty for giving me constant love and unconditional support all through my time here. I would like to thank my wife Sangeetha, for giving me unwavering love, comfort and support and for being the best partner that I could have asked for.

TABLE OF CONTENTS

SUMMARY

Randomness is an invaluable resource in theoretical computer science. However, pure random bits are hard to obtain. Quasi-randomness is a tool that has been widely used in eliminating/reducing the randomness from randomized algorithms. In this thesis, we study some aspects of quasi-randomness in graphs. Specifically, we provide an algorithm and a lower bound for two different kinds of regularity lemmas. Our algorithm for FK-regularity is derived using a spectral characterization of quasi-randomness. We also use a similar spectral connection to also answer an open question about quasi-random tournaments. We then provide a "Wowzer" type lower bound (for the number of parts required) for the strong regularity lemma. Finally, we study the derandomization of complexity classes using Turing machine simulations.

Connections between quasi-randomness and graph spectra. Quasi-random (or pseudo-random) objects are deterministic objects that behave almost like truly random objects. These objects have been widely studied in various settings (graphs, hypergraphs, directed graphs, set systems etc.) [65]. In many cases, quasi-randomness is very closely related to the spectral properties of the combinatorial object that is under study [3, 4, 19, 26, 61]. In this thesis, we discover the spectral characterizations of quasi-randomness in two different cases to solve open problems.

A Deterministic Algorithm for Frieze-Kannan Regularity. The Frieze-Kannan regularity lemma is a powerful tool in combinatorics. The lemma asserts that any given graph of large enough size can be partitioned into a number of parts such that, across parts, the graph is quasi-random. The algorithmic applications of this lemma require one to efficiently construct a partition satisfying the conditions of the lemma. Williams [104] had asked if one can construct a partition satisfying the conditions of

the Frieze-Kannan regularity lemma in deterministic sub-cubic time. In this thesis, we answer this question by designing an $\tilde{O}(n^{\omega})$ time algorithm for constructing such a partition, where $\omega < 2.376$ is the exponent of fast matrix multiplication. The algorithm relies on a spectral characterization of vertex partitions satisfying the properties of the Frieze-Kannan regularity lemma.

Even Cycles and Quasi-Random Tournaments. Chung and Graham in [22] had provided several equivalent characterizations of quasi-randomness in tournaments. One of them is about the number of *even cycles*, where even is defined in the following sense. A cycle $C = \{v_1, v_2, \ldots, v_1\}$ in a tournament T is said to be *even*, if when walking along C, an even number of edges point in the wrong direction, that is, they are directed from v_{i+1} to v_i. Chung and Graham [22] showed that if close to half of the 4-cycles in a tournament T are even, then T is quasi-random. They asked if the same statement is true if instead of 4-cycles, we consider k-cycles, for an even k. We resolve this open question by showing that for every fixed even integer $k \geq 4$, if close to half of the k-cycles in a tournament T are even, then T must be quasi-random.

A Wowzer type lower bound for the strong regularity lemma. The regularity lemma of Szemerédi asserts that one can partition every graph into a bounded number of quasi-random bipartite graphs. In some applications however, one would like to have a strong control on how quasi-random these bipartite graphs are. Alon, Fischer, Krivelevich and Szegedy [6] obtained a variant of the regularity lemma, that allows one to have an *arbitrary* control on this measure of quasi-randomness. However, their proof only guaranteed to produce a partition where the number of parts is given by the Wowzer function, which is the iterated version of the Tower function. We show here that a bound of this type is unavoidable by constructing a graph H, with the property that even if one wants a very mild control on the quasi-randomness of a regular partition, then any such partition of H must have a number of parts given by a Wowzer-type function.

How fast can we deterministically simulate nondeterminism? We study an approach towards derandomizing complexity classes using Turing machine simulations. We look at the problem of deterministically counting the exact number of accepting computation paths of a given nondeterministic Turing machine. We provide a deterministic algorithm, which runs in time roughly $\tilde{O}(\sqrt{S})$, where S is the size of the configuration graph. The best of the previously known methods required time linear in S. Our result implies a simulation of probabilistic time classes like PP, BPP and BQP in the same running time. This is an improvement over the currently best known simulation by van Melkebeek and Santhanam [103].

CHAPTER I

INTRODUCTION

Randomness holds an important place in theoretical computer science (TCS). Randomized algorithms, probabilistic analysis, probabilistic complexity classes, and probabilistically checkable proofs are just a few of the areas where we make use of randomness in a crucially important manner. Many of the recent major discoveries in TCS could be attributed to the use of randomness.

One of the most compelling reasons that randomness has been so useful in TCS is the existence of several problems that have an efficient randomized polynomial time algorithm, but not a deterministic one. What would have been a good example 10 years ago is not anymore – PRIMES. This is the problem of testing if a given integer is prime. The randomized Miller-Rabin primality test [81] was discovered in the seventies, and for more than two decades that followed, there was no deterministic test that ran in polynomial time. The deterministic algorithm [1] for PRIMES was discovered only in 2002. Examples of problems where randomness is helpful in getting an efficient algorithm are the DeMillo-Lipton-Schwartz-Zippel polynomial identity testing [32, 90, 106] (in fact, there is strong evidence that it is hard to hope for a deterministic polynomial identity testing algorithm [54]) and volume estimation by Dyer-Frieze-Kannan [34] as well as several approximate counting problems. The P vs. NP question is the foremost open question in TCS and tries to characterize which problems can or cannot be solved efficiently. While P vs. NP remains open, one can never rule out the possibility of deterministic polynomial time algorithms for these problems, but for now randomness seems to be helpful.

Theoretical computer scientists have been attempting to understand the necessity of randomness. Randomness is considered expensive because it is hard to find a real source of randomness. Moreover, one can "boost" the probability of success of randomized algorithms by repeating it with independent random choices. For these boosting applications, being able to find *independent* random bits is helpful. There is a huge body of literature that tries to minimize the amount of randomness used (see the surveys [71, 72]). Ideally, one would like to *derandomize* a given randomized algorithm, i.e., to completely eliminate the need for randomness from the algorithm. If this is hard to achieve, then one would like to make do with as little randomness as possible. One successful approach for reducing the randomness is to use a string of bits that have some dependence between them. For example, an algorithm might require only pairwise independent (or k-wise independent) random bits instead of fully independent random variables.

Quasi-random sequences of bits are not random, in fact they are deterministic, but they possess statistical properties that make them usable, instead of pure random bits, in randomized algorithms. Certain statistical properties of quasi-random bits are identical to that of pure random bits. For instance, the area of *quasi-Monte Carlo methods* [75] makes use of quasi-random bits instead of random ones, thereby saving in randomness. *Expander graphs* have been very useful in generating quasi-random sequences of bits, which could be used to derandomize an algorithm. This application of expander graphs has been widely studied starting with the work of Ajtai, Komlos and Szemerédi [2] (see [51] for more details on the applications of expanders).

In this thesis, we shall study some aspects of quasi-randomness in combinatorial structures. This introductory chapter is organized as follows. In the next section, we discuss quasi-random graphs as introduced by Chung, Graham and Wilson [26] and quasi-randomness in other combinatorial objects. In several of these cases, quasi-randomness of a combinatorial object is also captured by a spectral characterization. We study these in the Section 1.1.3. The *Regularity Lemma*, proved by Szemerédi

[93], is a powerful tool that helps decompose graphs into components that are quasi-random. We discuss the regularity lemma and its variants in Section 1.2. Finally, in Section 1.3, we give an overview of our contributions in this thesis and explain the organization of the rest of the thesis.

1.1 Quasi-randomness in Graphs and other Combinatorial Objects

In this section, we study quasi-randomness in graphs and other combinatorial objects like groups, hypergraphs, directed graphs, etc. The main motivation is to study objects that are deterministic but have random-like properties. This is an informal notion that we shall formalize in this section.

1.1.1 Quasi-randomness in Graphs

There is a natural way to define a random graph on a vertex set of size n. Of the $\binom{n}{2}$ pairs of vertices, each pair of vertices is connected by an edge independently with a probability p for a given constant $0 < p < 1$. The family of graphs obtained in this manner is the Erdős-Rényi model of random graphs [35]. This family of graphs is denoted $G(n, p)$.

A quasi-random graph is one that behaves like a random graph from the family $G(n, p)$. To formalize this statement, we shall identify a set of properties that are all equivalent to one another and are shared by the random graph family $G(n, p)$. We shall term these properties as *quasi-random* and we shall call the graphs that satisfy any (and therefore, all) of these properties *quasi-random graphs*. Quasi-random graphs were first studied by Thomason [97, 98] (he called them *jumbled graphs*) and these notions were made more concrete by Chung, Graham and Wilson [26].

Thomason noted that one of the most important characteristics of a truly random graph is its *edge-density*. For a graph $G = (V, E)$, let $U \subseteq V$. Then let $e(U)$ denote the number of edges with both endpoints in U. The edge density of U is given by

$$e(U)/\binom{|U|}{2} .$$

One characterization of quasi-random graphs is that the edge density of any large enough set U is close to p. We make this concrete using the following definition.

Definition 1.1 (Quasi-random graphs). *A graph $G = (V, E)$ is quasi-random if for all subsets $U \subseteq V$, we have*

$$\left| e(U) - p\binom{|U|}{2} \right| = o(n^2) , \tag{1}$$

where $n = |V|$ and $e(U)$ denotes the number of edges that are contained in U.

Consider $U, W \subseteq V$, and let $e(U, W)$ denote the number of edges having one endpoint in each of U and W, counting the edges contained in $U \cap W$ twice. In fact, the above definition is equivalent to the following: In a quasi-random graph $G = (V, E)$, for all sets $U, W \subseteq V$, we have

$$|e(U, W) - p|U||W|| = o(n^2) . \tag{2}$$

If (2) is true for all $U, W \subseteq V$, then G is quasi-random because we can set $U = W$. If G is quasi-random by Definition 1.1, then one can break down (2) and derive that (2) should be true for all $U, W \subseteq V$.

In [26], Chung, Graham and Wilson showed that many other properties of different natures were equivalent to the above property. Before stating their main theorem, let us introduce some notation. Let $G = (V, E)$ be a graph on n vertices. For a fixed graph L, let $N^*_G(L)$ denote the number of labeled induced copies of L in G, and let $N_G(L)$ denote the number of labeled but not necessarily induced copies of L in G. For a pair of vertices $x, y \in V(G)$, let $s(x, y)$ denote the number of vertices of G joined to x and y in the same way; either to both or to none. Let $\text{codeg}(x, y)$ denote the number of common neighbors of x and y in G. Finally let λ_i denote the eigenvalues of the adjacency matrix $A(G)$ of G ordered such that $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_n|$. The main theorem of [26] is the following:

Theorem 1.2 ([26])**.** *The following properties are equivalent:*

- $P_1(l)$*: For a fixed graph* L *on* $l \geq 4$ *vertices,*

$$N^*_G(L) = (1 + o(1))n^l p^{|E(L)|}(1-p)^{\binom{l}{2}-|E(L)|} \ .$$

- $P_2(t)$*: Let* C_t *denote the cycle of length* t*. Let* $t \geq 4$ *be even. Then,*

$$e(G) = \left(\frac{n^2 p}{2} + o(n^2)\right) \quad \text{and} \quad N_G(C_t) \leq (np)^t + o(n^t) \ .$$

- $P_3 :\ \ e(G) \geq \frac{n^2 p}{2} + o(n^2), \quad \lambda_1 = (1 + o(1))np, \quad |\lambda_2| = o(n).$

- P_4*: For each subset* $U \subseteq V(G)$, $e(U) = \frac{p}{2}|U|^2 + o(n^2)$.

- P_5*: For each subset* $U \subseteq V(G)$ *such that* $|U| = \lfloor n/2 \rfloor$*, we have* $e(U) = \left(\frac{p}{8} + o(1)\right) n^2$.

- $P_6 :\ \ \sum_{x,y \in V} |s(x,y) - (p^2 + (1-p)^2)n| = o(n^3) \ .$

- $P_7 :\ \ \sum_{x,y \in V} |codeg(x,y) - p^2 n| = o(n^3) \ .$

Notice that the properties P_1 through P_7 form a very diverse set, yet all of them could be easily verified to hold true for truly random graphs chosen from $G(n,p)$. The property $P_1(t)$ requires that the number of induced labeled copies of a given graph of size t occurs roughly the expected number of times in G. Property P_4 is just a restatement of Definition 1.1 and so we can immediately conclude that all of the above properties are equivalent definitions/characterizations of quasi-random graphs. As we observed in (2), we can add one more equivalent property to the above list.

- P'_4*: For each pair of subsets* $U, W \subseteq V(G)$, $e(U,W) = p|U||W| + o(n^2)$.

It is important to note that each of these properties are not only implied by the graph being quasi-random but that each of them form a characterization of quasi-randomness. That is, to check if a graph is quasi-random it is enough to test the graph for any one of these properties, whichever one turns out to be convenient to test.

It is notable that the property $P_2(4)$ only requires that the total number of edges in the graph and the total number of labeled copies of 4-cycles are roughly what we expect to see in a member of $G(n,p)$. It is a seemingly weak condition, but is still powerful enough to capture the notion of quasi-randomness. The property P_3 is a condition on the eigenvalues of the adjacency matrix of G. We note that this is a very interesting connection. This is representative of the spectral characterization of quasi-randomness in several combinatorial objects. We shall see this in greater detail in Section 1.1.3, and this is a key tool that we shall be using in this thesis.

Until now, we have seen different properties of quasi-random graphs. It can be easily verified that these properties are true (with high probability) for a member of $G(n,p)$. To distinguish a truly random graph and a quasi-random graph, we provide an example from [26]. The example is a deterministic graph called Paley graph, denoted by Q_n. It is defined for a prime $n \equiv 1 \pmod 4$, and has n vertices. Vertices i and j form an edge of Q_n if and only if $i-j$ is a quadratic residue of n. Using basic modular arithmetic and quadratic reciprocity, it can be verified that Q_n is a $(n-1)/2$ regular graph. We can also check that for distinct x, y, we have $s(x,y) = (n-3)/2$, hence Q_n satisfies property P_6. Hence Q_n is quasi-random, with the probability $p = 1/2$.

However, the size of the largest clique of Q_n has been found to be as large as $c \log n \log\log\log n$ for infinitely many primes n. But the expected size of the largest clique of a graph from $G(n,1/2)$ is $(1+o(1))\frac{\log n}{\log 2}$ [17]. Thus we note that Q_n deviates from the random graph family $G(n,1/2)$ in this aspect.

What we have not described yet is the case when p is sub-constant, when the graph G is sparse. Quasi-randomness in sparse graphs was studied in [25]. Three of the equivalent characterizations of a *sparse quasi-random graph* are given in the following theorem:

Theorem 1.3 (Sparse Quasi-Random Graphs[25])**.** *Suppose for some constant $c > 0$, $p(n) > cn^{-1+\frac{1}{t-1}}$, where $t \geq 2$. For any family of graphs $G = (V, E)$, $|E(G)| = (1+o(1))p\binom{n}{2}$, then subject to a technical condition*[1]*, the following equivalent properties capture quasi-randomness in G.*

1. *The number of labeled $2t$-cycles is given by $(1 + o(1))(np)^t$.*
2. *The eigenvalues λ_i of $A(G)$ satisfy $\lambda_1 = (1 + o(1))np$ and $|\lambda_2| = o(np)$.*
3. *For all $U, W \subseteq V$, $|e(U, W) - p|U||W|| = o(pn^2)$.*

In Theorem 1.2, we saw properties that characterize quasi-random graphs when p is a constant. We note that the properties in Theorem 1.3 are generalizations of properties in Theorem 1.2. The property 1 in Theorem 1.3 is the property $P_2(t)$ in Theorem 1.2 adapted to the case when G maybe sparse, that is when p maybe sub-constant. Similarly, properties 2 and 3 in Theorem 1.3 are modifications of P_3 and P_4' respectively. So even though not all of the properties of Theorem 1.2 generalize to the case when p maybe sub-constant, some of the properties indeed do. For more details on quasi-random graphs, we refer the reader to the survey of Krivelevich and Sudakov [65].

1.1.2 Quasi-randomness in other Combinatorial Objects

Consider a combinatorial object, for example a k-uniform hypergraph. A *k-uniform hypergraph* $G = (V, E)$ is a set of vertices V and a set of k-tuples $E \subseteq \{(v_1, \ldots, v_k)$:

[1] For the sake of simplicity, we omit the technical condition.

$v_1, \ldots, v_k \in V\}$. Notice that when $k = 2$, this is the definition of graphs. Like we saw in the case of graphs, there is a natural way to define a random k-uniform hypergraph – each possible k-tuple $(v_1, \ldots, v_k)$ is selected to be in E with probability p.

For a random 3-uniform hypergraph G on n vertices, the expected number of induced labeled copies of a given k-uniform hypergraph L of size l is $n^l p^{|E(L)|}(1 - p)^{\binom{l}{k} - |E(L)|}$. So with high probability, there would be $(1+o(1))n^l p^{|E(L)|}(1-p)^{\binom{l}{k}-|E(L)|}$ induced labeled copies of L in G. As in the case of graphs, this turns out to be one of the several equivalent characterizations of *quasi-random k-uniform hypergraphs* [23, 24]. The other characterizations of quasi-random hypergraphs are generalizations of the characterizations in Theorem 1.2. For further details and discussions, we refer the reader to the excellent surveys by Gowers [43] and Trevisan [100].

In a similar manner, quasi-randomness has been defined and studied for several other combinatorial objects. Some of the studied objects are set systems [21], tournaments [22], groups [45] and directed graphs [48]. In Chapter 2, we shall study quasi-random tournaments in some detail, and provide new characterizations for them, including a spectral characterization.

1.1.3 Expander Graphs and Spectral Characterizations of Quasi-randomness

Let us recall the properties based on eigenvalues from Theorems 1.2 and 1.3. These state that $\lambda_1 = (1 + o(1))np$ and $|\lambda_2| = o(np)$. The condition requires that the first eigenvalue λ_1 is close to np and the rest of the eigenvalues are small. This is a spectral characterization of quasi-randomness in graphs. We shall see that the spectrum captures the quasi-random properties of several combinatorial objects.

Expander graphs are a very good example for a class of sparse quasi-random graphs. They are typically d-regular graphs for a constant d, which means that they have $O(n)$ edges. However, they are very well connected, which is a consequence of their definition.

Definition 1.4 (Expander Graphs)**.** *For a graph G, we define* edge expansion*, denoted by $h(G)$, as follows:*

$$h(G) = \min_{0<|S|\leq \frac{n}{2}} \frac{|e(S,\bar{S})|}{|S|} \ , \tag{3}$$

where $e(S,\bar{S})$ denotes the number of edges from S to its complement $\bar{S}$.

G is an expander graph if $h(G) \geq \varepsilon$ for a fixed constant $\varepsilon > 0$.

Every set S of less than $n/2$ vertices is connected to at least $h(G)|S|$ more vertices. This ensures that the graph is well connected, despite being sparse. A consequence of this is that the diameter of an expander is $O(\log n)$. Random walks on an expander graph require much less random bits, because of the low degree. Hence expanders are used in derandomization.

The Cheeger's inequality relates the expansion of a graph to its eigenvalues.

Theorem 1.5 (Cheeger's Inequality [3, 20])**.** *Let G be a d-regular graph with eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$. Then*

$$\frac{d-|\lambda_2|}{2} \leq h(G) \leq \sqrt{2d(d-|\lambda_2|)} \ . \tag{4}$$

This helps us form a spectral characterization of expander graphs. For a d-regular graph to be an expander, the gap $d - |\lambda_2|$ should be an absolute constant which is strictly positive. This already gives us an indication that an expander graph is quasi-random, by the eigenvalue characterization in Theorem 1.3. In fact, given two sets $U, W \subseteq V$ of a d-regular expander $G = (V, E)$, the *expander mixing lemma* states that

$$\left| |E(U,W)| - \frac{d|U||W|}{n} \right| \leq |\lambda_2| \sqrt{|U||W|} \, , \tag{5}$$

thereby relating the edge density to the expansion as well. Thus when $|\lambda_2|$ is small, the expander graph is quasi-random. Note that the two terms on the left hand side are the actual number of edges from U to W and the expected number of edges from U to W in a random graph. This connection is called the expander mixing lemma because this provides a direct connection between the spectrum of the graph, and how quickly a random walk on the graph is likely to "mix". For more details about expanders, the reader is referred to the excellent survey by Hoory, Linial and Wigderson [51].

The spectral connection to quasi-randomness has been widely studied and used in different combinatorial objects [4, 19, 61]. In fact, it is a technique that we use twice in this thesis. In Chapter 2, we develop a spectral characterization of quasi-random tournaments and use it to prove some new results. In Chapter 3, we develop a spectral characterization of Frieze-Kannan regularity and use it towards a deterministic algorithm for finding an FK-regular partition of a graph.

1.2 The Regularity Lemma

The regularity lemma of Szemerédi [93] is one of the most widely used tools in extremal combinatorics. The lemma was originally devised as part of Szemerédi's proof of his (eponymous) theorem [92] on arithmetic progressions in dense sets of integers. Since then it has turned into a fundamental tool in extremal combinatorics, with applications in diverse areas such as theoretical computer science, additive number theory, discrete geometry and of course graph theory. We refer the reader to the survey by Komlos et. al. [64] and its references for more details on the rich history and applications of the regularity lemma.

1.2.1 Szemerédi's Regularity

Szemerédi's regularity lemma roughly states that every dense graph can be approximated by a union of induced quasi-random bipartite graphs. The regularity lemma helps us use quasi-randomness in analyzing an arbitrary dense graph. It also allows us to use probabilistic intuition to problems that are deterministic in nature.

In order to describe the regularity lemma more formally, let us set up some notation. For a pair of subsets $A, B \subseteq V(G)$ in a graph $G = (V, E)$, let $e(A, B)$ denote the number of edges between A and B, counting each of the edges contained in $A \cap B$

twice. The density $d(A, B)$ is defined to be $d(A, B) = \frac{e(A,B)}{|A||B|}$. We will frequently deal with a partition of the vertex set $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$. The *order* of such a partition is the number of sets V_i (k in the above partition). A partition is *equitable* if all sets are of size $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$. We will make use of the shorthand notation for density across parts, $d_{ij} = d(V_i, V_j)$ whenever $i \neq j$. Also, we set $d_{ii} = 0$ for all i.

The key notion in Szemerédi's regularity lemma [93] is the notion of ε-regularity, as defined below:

Definition 1.6 (ε-regular)**.** *Let A, B be disjoint sets of vertices of G. We say that (A, B) is* ε-regular *if $|d(A, B) - d(A', B')| \leq \varepsilon$ for all $A' \subseteq A$ and $B' \subseteq B$ satisfying $|A'| \geq \varepsilon|A|$ and $|B'| \geq \varepsilon|B|$.*

A partition $\mathcal{P} = \{V_1, \ldots, V_k\}$ of V is called a ε-regular partition *if it is equitable, and all but εk^2 of the pairs (i, j) are such that (V_i, V_j) is ε-regular.*

It is not hard to see that ε-regular bipartite graphs are quasi-random. Szemerédi's Regularity Lemma states the following:

Theorem 1.7 (Szemerédi's Regularity Lemma [93])**.** *Given $\varepsilon > 0$ there is a constant $S(\varepsilon)$, such that the vertex set of any graph $G = (V, E)$ can be partitioned into $k \leq S(\varepsilon)$ sets $\mathcal{P} = \{V_1, \ldots, V_k\}$, such that $\mathcal{P}$ is ε-FK-regular.*

One of the useful aspects of an ε-regular partition of a graph is that it allows one to estimate the number of edges in certain partitions of G. For example, given an ε-regular partition, one can estimate the value of the Max-Cut in G within an error of εn^2, in time that depends only on the order of the partition (and independent of the order of G!). Hence, one can think of Szemerédi's regularity lemma as saying that any graph can be approximated by a constant sized graph. This aspect of the regularity lemma has turned out to be extremely useful for designing approximation algorithms.

1.2.2 Frieze-Kannan Regularity

The main drawback of Szemerédi's regularity lemma is that the constants involved are huge; Gowers [42] proved that in some cases the number of parts in a Szemerédi regular partition grows as a tower of exponents of height polynomial in $1/\varepsilon$, where ε is the parameter for regularity. It is thus natural to ask if one can find a slightly weaker regularity lemma that would be applicable, while at the same time not involve such huge constants. Such a lemma was indeed considered in [92] for bipartite graphs and in [33] for arbitrary graphs. Subsequently, Frieze and Kannan [38, 39] devised an elegant regularity lemma of this type. They formulated a slightly weaker notion of regularity that we will refer to as FK-regularity. They proved that any graph has an FK-regular partition involving drastically fewer parts compared to Szemerédi's lemma. They also showed that an FK-regular partition can still be used in some of the cases where Szemerédi's lemma was previously used. The notion of FK-regularity has been investigated extensively in the past decade. For example, it is a key part of the theory of graph limits developed in recent years, see the survey of Lovász [67]. Finally, FK-regularity was a key tool in the recent breakthrough of Bansal and Williams [12], where they obtained new bounds for combinatorial boolean matrix multiplication.

While all variants of Szemerédi's regularity lemma attempt to approximate a given dense graph using a number of quasi-random bipartite graphs, they vary in the manner in which they approximate the graph. We use the same notation that we used while describing Szemerédi's regularity.

Definition 1.8 (ε-FK-regular)**.** *Let $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ be a partition of $V(G)$. For subsets $S, T \subseteq V$ and $1 \le i \le k$, let $S_i = S \cap V_i$ and $T_i = T \cap V_i$. Define $\Delta(S,T)$ for subsets $S, T \subseteq V$ as follows:*

$$\Delta(S,T) = e(S,T) - \sum_{i \neq j} d_{ij} |S_i||T_j|. \tag{6}$$

The partition $\mathcal{P}$ is said to be ε-FK-regular if it is equitable and

$$\textit{for all subsets } S, T \subseteq V, \;\; |\Delta(S,T)| \le \varepsilon n^2. \tag{7}$$

If $|\Delta(S,T)| > \varepsilon n^2$ then S, T are said to be witnesses to the fact that $\mathcal{P}$ is not ε-FK-regular.

One can think of Szemerédi's regularity as dividing the graph into parts such that across most of the parts the graph looks like a random graph. In FK-regularity, we just want to partition the graph so that any cut of the graph contains roughly the expected number of edges as dictated by the densities d_{ij}. Another way to think about FK-regularity is that we want the bipartite graphs to be ε-regular (in the sense of Szemerédi) only on average.

Theorem 1.9 (Frieze-Kannan Regularity Lemma [38, 39])**.** *Given $\varepsilon > 0$ there is a constant $S_{FK}(\varepsilon)$, such that the vertex set of any graph $G = (V, E)$ can be partitioned into $k \le S_{FK}(\varepsilon)$ sets $\mathcal{P} = \{V_1, \ldots, V_k\}$, such that $\mathcal{P}$ is ε-FK-regular.*

Like we mentioned before, the main novelty in this (weaker[2]) notion of regularity is that it allows one to compute useful statistics on the graph (such as estimating Max-Cut) while at the same time having the property that any graph can be partitioned into an ε-FK-regular partition of order $2^{100/\varepsilon^2}$, which is drastically smaller than the tower-type order of a Szemerédi partition.

1.2.3 Strong Regularity

One feature of Szemerédi's regularity is that the measure of quasi-randomness (i.e., ε) remains independent of the order of the partition considered. As we mentioned before, in a breakthrough result, Gowers [42] proved that for any $\varepsilon > 0$, there exists

[2] It is not hard to see that an ε-regular partition (in the sense of Szemerédi's lemma) is indeed ε-FK-regular.

a graph where any ε-regular partition must have size at least $T(1/\varepsilon^{1/16})$, where $T(x)$ denotes a tower of twos of height x.

Gowers' lower bound can be stated as saying that if one wants a regular partition of order k, then the best quasi-randomness measure one can hope to obtain is merely $1/\log^*(k)$. Suppose however that for some $f : \mathbb{N} \mapsto (0,1)$, we would like to find a partition of a graph of order k that will be "close" to being $f(k)$-regular. Alon, Fischer, Krivelevich and Szegedy [6] formulated the following notion of being close to $f(k)$-regular.

Definition 1.10 ((ε, f)-regular partition)**.** *Let f be a function $f : \mathbb{N} \mapsto (0,1)$. An (ε, f)-regular partition of a graph G is a pair of partitions $\mathcal{A} = \{V_i \ : \ 1 \leq i \leq k\}$ and $\mathcal{B} = \{U_{i,i'} \ : \ 1 \leq i \leq k, 1 \leq i' \leq \ell\}$ of G, where $\mathcal{B}$ is a refinement of $\mathcal{A}$ and the following two conditions hold:*

1. *$\mathcal{B}$ is $f(k)$-regular (as in Definition 1.6).*
2. *Say that a pair (V_i, V_j) of clusters of $\mathcal{A}$ is* good *if all but at most εl^2 of pairs $1 \leq i', j' \leq \ell$ satisfy $|d(U_{i,i'}, U_{j,j'}) - d(V_i, V_j)| < \varepsilon$. Then, at least $(1-\varepsilon)\binom{k}{2}$ of the pairs are good.*

One useful way of thinking about the above notion is to "forget" for a moment about the partition $\mathcal{B}$ and just treat partition $\mathcal{A}$ as an $f(k)$-regular partition. One then tries to extract some useful information from the assumption that $\mathcal{A}$ itself is $f(k)$-regular. Finally, one uses the second property of Definition 1.10, which says that the two partitions are *similar*, in order to show that the information deduced from the *assumption* that $\mathcal{A}$ is $f(k)$-regular can actually be deduced from the *fact* that $\mathcal{B}$ is $f(k)$-regular.

One of the main results of [6] was that given a graph G and *any* function f, one can construct an (ε, f)-regular partition of G of bounded size. This version of the regularity lemma is sometimes referred to as the *strong regularity lemma*.

Theorem 1.11 (Strong Regularity Lemma [6]). *For every $\varepsilon > 0$ and $f : \mathbb{N} \mapsto (0,1)$, there is an integer $S = S_{AFKS}(\varepsilon, f)$ such that any graph $G = (V, E)$ has an (ε, f)-regular partition $(\mathcal{A}, \mathcal{B})$ where $1/\varepsilon \leq |\mathcal{A}|, |\mathcal{B}| \leq S$.*

Let us describe two cases where one needs to have a better control of the measure of quasi-randomness of a regular partition. A first example is when proving certain variant of the graph removal lemma [86]. In such a scenario we are given a regular partition and would like to be able to say that since the partition behaves in a quasi-random way, then we can find "small" subgraphs that we expect to find in a truly random graph. The only problem is that as the "small" structure we are trying to find becomes larger, we need the measure of quasi-randomness to decrease with it. Some examples where Theorem 1.11 was used to overcome such difficulties can be found in [6, 8, 10, 11, 62, 82]. We note that in some of these papers, Theorem 1.11 was used with functions f that go to zero extremely fast, so the ability to apply the theorem with arbitrary functions was crucial.

Another example when one wants a better control of the measure of quasi-randomness is when the graph we are trying to partition is very sparse. It is not hard to see that for the notion of ε-regularity to make sense, the graph we are trying to partition should have density at least ε. A well known case where one is faced with increasingly sparse graphs is in the proofs of the hypergraph regularity lemma, that were obtained independently by Gowers [43] and by Rödl et al. [37, 73, 84] and later also by Tao [94]. In those proofs, one is partitioning not only the vertices of the hypergraph (as in Theorem 1.7) but also the pairs of vertices into quasi-random bipartite graphs. However, in the process these bipartite graphs become sparser so one needs to control their quasi-randomness as a function of their density. See the survey of Gowers [43] for an excellent account of this issue.

We finally note that the strong regularity lemma is also related to the notion of a limit of convergent graph sequences defined and studied in [18]. Without defining these notions explicitly, we just mention that many of the results mentioned above that were proved using Theorem 1.11, were later reproved using graph limits, see e.g. Lovász and Szegedy [70]. Furthermore, some of the important properties of the limit of a convergent graph sequence, such as its uniqueness [68], also hold for (ε, f)-regular partitions, see [10]. Hence, one can view an (ε, f)-regular partition as the discrete analogue of the (analytic) limit of a convergent graph sequence.

1.2.4 Discussions about Regularity

The regularity lemmas are a fundamental tool in graph theory and combinatorics. The proofs of the regularity lemmas follow the structure vs. randomness paradigm [95]. The structure vs. randomness paradigm states that a given object (examples are functions, sets, graphs, vectors etc.) can be decomposed into a structured component and a component that is quasi-random, up to some error. A partition given by the regularity lemma has a structured component, the underlying density graph that is constant size, and a random component, the quasi-random bipartite graphs connecting each of the parts.

The proofs of the regularity lemmas follow a similar path. They start off with an arbitrary partition, which may or may not be regular. This partition, if it is not regular, can be refined further to obtain a new partition. This refinement can be viewed as adding to the structured component. This is repeated multiple times to reach a regular partition. The evidence that we are indeed progressing towards a desired decomposition is a potential function that increases by at least a fixed constant on each iteration. Though there are different variants of the regularity lemma, as we have already seen, the proofs of all of them follow the same pattern. For more details on the variants and proofs of the regularity lemmas, we refer the reader to the survey by Rödl and Schacht [83]. In addition to proving regularity lemmas in graphs, this proof method has been used in proving regularity lemmas in other objects as well;

the regularity lemmas on groups [47] and permutations [28] being notable examples.

It is not too hard to order the regularity lemmas covered in the previous sections in the order of their strengths – the Frieze-Kannan regularity is weaker than the Szemerédi regularity, which is in turn weaker than the strong regularity. However, it is quite interesting to note that the Frieze-Kannan regularity lemma can be iterated repeatedly to derive the Szemerédi regularity lemma (see [83]). Also the Szemerédi regularity lemma can be iterated to obtain the strong regularity lemma.

Finally, we point out another connection between Szemerédi's regularity and quasi-randomness in graphs. Simonovits and Sós [91] noted that a graph G is quasi-random with edge density p (as in Definition 1.1) if and only if almost all bipartite graphs formed in the Szemerédi regularity partitions of G are quasi-random with density $p + o(1)$.

1.3 Our Contributions and Thesis Organization

As we have already seen, quasi-randomness and regularity have been studied extensively. It would be interesting to generalize the spectral connection towards quasi-randomness and have universal properties that characterize quasi-randomness in the case of different combinatorial objects. What one would require is, in the case of each combinatorial object, a suitable model for quasi-randomness that would help obtain the spectral characterization. As we saw in Section 1.1.3, there are several examples of such a characterization already. In Chapter 2, we show progress in this direction by providing a spectral characterization for quasi-random tournaments (as defined by Chung and Graham [22]) and quasi-random directed graphs (as defined by Griffiths [48]). Such a characterization turns out to be very useful because it helps us extend one of the characterizations of quasi-random tournaments, thereby answering an open question asked by Chung and Graham in [22]. This work is joint with Asaf Shapira, and originally appeared in [57].

The regularity lemmas are very useful and applicable in the area of combinatorics. So it is quite relevant to see the limits of their usefulness. In order to apply a regularity lemma in an algorithm, one needs to actually find the regular partition. So we need an algorithmic version of the regularity lemma. In the case of Frieze-Kannan regularity, we obtain a deterministic algorithm that runs in $\tilde{O}(n^\omega)$ time in Chapter 3 of this thesis. We develop a spectral characterization of FK-regularity and this characterization is used in getting the deterministic algorithm. Our algorithm is the first deterministic algorithm that runs in sub-cubic time, and the spectral characterization was hitherto unknown for FK-regularity. This is joint work with Domingos Dellamonica, Daniel Martin, Vojtěch Rödl and Asaf Shapira. This appeared in the *Proceedings of APPROX/RANDOM 2011* [31]. The full version has been accepted for publication in the *SIAM Journal on Discrete Math* and is yet to appear.

One important aspect of applying the regularity lemmas is the number of parts required in a partition that satisfies the condition of the lemma. Several fundamental results applied Szemerédi's regularity lemma [85, 86] and the original proof indicated that the number of parts required in a regularity partition might be a tower of exponents, where the height of the tower depends on the measure of regularity, usually denoted by ε. As we have already mentioned, Gowers [42] proved that a tower type dependence is unavoidable. In Chapter 4 of this thesis, we provide a lower bound for the number of parts required by a partition that satisfies the conditions of the strong regularity lemma by Alon, Fischer, Krivelevich and Szegedy (Theorem 1.11). The bound that we provide is a Wowzer type bound, which is the tower function iterated multiple times. Wowzer type functions are one level higher in the Ackermann hierarchy than the tower functions. Our result is the first[3] such lower bound for the strong regularity lemma. This is joint work with Asaf Shapira [58].

[3] After completing our work, we learned that Conlon and Fox [27] have independently (and simultaneously) obtained a result similar to ours.

Finally, in Chapter 5 of this thesis, we study a different approach towards the derandomization of complexity classes. Though it is not directly connected towards quasi-randomness and regularity, we think it is relevant because it is a novel approach that could potentially be helpful in other problems. We study the problem of deterministically counting the number of accepting computations of a nondeterministic Turing machine. We obtain an algorithm which is a *square-root* improvement over what is currently known. This implies a faster deterministic simulation of the class #P, and probabilistic classes PP, BPP and BQP. This chapter is a result of joint work with Richard Lipton, Kenneth Regan and Farbod Shokrieh [55, 56]. Part of the work [55] appeared in the journal *Theoretical Computer Science*. A preliminary version appeared in *MFCS 2010: Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*.

CHAPTER II

EVEN CYCLES AND QUASI-RANDOM TOURNAMENTS

2.1 Introduction

As we have already seen, quasi-random objects are *deterministic* objects that possess the properties we expect truly *random* ones to have. One of the most surprising phenomena in this area is the fact that in many cases, if an object satisfies a single *deterministic* property then it must "behave" like a typical random object in many useful aspects. In this chapter we study one such phenomenon related to quasi-random tournaments. The notion of quasi-randomness has been widely studied for different combinatorial objects, like graphs, hypergraphs, groups and set systems [21, 24, 26, 45]. In this chapter, we show that for every fixed even integer $k \geq 4$, if close to half of the k-cycles in a tournament T are even, then T must be quasi-random. This resolves an open question raised in 1991 by Chung and Graham [22].

A directed graph $D = (V, E)$ consists of a set of vertices and a set of directed edges $E \subseteq V \times V$. We use the ordered pair $(u, v) \in V \times V$ to denote directed edge from u to v. A tournament $T = (V, E)$ is a directed graph such that given any two distinct vertices $u, v \in V$, there exists exactly one of the two directed edges (u, v) or (v, u) in $E(T)$. There are no loops, i.e. directed edges of the form (u, u), in a tournament. One can also think of a tournament as an orientation of an underlying complete graph on V. We shall use n to denote $|V|$.

Consider a tournament $T = (V, E)$. For $Y \subseteq V$, and $v \in V$, let $d^+(v, Y)$ denote the number of directed edges going from v to Y and $d^-(v, Y)$ denote the number of directed edges going from Y to v. A purely random tournament is one where for each pair of distinct vertices u and v of V, the directed edge between them is chosen randomly to be either (u, v) or (v, u) with probability 1/2. It is not too hard to observe that in a random tournament T, with high probability, we have $\sum_{v \in X} |d^+(v, Y) - d^-(v, Y)| = o(n^2)$ for all $X, Y \subseteq V(T)$. If there exists $X, Y \subseteq V(T)$ such that $\sum_{v \in X} |d^+(v, Y) - d^-(v, Y)| = cn^2$, for some constant $c > 0$, then we can get sets $X', Y' \subseteq V(T)$ such that $c'n^2$ directed edges are oriented from X' to Y'. With high probability, this cannot happen in a random tournament. Let us define the corresponding property $\mathcal{Q}$ as follows:

Definition 2.1. *A tournament T on n vertices satisfies property $\mathcal{Q}$ if*

$$\sum_{v \in X} \left| d^+(v, Y) - d^-(v, Y) \right| = o(n^2) \quad \textit{for all } X, Y \subseteq V(T).$$

The notion of quasi-randomness in tournaments was introduced by Chung and Graham [22]. They defined several properties of tournaments, all of which are satisfied by purely random tournaments, including the property $\mathcal{Q}$ above. They also showed that all these properties are equivalent, namely, if a tournament satisfies one of these properties, then it must also satisfy all the other. They then defined a tournament to be quasi-random if it satisfies any (and therefore, all) of these properties. For the sake of clarity, we will focus on property $\mathcal{Q}$ (defined above) that will turn out to be the easiest one to work with in the current context.

Another property studied in [22] was related to even cycles in tournaments. A k-cycle is an ordered sequence of vertices $(v_1, v_2, \dots, v_k, v_1)$ such that no vertex is repeated immediately in the sequence. That is, $v_i \neq v_{i+1}$ for all $i \leq k - 1$ and $v_k \neq v_1$. We say that a k-cycle (for an integer $k \geq 2$) is *even* if as we traverse the cycle, we see an even number of directed edges opposite to the direction of the traversal. If a k-cycle is not even, we call it *odd*. Let $\mathsf{E}_k(T)$ denote the number of even k-cycles in a tournament T. Clearly, the number of k-cycles in an n-vertex tournament is $n^k - o(n^k)$. In fact, it can be shown that that the exact number is given by $(n-1)^k + (-1)^k(n-1)$ (see Section 2.3.2). In a random tournament, we expect about half of the k-cycles to be even. This motivated Chung and Graham [22] to define the following property:

Definition 2.2. *A tournament T on n vertices satisfies*[1] *property $\mathcal{P}(k)$ if $\mathsf{E}_k(T) = (1/2 \pm o(1))n^k$.*

Notice that when k is an odd integer, $\mathsf{E}_k(T)$ is *exactly* half the number of k-cycles in T, since an even cycle becomes odd upon traversal in the reverse direction. Hence, property $\mathcal{P}(k)$ cannot be equivalent to property $\mathcal{Q}$ when k is odd.

In [22] Chung and Graham show that $\mathcal{P}(4)$ is quasi-random. In other words, a tournament has (approximately) the correct number of even 4-cycles we expect to find in a random tournament, if and only if it satisfies property $\mathcal{Q}$. A question that was left open in [22] was whether $\mathcal{P}(k)$ is equivalent to $\mathcal{Q}$ for all even $k \geq 4$. One motivating reason for this question is the fact that we simply expect the property $\mathcal{P}(k)$ to be true for all even $k \geq 4$. A deeper reason is that in the definition of quasi-random graphs by Chung, Graham and Wilson [26](as we saw in Section 1.1.1), one of the characterizations of quasi-randomness depends only on the number of k-length cycles for a given even integer $k \geq 4$. Our main result answers their question positively by proving the following:

Theorem 2.3. *The following holds for every fixed even integer $k \geq 4$: A tournament satisfies property $\mathcal{Q}$ if and only if it satisfies property $\mathcal{P}(k)$.*

When we say that property $\mathcal{Q}$ implies property $\mathcal{P}(k)$ we mean that for every ε there is a $\delta = \delta(\varepsilon)$, such that any large enough tournament satisfying $\sum_{v \in X} |d^+(v, Y) - d^-(v, Y)| \leq \delta n^2$ for all X, Y has $(1/2 \pm \varepsilon)n^k$ even cycles. The meaning of $\mathcal{P}(k)$ implies $\mathcal{Q}$ is defined similarly.

[1] Observe that our definition of a k-cycle allows repeated vertices in the cycle. Note however, that forbidding repeated vertices (that is, requiring the k-cycles to be simple) would have resulted in the same property $\mathcal{P}(k)$ since the number of k-cycles with repeated vertices is $o(n^k)$. Allowing repeated vertices simplifies some of the notation.

2.2 Proof of Main Result

To prove Theorem 2.3, we shall go through a spectral characterization of quasi-randomness. We use the following adjacency matrix A to represent the tournament T. For every $u, v \in V$

$$A_{u,v} = \begin{cases} 1 & \text{if } (u,v) \in E(T) \\ -1 & \text{if } (v,u) \in E(T) \\ 0 & \text{if } u = v \end{cases}.$$

A key observation that we will use is that the matrix A is skew-symmetric. Recall that a real skew symmetric matrix can be diagonalized and all its eigenvalues are purely imaginary. It follows that all the eigenvalues of A^2 are non-positive. This implies the following claim, which will be crucial in our proof.

Claim 2.4. *For* $k \equiv 2 \pmod 4$*, all the eigenvalues of* A^k *are non-positive. For* $k \equiv 0 \pmod 4$*, all the eigenvalues of* A^k *are non-negative.*

For a matrix M, we let $\text{tr}(M) = \sum_{i=1}^{n} M_{i,i}$ denote the trace of the matrix M. Before we prove Lemmas 2.6 and 2.7, we make the following claim.

Claim 2.5. *Let* A *be the adjacency matrix of the tournament* T*. Then for an even integer* $k \geq 4$*, we have*

$$tr(A^k) = 2\mathsf{E}_k(T) - (n-1)^k - (n-1).$$

In particular, T *satisfies the property* $\mathcal{P}(k)$ *if and only if* $|tr(A^k)| = o(n^k)$*.*

Proof. Notice that the (u, u)th entry of A^k is the number of even k-cycles starting and ending at u minus the number of odd k-cycles starting and ending at u. So the sum of all diagonal entries, $\text{tr}(A^k)$, is the difference between all labeled even k-cycles and all labeled odd k-cycles. Recall that the total number of k-cycles is $(n-1)^k + (n-1)$ for even k. Thus we have that $\text{tr}(A^k) = 2\mathsf{E}_k(T) - (n-1)^k - (n-1)$.

We have $\mathrm{tr}(A^k) = 2\mathsf{E}_k(T) - n^k + o(n^k)$. Notice that T satisfies property $\mathcal{P}(k)$ when $\mathsf{E}_k(T) = (1/2 \pm o(1))n^k$, which happens if and only if $|\mathrm{tr}(A^k)| = o(n^k)$. □

We are now ready to prove the first direction of Theorem 2.3.

Lemma 2.6. *Let $k \geq 4$ be an even integer. If a tournament satisfies $\mathcal{P}(k)$ then it satisfies $\mathcal{Q}$.*

Proof. Let $\lambda_1(A), \ldots, \lambda_n(A)$ be the eigenvalues of A sorted by their absolute value, so that $\lambda_1(A)$ has the largest absolute value. We first claim that $|\lambda_1(A)| = o(n)$. Assume first that $k \equiv 0 \pmod 4$. Then by Claim 2.4 all the eigenvalues of A^k are non-negative, implying that

$$\mathrm{tr}(A^k) = \sum_{i=1}^{n} \lambda_i(A^k) \geq \lambda_1(A^k) = \lambda_1(A)^k \ . \tag{8}$$

Now, since we assume that T satisfies $\mathcal{P}(k)$, we get from Claim 2.5 that $|\mathrm{tr}(A^k)| = o(n^k)$. Equation (8) now implies that $|\lambda_1(A)| = o(n)$. A similar argument works when $k \equiv 2 \pmod 4$ only now all the terms in (8) would be non-positive.

We now claim that the fact that $|\lambda_1(A)| = o(n)$ implies that T satisfies $\mathcal{Q}$. Suppose it does not, and let $X, Y \subseteq V$ be two sets satisfying $\sum_{v \in X} |d^+(v,Y) - d^-(v,Y)| = cn^2$, for some $c > 0$. Let $\mathbf{y} \in \{0,1\}^n$ be the indicator vector for Y. We pick the vector $\mathbf{x}$ in the following way: if $v \notin X$, then set the corresponding coordinate $\mathbf{x}_v = 0$. For $v \in X$ such that $d^+(v,Y) - d^-(v,Y) \geq 0$, we set $\mathbf{x}_v = 1$. For all other $v \in X$, we set $\mathbf{x}_v = -1$. Now notice that for these vectors $\mathbf{x}$ and $\mathbf{y}$, we have $\mathbf{x}^T A \mathbf{y} = \sum_{v \in X} |d^+(v,Y) - d^-(v,Y)| = cn^2$. We can normalize $\mathbf{x}$ and $\mathbf{y}$ to get unit vectors $\tilde{\mathbf{x}} = \mathbf{x}/\sqrt{|X|}$ and $\tilde{\mathbf{y}} = \mathbf{y}/\sqrt{|Y|}$ satisfying

$$\tilde{\mathbf{x}}^T A \tilde{\mathbf{y}} = (\mathbf{x}^T A \mathbf{y})/\sqrt{|X||Y|} \geq cn^2/n = cn \ , \tag{9}$$

where the inequality follows since $|X|, |Y| \leq n$. We have thus found two unit vectors $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ such that $\tilde{\mathbf{x}}^T A \tilde{\mathbf{y}} \geq cn$.

We finish the proof by showing that (9) contradicts the fact that $|\lambda_1(A)| = o(n)$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the orthonormal eigenvectors corresponding to the eigenvalues of A. Let $\tilde{\mathbf{x}} = \sum_i \alpha_i \mathbf{v}_i$ and $\tilde{\mathbf{y}} = \sum_i \beta_i \mathbf{v}_i$ be the decomposition of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ along the eigenvectors (note that α_i and β_i might be complex numbers). We have

$$\tilde{\mathbf{x}}^T A \tilde{\mathbf{y}} = \left| \sum_i \alpha_i \lambda_i(A) \beta_i \right| \leq \sqrt{\sum_i |\overline{\alpha_i}|^2 \cdot \sum_i |\lambda_i(A)\beta_i|^2} = \sqrt{\sum_i |\lambda_i(A)|^2 |\beta_i|^2} \leq |\lambda_1(A)| \, , \tag{10}$$

where the first inequality follows by using Cauchy-Schwarz ($\overline{\alpha}$ denotes the complex conjugate of α). We then use the fact that $\sum_i |\alpha_i|^2 = \sum_i |\beta_i|^2 = 1$ which follow from the fact that $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ are unit vectors. Finally, since we have that $|\lambda_1(A)| = o(n)$ and that $\tilde{\mathbf{x}}^T A \tilde{\mathbf{y}} \geq cn$ equation (10) gives a contradiction. So T must satisfy $\mathcal{Q}$. □

We now turn to prove the second direction of Theorem 2.3.

Lemma 2.7. *Let $k \geq 4$ be an even integer. If a tournament satisfies $\mathcal{Q}$ then it satisfies $\mathcal{P}(k)$.*

Proof. Suppose T satisfies $\mathcal{Q}$. Then by the result of [22] mentioned earlier, T must also satisfy $\mathcal{P}(4)$. From Claim 2.5, we have that

$$|\mathrm{tr}(A^4)| = \left| \sum_{i=1}^{n} \lambda_i^4 \right| = o(n^4) \, , \tag{11}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. We will now apply induction to show that $|\mathrm{tr}(A^k)| = o(n^k)$ for all even integers $k \geq 4$. Claim 2.5 would then imply that $\mathcal{P}(k)$ is true for all even integers $k \geq 4$.

Now note the following for an even integer $k > 4$:

$$|\mathrm{tr}(A^k)| = \left| \sum_i \lambda_i^k \right| \leq \sqrt{\sum_i \lambda_i^4 \sum_i \lambda_i^{2k-4}} \leq \sqrt{\sum_i \lambda_i^4} \cdot \left| \sum_i \lambda_i^{k-2} \right| = o(n^k) \, .$$

The first inequality is Cauchy-Schwarz. For the second inequality, recall that by Claim 2.4 we have that λ_i^k are either all non-negative or non-positive. This means that $(\sum_{i=1}^n \lambda_i^{k-2})^2 \geq \sum_{i=1}^n \lambda_i^{2k-4}$ since we lose only non-negative terms. The last equality follows by applying the induction hypothesis and (11). □

2.3 *Discussions*

2.3.1 Spectral Characterization of Quasi-random Tournaments

First of all, the proof of Lemma 2.6 shows that if T satisfies the property $\mathcal{P}(4)$, then $|\lambda_1(A)| = o(n)$ which in turn implies that T satisfies $\mathcal{Q}$. Since we also know that $\mathcal{Q}$ implies $\mathcal{P}(4)$ we conclude the following:

Theorem 2.8 (Spectral Characterization of Quasi-random Tournaments)**.** *A tournament T is quasi-random if and only if the largest eigenvalue of its adjacency matrix satisfies $|\lambda_1(A)| = o(n)$.*

This is in line with other spectral characterizations of quasi-randomness for other combinatorial objects [3, 4, 19, 26, 61].

2.3.2 Connection between $\mathsf{E}_k(T)$ and parity of $k/2$

Let $k \geq 4$ be an even integer. Now we make an observation about $\mathsf{E}_k(T)$ for an arbitrary tournament T (which is not necessarily quasi-random). The total number of distinct k-cycles of T is $\operatorname{tr}(B^k)$, where B is the adjacency matrix of the undirected complete graph on n vertices. Since the spectrum of B is $\{n-1, -1, \ldots, -1\}$ we get $\operatorname{tr}(B^k) = (n-1)^k + (n-1)$. For $k \equiv 0 \pmod 4$, by Claim 2.4, the eigenvalues of A^k are all non-negative and thus we have $\operatorname{tr}(A^k) \geq 0$. By Claim 2.5, we have that $\mathsf{E}_k(T) \geq ((n-1)^k + (n-1))/2$. For $k \equiv 2 \pmod 4$, we can conclude similarly using Claims 2.4 and 2.5 that $\mathsf{E}_k(T) \leq ((n-1)^k + (n-1))/2$.

2.3.3 Quasi-random Directed Graphs

Tournaments are a special case of general directed graphs. So it is natural to ask whether the results proved in this chapter can be generalized to directed graphs. We note that this is indeed the case; we can use the ideas we used here to prove similar results for general directed graphs as defined by Griffiths [48]. The adjacency matrix

A for a directed graph D is defined in the following way. For every $u, v \in V$,

$$A_{u,v} = \begin{cases} 1 & \text{if } (u,v) \in E(T) \\ -1 & \text{if } (v,u) \in E(T) \\ 0 & \text{if } u \text{ and } v \text{ are not connected} \end{cases} .$$

Also, let $\lambda_1(A), \ldots, \lambda_n(A)$ be the eigenvalues of A sorted by their absolute value, so that $\lambda_1(A)$ has the largest absolute value. Griffiths defined quasi-random directed graphs and showed that quasi-randomness is characterized by several equivalent properties. One of these properties is the following:

Definition 2.9 ([48]). *A directed graph D on n vertices is quasi-random if and only if $|\lambda_1(A)| = o(n)$.*

Let us extend the definition of cycles and even cycles for directed graphs as well. Let $\mathsf{C}_k(D)$ denote the total number of k-cycles in D and as before, let $\mathsf{E}_k(D)$ denote the number of even k-cycles in D. We extend Definition 2.2 of $\mathcal{P}(k)$ to directed graphs as below.

Definition 2.10. *A directed graph D on n vertices satisfies property $\mathcal{P}(k)$ if $\mathsf{E}_k(D) = 1/2\mathsf{C}_k(D) + o(n^k)$.*

We prove the following result, analogous to Theorem 2.3:

Theorem 2.11. *The following holds for every fixed even integer $k \geq 4$: A directed graph is quasi-random if and only if it satisfies property $\mathcal{P}(k)$.*

Much of the proof is similar to the proof of Theorem 2.3. We first note that Claim 2.4 is true for directed graphs as well, and hence for all even k, the eigenvalues of A^k are either all non-negative or all non-positive. The claim below is the directed graph analogue of Claim 2.5.

Claim 2.12. *Let A be the adjacency matrix of the directed graph D. Then for an even integer $k \geq 4$, we have*

$$tr(A^k) = 2\mathsf{E}_k(D) - \mathsf{C}_k(D).$$

In particular, D satisfies the property $\mathcal{P}(k)$ if and only if $|tr(A^k)| = o(n^k)$.

Proof. The proof is similar to the proof of Claim 2.5. We first observe that $\mathrm{tr}(A^k)$ is the difference between all labeled even k-cycles and all labeled odd k-cycles. Thus it follows that $\mathrm{tr}(A^k) = 2\mathsf{E}_k(D) - \mathsf{C}_k(D)$.

Now, by Definition 2.10, we can conclude that D satisfies $\mathcal{P}(k)$ if $|\mathrm{tr}(A^k)| = o(n^k)$. □

We now note that the proof of Theorem 2.11 follows from the analogues of Lemmas 2.6 and 2.7. We state the corresponding lemmas below. We remark that the proofs are very similar to the case of tournaments, and so we omit them.

Lemma 2.13. *Let $k \geq 4$ be an even integer. If a directed graph satisfies $\mathcal{P}(k)$ then it is quasi-random.*

Lemma 2.14. *Let $k \geq 4$ be an even integer. If a directed graph is quasi-random then it satisfies $\mathcal{P}(k)$.*

CHAPTER III

A DETERMINISTIC ALGORITHM FOR THE FRIEZE-KANNAN REGULARITY LEMMA

3.1 Introduction

The Regularity Lemma of Szemerédi [93] is one of the most powerful tools in tackling combinatorial problems in various areas like extremal graph theory, additive combinatorics and combinatorial geometry. The regularity lemma guarantees that the vertex set of any (dense) graph $G = (V, E)$ can be partitioned into a *bounded* number of vertex sets $V_1, \ldots, V_k$ such that almost all the bipartite graphs (V_i, V_j) are quasi-random. Hence, one can think of Szemerédi's regularity lemma as saying that any graph can be approximated by a finite structure. This aspect of the regularity lemma has turned out to be extremely useful for designing approximation algorithms, since in some cases one can approximate certain properties of a graph (say, the Max-Cut of the graph) by investigating its regular partition (which is of constant size). In order to apply this algorithmic scheme one should be able to efficiently construct a partition satisfying the condition of the lemma. While Szemerédi's proof of his lemma was only existential, it is known how to efficiently construct a partition satisfying the conditions of the lemma. The first to achieve this goal were Alon et al. [5] who showed that this task can be carried out in time $O(n^\omega)$, where here and throughout this chapter ω is the exponent of fast matrix multiplication. The algorithm of Coppersmith and Winograd [30] gives $\omega < 2.376$. The $O(n^\omega)$ algorithm of Alon et al. [5] was later improved by Kohayakawa, Rödl and Thoma [63] who gave a deterministic $O(n^2)$ algorithm.

We have already seen the main drawback of Szemerédi regularity in Section 1.2.2, the number of parts required for the regularity partition can be huge. Frieze and Kannan devised a weaker notion of regularity (FK-regularity) that would be applicable, but does not involve such huge constants. As in the case of Szemerédi's regularity lemma, in order to algorithmically apply the FK-regularity lemma, one needs to be able to efficiently construct a partition satisfying the conditions of the lemma. Frieze and Kannan also showed that this task can be performed in *randomized* $O(n^2)$ time. Alon and Naor [7] have shown that one can construct such a partition in deterministic polynomial time. The algorithm of Alon and Naor [7] requires solving a semi-definite program (SDP) and hence is not very efficient[1]. The fast boolean matrix multiplication of Bansal and Williams [12] applies the randomized algorithm of [38, 39] for constructing an FK-regular partition. In an attempt to derandomize their matrix multiplication algorithm, Williams [104] asked if one can construct an FK-regular partition in deterministic time $O(n^{3-c})$ for some $c > 0$. Our main result in this chapter answers this question by exhibiting a deterministic $\tilde{O}(n^{\omega})$ time algorithm. Furthermore, as part of the design of this algorithm, we also show that one can find an approximation[2] to the first eigenvalue of a symmetric matrix in *deterministic* time $\tilde{O}(n^{\omega})$.

Besides the above algorithmic motivation for our work, a further combinatorial motivation comes from the study of quasi-random structures. Different notions of quasi-randomness have been extensively studied in the last decade, both in theoretical computer science and in discrete mathematics. A key question that is raised in such cases is: Does there exist a *deterministic* condition that guarantees that a certain structure (say, graph or boolean function) behaves like a typical *random* structure? A well known result of this type is the discrete Cheeger's inequality [3], which relates the expansion of a graph to the spectral gap of its adjacency matrix. Other results

[1] In fact, after solving the SDP, the algorithm of [7] needs time $O(n^3)$ to round the SDP solution.

[2] The necessity of approximation when dealing with eigenvalues is due to the non-existence of algebraic roots of high degree polynomials.

of this type relate the quasi-randomness of functions over various domains to certain norms (the so-called Gowers norms). We refer the reader to the surveys of Gowers [43] and Trevisan [100] for more examples and further discussion on different notions of quasi-randomness. An FK-regular partition is useful since it gives a quasi-random description of a graph. Hence, it is natural to ask if one can characterize this notion of quasi-randomness using a deterministic condition. The work of Alon and Naor [7] gives a condition that can be checked in polynomial time. However, as we mentioned before, verifying this condition requires one to solve a semi-definite program and is thus not efficient. In contrast, our main result in this chapter gives a deterministic condition for FK-regularity that can be stated very simply and checked very efficiently.

3.1.1 The main result

We recall the definitions related to the regularity lemma. For a pair of subsets $A, B \subseteq V(G)$ in a graph $G = (V, E)$, let $e(A, B)$ denote the number of edges between A and B, counting each of the edges contained in $A \cap B$ twice. The density $d(A, B)$ is defined to be $d(A, B) = \frac{e(A,B)}{|A||B|}$. We will frequently deal with a partition of the vertex set $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$. The *order* of such a partition is the number of sets V_i (k in the above partition). A partition is *equitable* if all sets are of size $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$. We will make use of the shorthand notation for density across parts, $d_{ij} = d(V_i, V_j)$ whenever $i \neq j$. Also, we set $d_{ii} = 0$ for all i.

The key notion in Szemerédi's regularity lemma [93] is the following: Let A, B be disjoint sets of vertices. We say that (A, B) is ε-regular if $|d(A, B) - d(A', B')| \leq \varepsilon$ for all $A' \subseteq A$ and $B' \subseteq B$ satisfying $|A'| \geq \varepsilon|A|$ and $|B'| \geq \varepsilon|B|$. It is not hard to show (see [64]) that ε-regular bipartite graphs behave like random graphs in many ways. Szemerédi's Regularity Lemma [93] states that given $\varepsilon > 0$ there is a constant $T(\varepsilon)$, such that the vertex set of any graph $G = (V, E)$ can be partitioned into k equitable sets $V_1, \ldots, V_k$, where $k \leq T(\varepsilon)$ and all but εk^2 of the pairs (i, j) are such that (V_i, V_j)

is ε-regular.

One of the useful aspects of an ε-regular partition of a graph is that it allows one to estimate the number of edges in certain partitions of G. For example, given an ε-regular partition, one can estimate the value of the Max-Cut in G within an error of εn^2, in time that depends only on the order of the partition (and independent of the order of G!). Hence, one would like the order of the partition to be relatively small. However, as we have mentioned above, Gowers [42] has shown that there are graphs whose ε-regular partitions have size at least Tower$(1/\varepsilon^{1/16})$, namely a tower of exponents of height $1/\varepsilon^{1/16}$.

To remedy this, Frieze and Kannan [38, 39] introduced the following relaxed notion of regularity, which we will call ε-FK-regularity.

Definition 3.1 (ε-FK-regular)**.** *Let $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ be a partition of $V(G)$. For subsets $S, T \subseteq V$ and $1 \le i \le k$, let $S_i = S \cap V_i$ and $T_i = T \cap V_i$. Define $\Delta(S,T)$ for subsets $S, T \subseteq V$ as follows:*

$$\Delta(S,T) = e(S,T) - \sum_{i,j} d_{ij}|S_i||T_j|. \tag{12}$$

The partition $\mathcal{P}$ is said to be ε-FK-regular if it is equitable and

$$\text{for all subsets } S, T \subseteq V, \quad |\Delta(S,T)| \le \varepsilon n^2. \tag{13}$$

If $|\Delta(S,T)| > \varepsilon n^2$ then S, T are said to be witnesses to the fact that $\mathcal{P}$ is not ε-FK-regular.

As we have mentioned before, Frieze and Kannan [38, 39] proved that one can construct an ε-FK regular partition of a graph in *randomized* time $O(n^2)$. Our main result in this chapter is the following deterministic algorithmic version of the FK-regularity lemma that answers a question of Williams [104].

Theorem 3.2 (Main Result)**.** *Given $\varepsilon > 0$ and an n vertex graph $G = (V, E)$, one can find in deterministic time $O\left(\frac{1}{\varepsilon^6} n^\omega \log\log n\right)$ an ε-FK-regular partition of G of order at most $2^{10^8/\varepsilon^7}$.*

3.1.2 Chapter overview

The rest of the chapter is organized as follows. As we have mentioned earlier, the relation between quasi-random properties and spectral properties of graphs goes back to the Cheeger's Inequality [3]. Furthermore, it was shown in [40] that one can characterize the notion of Szemerédi's regularity using a spectral condition. In Section 3.2 we introduce a spectral condition for ε-FK-regularity and show that it characterizes this property. In order to be able to check this spectral condition efficiently, one has to be able to approximately compute the first eigenvalue of a matrix. Hence, in Section 3.3 we show that this task can be carried out in deterministic time $\tilde{O}(n^{\omega})$. We use a deterministic variant of the randomized power iteration method. Since we could not find a reference for this, we include the proof for completeness. As in other algorithmic versions of regularity lemmas, the non-trivial task is that of checking whether a partition is regular, and if it is not, then finding sets S, T that violate this property (recall Definition 3.1). This key result is stated in Corollary 3.9. We explain the (somewhat routine) process of deducing Theorem 3.2 from Corollary 3.9 in Section 3.4. Finally, Section 3.5 contains some concluding remarks and open problems.

3.2 A Spectral Condition for FK-Regularity

In this section we introduce a spectral condition that "characterizes" partitions that are ε-FK regular. Actually, the condition will allow us to quickly distinguish between partitions that are ε-FK regular from partitions that are not $\varepsilon^3/1000$-FK regular. As we will show later on, this is all one needs in order to efficiently construct an ε-FK regular partition. Our spectral condition relies on the following characterization of eigenvalues of a matrix. We omit the proof of this standard fact.

Lemma 3.3 (First eigenvalue)**.** *For a diagonalizable matrix M, the absolute value of* *the first eigenvalue $\lambda_1(M)$ is given by the following:*

$$|\lambda_1(M)| = \max_{\|\mathbf{x}\|=\|\mathbf{y}\|=1} \mathbf{x}^T M \mathbf{y}.$$

We say that an algorithm computes a δ-approximation to the first eigenvalue of a matrix M if it finds two unit vectors $\mathbf{x}, \mathbf{y}$ achieving $\mathbf{x}^T M \mathbf{y} \geq (1-\delta)|\lambda_1(M)|$. Our goal in this section is to prove the following theorem:

Theorem 3.4. *Suppose there is an $S(n)$ time algorithm for computing a $1/2$-approximation of the first eigenvalue of a symmetric $n \times n$ matrix. Then there is an $O(n^2 + S(n))$ time algorithm that given $\varepsilon > 0$, and a partition $\mathcal{P}$ of the vertices of an n-vertex graph $G = (V, E)$, does one of the following:*

1. *Correctly states that $\mathcal{P}$ is ε-FK-regular.*

2. *Produces sets S, T that witness the fact that $\mathcal{P}$ is not $\varepsilon^3/1000$-FK-regular.*

Let A be the adjacency matrix of the graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\} = [n]$. Let $S, T \subseteq V$ be subsets of the vertices and $\mathbf{x}_S, \mathbf{x}_T$ denote the corresponding indicator vectors. We would like to test if a partition $\mathcal{P} = V_1, \ldots, V_k$ of V is a ε-FK-regular partition. We define a matrix $D = D(\mathcal{P})$ in the following way. Let $1 \leq i, j \leq n$ and suppose vertex i belongs to V_{l_i} in $\mathcal{P}$ and vertex j belongs to V_{l_j}, for some $1 \leq l_i, l_j \leq k$. Then the (i,j)th entry of D is given by $D_{ij} = d_{l_i l_j}$. Thus the matrix D is a block matrix (each block corresponding to the parts in the partition), where each block contains the same value at all positions, the value being the density of edges corresponding to the two parts. Now define $\Delta = A - D$. For $S, T \subseteq V$ and an $n \times n$ matrix M, define

$$M(S,T) = \sum_{i \in S, j \in T} M(i,j) = \mathbf{x}_S^T M \mathbf{x}_T.$$

Notice that for the matrix Δ, the above definition coincides with (12):

$$\begin{aligned}\Delta(S,T) &= A(S,T) - D(S,T)\\ &= e(S,T) - \sum_{i,j} d_{ij}|S_i||T_j|,\end{aligned}$$

where $S_i = S \cap V_i$ and $T_j = T \cap V_j$.

Summarizing, $\mathcal{P}$ is an ε-FK-regular partition of V if and only if for all $S, T \subseteq V$, we have $|\Delta(S,T)| \leq \varepsilon n^2$.

Let $G = (V,E)$ be an n-vertex graph, let $\mathcal{P}$ be a partition of $V(G)$ and let Δ be the matrix defined above. Notice that by construction, Δ is a symmetric matrix and so it can be diagonalized with real eigenvalues. Lemmas 3.5 and 3.7 below will establish a relation between the first eigenvalue of Δ and the FK-regularity properties of $\mathcal{P}$.

Lemma 3.5. *If $|\lambda_1(\Delta)| \leq \gamma n$ then $\mathcal{P}$ is γ-FK-regular.*

Proof. Suppose $\mathcal{P}$ is not γ-FK-regular and let S, T be two sets witnessing this fact, that is, satisfying $|\Delta(S,T)| = |\mathbf{x}_S^T \Delta \mathbf{x}_T| > \gamma n^2$. Normalizing the vectors $\mathbf{x}_S$ and $\mathbf{x}_T$, we have $\tilde{\mathbf{x}}_S = \mathbf{x}_S/\|\mathbf{x}_S\| = \mathbf{x}_S/\sqrt{|S|}$ and $\tilde{\mathbf{x}}_T = \mathbf{x}_T/\|\mathbf{x}_T\| = \mathbf{x}_T/\sqrt{|T|}$. We get

$$|\tilde{\mathbf{x}}_S^T \Delta \tilde{\mathbf{x}}_T| > \gamma n^2/(\sqrt{|S|\,|T|}) \geq \gamma n\ ,$$

where the last inequality follows since $|S|, |T| \leq n$. By the characterization of the first eigenvalue, we have that $|\lambda_1(\Delta)| > \gamma n$. □

Claim 3.6. *Suppose two vectors $\mathbf{p}, \mathbf{q} \in [-1,1]^n$ satisfying $\mathbf{p}^T\Delta\mathbf{q} > 0$ are given. Then, in deterministic time $O(n^2)$, we can find sets $S, T \subseteq [n]$ satisfying $|\Delta(S,T)| \geq \frac{1}{4}\mathbf{p}^T\Delta\mathbf{q}$.*

Proof. Let us consider the positive and negative parts of the vectors $\mathbf{p}$ and $\mathbf{q}$. Of the four combinations, $(\mathbf{p}^+, \mathbf{q}^+)$, $(\mathbf{p}^+, \mathbf{q}^-)$, $(\mathbf{p}^-, \mathbf{q}^+)$ and $(\mathbf{p}^-, \mathbf{q}^-)$, at least one pair should give rise to a product at least $\mathbf{p}^T\Delta\mathbf{q}/4$. Let us call this pair the good pair.

Suppose the good pair is $\mathbf{p}^+, \mathbf{q}^+$. Let Δ_i, Δ^j denote respectively the ith row and jth column of Δ. We can write $(\mathbf{p}^+)^T \Delta \mathbf{q}^+ = \sum_i p_i^+ \langle \Delta_i, \mathbf{q}^+ \rangle$. Compute the n products, $\langle \Delta_i, \mathbf{q}^+ \rangle$. We put vertex i in S if and only if $\langle \Delta_i, \mathbf{q}^+ \rangle \geq 0$. For this choice of S, we have $\mathbf{x}_S^T \Delta \mathbf{q}^+ \geq (\mathbf{p}^+)^T \Delta \mathbf{q}^+$. Similarly as before, we have $\mathbf{x}_S^T \Delta \mathbf{q}^+ = \sum_j q_j^+ \langle \mathbf{x}_S, \Delta^j \rangle$, therefore depending on the signs of $\langle \mathbf{x}_S, \Delta^j \rangle$, we define whether j belongs to T. Thus we get sets S, T such that $\Delta(S,T) = \mathbf{x}_S^T \Delta \mathbf{x}_T \geq (\mathbf{p}^+)^T \Delta \mathbf{q}^+ \geq \mathbf{p}^T \Delta \mathbf{q}/4$. Notice that this rounding takes $O(n^2)$ time, since we need to perform $2n$ vector products, each of which takes $O(n)$ time.

If exactly one of $\mathbf{p}^-$ or $\mathbf{q}^-$ is part of the good pair, then we could replicate the above argument in a similar manner. Thus we would get $\Delta(S,T) \leq -\mathbf{p}^T \Delta \mathbf{q}/4$. If the good pair is $(\mathbf{p}^-, \mathbf{q}^-)$, we would again get $\Delta(S,T) \geq \mathbf{p}^T \Delta \mathbf{q}/4$. □

Lemma 3.7. *If $|\lambda_1(\Delta)| > \gamma n$, then $\mathcal{P}$ is not $\gamma^3/108$-FK-regular. Furthermore, given unit vectors $\mathbf{x}, \mathbf{y}$ satisfying $\mathbf{x}^T \Delta \mathbf{y} > \gamma n$, one can find sets S, T witnessing this fact in deterministic time $O(n^2)$.*

Proof. As per the previous observation, it is enough to find sets S, T such that $|\Delta(S,T)| > \gamma^3 n^2/108$. By Claim 3.6, it is enough to find vectors $\mathbf{p}$ and $\mathbf{q}$ in $[-1,1]^n$ satisfying $\mathbf{p}^T \Delta \mathbf{q} > \gamma^3 n^2/27$.

Suppose that $|\lambda_1(\Delta)| > \gamma n$ and let $\mathbf{x}, \mathbf{y}$ satisfy $\|x\| = \|\mathbf{y}\| = 1$ and $\mathbf{x}^T \Delta \mathbf{y} > \gamma n$. Let $\beta > 1$ (β will be chosen to be $3/\gamma$ later on) and define $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ in the following manner:

$$\hat{x}_i = \begin{cases} x_i : \text{ if } |x_i| \leq \frac{\beta}{\sqrt{n}} \\ 0 : \text{ otherwise} \end{cases}, \qquad \hat{y}_i = \begin{cases} y_i : \text{ if } |y_i| \leq \frac{\beta}{\sqrt{n}} \\ 0 : \text{ otherwise} \end{cases}.$$

We claim that

$$\hat{\mathbf{x}}^T \Delta \hat{\mathbf{y}} > (\gamma - 2/\beta) n . \tag{14}$$

To prove this, note that

$$\begin{aligned}
\hat{\mathbf{x}}^T \Delta \hat{\mathbf{y}} &= \mathbf{x}^T \Delta \mathbf{y} - (\mathbf{x} - \hat{\mathbf{x}})^T \Delta \mathbf{y} - \hat{\mathbf{x}}^T \Delta (\mathbf{y} - \hat{\mathbf{y}}) \\
&> \gamma n - (\mathbf{x} - \hat{\mathbf{x}})^T \Delta \mathbf{y} - \hat{\mathbf{x}}^T \Delta (\mathbf{y} - \hat{\mathbf{y}}) \\
&\geq \gamma n - |(\mathbf{x} - \hat{\mathbf{x}})^T \Delta \mathbf{y}| - |\hat{\mathbf{x}}^T \Delta (\mathbf{y} - \hat{\mathbf{y}})| \ .
\end{aligned}$$

Hence, to establish (14) it would suffice to bound $|(\mathbf{x} - \hat{\mathbf{x}})^T \Delta \mathbf{y}|$ and $|\hat{\mathbf{x}}^T \Delta (\mathbf{y} - \hat{\mathbf{y}})|$ from above by n/β. To this end, let $C(\mathbf{x}) = \{i : \ |x_i| \geq \beta/\sqrt{n}\}$, and note that since $\|\mathbf{x}\| = 1$ we have $|C(\mathbf{x})| \leq n/\beta^2$. Now define Δ' as

$$\Delta'_{ij} = \begin{cases} \Delta_{ij} & \text{if } i \in C(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} .$$

We now claim that the following holds:

$$\begin{aligned}
|(\mathbf{x} - \hat{\mathbf{x}})^T \Delta \mathbf{y}| = |(\mathbf{x} - \hat{\mathbf{x}})^T \Delta' \mathbf{y}| &\leq \|(\mathbf{x} - \hat{\mathbf{x}})^T\| \|\Delta' \mathbf{y}\| \\
&\leq \|\Delta' \mathbf{y}\| \\
&\leq \|\Delta'\|_F \|\mathbf{y}\| \\
&= \|\Delta'\|_F \\
&\leq n/\beta \ .
\end{aligned}$$

Indeed, the first inequality is Cauchy-Schwarz and in the second inequality we use the fact that $\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \|\mathbf{x}\| = 1$. In the third inequality $\|\Delta'\|_F$ denotes $\sqrt{\sum_{i,j} (\Delta'_{ij})^2}$ and the inequality follows from Cauchy-Schwarz. The fourth line is an equality that follows from $\|\mathbf{y}\| = 1$. The last inequality follows from observing that since $|C(\mathbf{x})| \leq n/\beta^2$ the matrix Δ' has only n^2/β^2 non-zero entries, and each of these entries is of absolute value at most 1. It follows from an identical argument that $|\hat{\mathbf{x}}^T \Delta (\mathbf{y} - \hat{\mathbf{y}})| \leq n/\beta$, thus proving (14). After rescaling $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, we get

$$((\sqrt{n}/\beta)\hat{\mathbf{x}})^T \Delta ((\sqrt{n}/\beta)\hat{\mathbf{y}}) > (\gamma - 2/\beta) n^2/\beta^2 \ .$$

Setting $\beta = 3/\gamma$ so that $(\gamma - 2/\beta)/\beta^2$ is maximized, the right hand side of the inequality is $\gamma^3 n^2/27$. Now that we have the necessary vectors $\mathbf{p} = (\sqrt{n}/\beta)\hat{\mathbf{x}}$ and $\mathbf{q} = (\sqrt{n}/\beta)\hat{\mathbf{x}}$, an application of Claim 3.6 completes the proof. □

The proof of Theorem 3.4 now follows easily from Lemmas 3.5 and 3.7.

Proof of Theorem 3.4. We start with describing the algorithm. Given $G = (V, E)$, $\varepsilon > 0$ and a partition $\mathcal{P}$ of $V(G)$, the algorithm first computes the matrix $\Delta = A - D$ (in time $O(n^2)$) and then computes unit vectors $\mathbf{x}, \mathbf{y}$ satisfying $\mathbf{x}^T \Delta \mathbf{y} \geq \frac{1}{2}|\lambda_1(\Delta)|$ (in time $S(n)$). If $\mathbf{x}^T \Delta \mathbf{y} \leq \varepsilon n/2$ the algorithm declares that $\mathcal{P}$ is ε-FK-regular, and if $\mathbf{x}^T \Delta \mathbf{y} > \varepsilon n/2$ it declares that $\mathcal{P}$ is not $\varepsilon^3/1000$-FK-regular and then uses the $O(n^2)$ time algorithm of Lemma 3.7 in order to produce sets S, T that witness this fact. The running time of the algorithm is clearly $O(n^2 + S(n))$.

Now let us discuss the correctness of the algorithm. If $\mathbf{x}^T \Delta \mathbf{y} \leq \varepsilon n/2$ then since $\mathbf{x}^T \Delta \mathbf{y}$ is a 1/2-approximation for $|\lambda_1(\Delta)|$, we can conclude that $|\lambda_1(\Delta)| \leq \varepsilon n$. Hence, by Lemma 3.5 we have that $\mathcal{P}$ is indeed ε-FK-regular. If $\mathbf{x}^T \Delta \mathbf{y} > \varepsilon n/2$ then by Lemma 3.7 we are guaranteed to obtain sets S, T that witness the fact that $\mathcal{P}$ is not $\varepsilon^3/(108 \cdot 8) \geq \varepsilon^3/1000$-FK-regular. □

3.3 Finding the First Eigenvalue Deterministically

In order to efficiently apply Theorem 3.4 from the previous section, we will need an efficient algorithm for approximating the first eigenvalue of a symmetric matrix. Such an algorithm is guaranteed by the following theorem that we prove in this section:

Theorem 3.8. *Given an $n \times n$ symmetric matrix H, and a parameter $0 < \delta < 1$, one can find in deterministic time $O\left(n^\omega \log\left(\frac{1}{\delta}\log\left(\frac{n}{\delta}\right)\right)\right)$ unit vectors $\mathbf{x}, \mathbf{y}$ satisfying*

$$\mathbf{x}^T H \mathbf{y} \geq (1 - \delta)|\lambda_1(H)|.$$

Setting $H = \Delta$ and $\delta = 1/2$ in Theorem 3.8, and using Theorem 3.4 we infer the following corollary.

Corollary 3.9. *There is an $O(n^{\omega} \log\log n)$ time algorithm, that given $\varepsilon > 0$, an n-vertex graph $G = (V, E)$ and a partition $\mathcal{P}$ of $V(G)$, does one of the following:*

1. *Correctly states that $\mathcal{P}$ is ε-FK-regular.*

2. *Finds sets S, T that witness the fact that $\mathcal{P}$ is not $\varepsilon^3/1000$-FK-regular.*

As we have mentioned in Section 3.1, one can derive our main result stated in Theorem 3.2 from Corollary 3.9 using the proof technique of Szemerédi [93]. This is discussed in Section 3.4.

We also note that the proof of Theorem 3.8 can be modified to approximate the quantity $\max_{\|\mathbf{x}\|=\|\mathbf{y}\|=1} \mathbf{x}^T H \mathbf{y}$ for any matrix H. This quantity is the so-called first singular value of H. But since we do not need this for our specific application to FK-regularity, we state the theorem "only" for symmetric matrices H.

Getting back to the proof of Theorem 3.8 we first recall that for any matrix H we have $|\lambda_1(H)| = \sqrt{\lambda_1(H^2)}$ (notice that H^2 is positive semi-definite, so all its eigenvalues are non-negative). Hence, in order to compute an approximation to $|\lambda_1(H)|$, we shall compute an approximation to $\lambda_1(H^2)$. Theorem 3.8 will follow easily once we prove the following:

Theorem 3.10. *Given an $n \times n$ positive semi-definite matrix M, and a parameter $0 < \delta < 1$, there exists an algorithm that runs in $O\left(n^{\omega} \log\left(\frac{1}{\delta}\log\left(\frac{n}{\delta}\right)\right)\right)$ time and outputs a vector $\mathbf{b}$ such that*

$$\frac{\mathbf{b}^T M \mathbf{b}}{\mathbf{b}^T \mathbf{b}} \geq (1 - \delta)\lambda_1(M).$$

We shall first derive Theorem 3.8 from Theorem 3.10.

Proof of Theorem 3.8. As mentioned above, $|\lambda_1(H)| = \sqrt{\lambda_1(H^2)}$. Since H^2 is positive semi-definite we can use Theorem 3.10 to compute a vector $\mathbf{b}$ satisfying

$$\frac{\mathbf{b}^T H^2 \mathbf{b}}{\mathbf{b}^T \mathbf{b}} = \hat{\lambda}_1 \geq (1 - \delta)\lambda_1(H^2).$$

We shall see that $\sqrt{\hat{\lambda}_1}$ is a $(1-\delta)$ approximation to the first eigenvalue of H. To recover the corresponding vectors as in Lemma 3.3, notice that

$$\mathbf{b}^T H^2 \mathbf{b} = \|H\mathbf{b}\|^2 = \hat{\lambda}_1 \|\mathbf{b}\|^2 \quad \Longrightarrow \quad \|H\mathbf{b}\| = \sqrt{\hat{\lambda}_1}\|\mathbf{b}\|.$$

Setting $\mathbf{x} = \frac{H\mathbf{b}}{\sqrt{\hat{\lambda}_1}\|\mathbf{b}\|}$ and $\mathbf{y} = \frac{\mathbf{b}}{\|\mathbf{b}\|}$, we obtain unit vectors $\mathbf{x}$ and $\mathbf{y}$ satisfying

$$\mathbf{x}^T H \mathbf{y} = \sqrt{\hat{\lambda}_1} \geq \sqrt{(1-\delta)\lambda_1(H^2)} \geq (1-\delta)|\lambda_1(H)| \, .$$

The main step that contributes to the running time is the computation of $\mathbf{b}$ using Theorem 3.10 and hence the running time is $O\left(n^\omega \log\left(\frac{1}{\delta}\log\left(\frac{n}{\delta}\right)\right)\right)$, as needed. □

We turn to prove Theorem 3.10. We shall apply the *power iteration method* to compute an approximation of the first eigenvalue of a positive semi-definite (PSD) matrix. Power iteration is a technique that can be used to compute the largest eigenvalues and is a very widely studied method. For instance, the paper [66] by Kuczyński and Woźniakowski has a very thorough analysis of the method. The earlier work of [76] shows that power iteration is much more effective with PSD matrices. A much simpler (albeit slightly weaker) analysis was given in [101].

A PSD matrix M has all nonnegative eigenvalues. The goal of power iteration is to find the first eigenvalue and the corresponding eigenvector of M. The basic idea is that an arbitrary vector $\mathbf{r}$ is taken, and is repeatedly multiplied with the matrix M. The eigenvectors of M provide an orthonormal basis for $\mathbb{R}^n$. The vector $\mathbf{r}$ can be seen as a decomposition into components along the direction of each of the eigenvectors of the matrix. With each iteration of multiplication by M, the component of $\mathbf{r}$ along the direction of the first eigenvector gets magnified more than the component of $\mathbf{r}$ along the direction of the other eigenvectors. This is because the first eigenvalue is larger than the other eigenvalues. One of the key properties that is required of $\mathbf{r}$ is that it has a nonzero component along the first eigenvector. This is typically ensured by setting $\mathbf{r}$ to be a random unit vector. However, since we are looking for a deterministic algorithm, we ensure that by using n different orthogonal basis vectors.

We first need the following key lemma.

Lemma 3.11. *Let M be a positive semi-definite matrix. Let $\mathbf{a} \in \mathbb{R}^n$ be a unit vector such that $|\langle \mathbf{v}_1, \mathbf{a} \rangle| \geq 1/\sqrt{n}$. Then, for every positive integer s and $0 < \delta < 1$, for $\mathbf{b} = M^s \mathbf{a}$, we have*

$$\frac{\mathbf{b}^T M \mathbf{b}}{\mathbf{b}^T \mathbf{b}} \geq \lambda_1 \cdot \left(1 - \frac{\delta}{2}\right) \cdot \frac{1}{1 + n\left(1 - \frac{\delta}{2}\right)^{2s}} \ ,$$

where λ_1 denotes the first eigenvalue of M.

Proof. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ be the n eigenvalues of M (with multiplicities), and let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the corresponding orthonormal eigenvectors. We can write $\mathbf{a}$ as a linear combination of the eigenvectors of M.

$$\mathbf{a} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \ldots + \alpha_n \mathbf{v}_n,$$

where the coefficients are $\alpha_i = \langle \mathbf{a}, \mathbf{v}_i \rangle$. By assumption, we have $|\alpha_1| \geq 1/\sqrt{n}$ and since $\mathbf{a}$ is a unit vector, $\sum_i \alpha_i^2 = 1$. Now, we can write $\mathbf{b}$ as follows:

$$\mathbf{b} = \alpha_1 \lambda_1^s \mathbf{v}_1 + \alpha_2 \lambda_2^s \mathbf{v}_2 + \ldots + \alpha_n \lambda_n^s \mathbf{v}_n \ .$$

So we have

$$\mathbf{b}^T M \mathbf{b} = \sum_i \alpha_i^2 \lambda_i^{2s+1} \text{ , and}$$
$$\mathbf{b}^T \mathbf{b} = \sum_i \alpha_i^2 \lambda_i^{2s} \ .$$

We will compute a lower bound to the numerator and upper bound to the denominator, resulting in a lower bound for the fraction.

Let ℓ be the number of eigenvalues larger than $\lambda_1 \cdot (1 - \frac{\delta}{2})$. Since the eigenvalues are numbered in non-increasing order and since M is positive semi-definite [3], we have

$$\mathbf{b}^T M \mathbf{b} \geq \sum_{i=1}^{\ell} \alpha_i^2 \lambda_i^{2s+1} \geq \lambda_1 \left(1 - \frac{\delta}{2}\right) \sum_{i=1}^{\ell} \alpha_i^2 \lambda_i^{2s}. \tag{15}$$

[3] We are dropping terms to get an inequality, implicitly assuming that the dropped terms are nonnegative. If the eigenvalues are negative, this need not hold.

We also have

$$\sum_{i=\ell+1}^{n} \alpha_i^2 \lambda_i^{2s} \leq \lambda_1^{2s} \cdot \left(1 - \frac{\delta}{2}\right)^{2s} \sum_{i=\ell+1}^{n} \alpha_i^2 \leq \lambda_1^{2s} \cdot \left(1 - \frac{\delta}{2}\right)^{2s},$$

where the last inequality follows since $\sum_{i=\ell+1}^{n} \alpha_i^2 \leq \sum_{i=1}^{n} \alpha_i^2 = 1$. Continuing using the fact that $1 \leq n\alpha_1^2$, we have,

$$\lambda_1^{2s} \cdot \left(1 - \frac{\delta}{2}\right)^{2s} \leq n\alpha_1^2 \lambda_1^{2s} \cdot \left(1 - \frac{\delta}{2}\right)^{2s} \leq n \left(1 - \frac{\delta}{2}\right)^{2s} \sum_{i=1}^{\ell} \alpha_i^2 \lambda_i^{2s}.$$

Thus we get,

$$\mathbf{b}^T \mathbf{b} \leq \left(1 + n\left(1 - \frac{\delta}{2}\right)^{2s}\right) \cdot \sum_{i=1}^{\ell} \alpha_i^2 \lambda_i^{2s}\ . \tag{16}$$

From (15) and (16) we deduce that

$$\frac{\mathbf{b}^T M \mathbf{b}}{\mathbf{b}^T \mathbf{b}} \geq \lambda_1 \cdot \left(1 - \frac{\delta}{2}\right) \cdot \frac{1}{1 + n\left(1 - \frac{\delta}{2}\right)^{2s}}\ ,$$

thus completing the proof. □

Now we are ready to analyze the power iteration algorithm and to prove Theorem 3.10.

Proof of Theorem 3.10. Consider the n canonical basis vectors, denoted by $\mathbf{e}_i$, for $i = 1, \dots, n$. We can decompose the first eigenvector $\mathbf{v}_1$ of M along these n basis vectors. Since $\mathbf{v}_1$ has norm 1, there must exist an i such that $|\langle \mathbf{v}_1, \mathbf{e}_i \rangle| \geq 1/\sqrt{n}$, by pigeonhole principle. We can perform power iteration of M, starting at these n basis vectors. We would get n output vectors, and for each output vector $\mathbf{x}$, we compute $\mathbf{x}^T M \mathbf{x}/(\mathbf{x}^T \mathbf{x})$, and choose the one that gives us the maximum. By Lemma 3.11, one of these output vectors $\mathbf{x}$ is such that

$$\frac{\mathbf{x}^T M \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \geq \lambda_1(M) \cdot \left(1 - \frac{\delta}{2}\right) \cdot \frac{1}{1 + n\left(1 - \frac{\delta}{2}\right)^{2s}}.$$

If we use $s = O\left(\frac{1}{\delta} \log\left(\frac{n}{\delta}\right)\right)$, we can eliminate the factor n in the denominator, and the denominator would become $(1 + \frac{\delta}{2})$, giving us an estimate of at least $\lambda_1 \cdot (1 - \delta)$, which is what we require.

To perform the n power iterations efficiently, consider taking the sth power of M. Let $N = M^s = M^s \cdot I$. We can think of this as performing n power iteration algorithms in parallel, each one starting with a different canonical basis vector. For each vector $\mathbf{x} = M^s \mathbf{e}_i$, we need to compute $(\mathbf{x}^T M \mathbf{x})/(\mathbf{x}^T \mathbf{x})$. For that we compute the products $P = N^T M N$ and $Q = N^T N$. To get the $\mathbf{x}$ that maximizes the answer, we choose $\max\{P_{ii}/Q_{ii} : 1 \leq i \leq n\}$. The maximized ratio is the approximation to the first eigenvalue, and the corresponding ith column of N is the estimation of the maximizing eigenvector.

For the running time analysis, the most time consuming step is taking the sth power of the matrix M. Using repeated squaring, this can be done in $2 \log s$ matrix multiplications, each of which takes time $O(n^\omega)$. Since we need $s = O\left(\frac{1}{\delta} \log\left(\frac{n}{\delta}\right)\right)$, the running time required by the entire algorithm is bounded by $O\left(n^\omega \log\left(\frac{1}{\delta} \log\left(\frac{n}{\delta}\right)\right)\right)$.

$\square$

3.4 Constructing an FK-Regular Partition

In this section we show how to derive Theorem 3.2 from Corollary 3.9. We start with defining the *index* of a partition, which will be helpful in showing that the algorithm terminates within a bounded number of iterations.

Definition 3.12. *For a partition $\mathcal{P} = (V_1, V_2, \ldots, V_k)$ of the vertex sets of a graph $G = (V, E)$, the index of $\mathcal{P}$ is defined by*

$$ind(\mathcal{P}) = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}^2 |V_i| \, |V_j| \; .$$

Notice that $0 \leq \text{ind}(\mathcal{P}) \leq 1$ for any partition $\mathcal{P}$. We make use of the following theorem (using ideas from the original Szemerédi paper [93]) to refine the partition, whenever the original partition is not ε-FK-regular and improve the index. Since the index is upper bounded by 1, we should not be able to use the following theorem too many times. This implies that refining a finite number of times would result in an ε-FK-regular partition.

Theorem 3.13. *Let $\varepsilon' > 0$. Given a graph $G = (V, E)$ and a partition $\mathcal{P}$ that is not ε'-FK-regular, and sets $S, T \subseteq V$ that violate the condition, the partition can be refined in $O(n)$ time to get a new equitable partition $\mathcal{Q}$, such that ind$(\mathcal{Q}) \geq$ ind$(\mathcal{P}) + \varepsilon'^2/2$. Moreover the new partition $\mathcal{Q}$ has size at most $8/\varepsilon'^2$ times the size of the original partition $\mathcal{P}$.*

Before proving the above theorem, we would need the following form of Cauchy-Schwarz inequality, which we quote from [83] without proof.

Lemma 3.14. *Let $1 \leq M \leq N$, let $\zeta_1, \ldots, \zeta_N$ be positive and $d_1, \ldots, d_N$ and d be reals. If $\sum_{i=1}^N \zeta_i = 1$ and $d = \sum_{i=1}^N d_i\zeta_i$ then*

$$\sum_{i=1}^{N} d_i^2 \zeta_i \geq d^2 + \left(d - \frac{\sum_{i=1}^M d_i\zeta_i}{\sum_{i=1}^M \zeta_i}\right)^2 \frac{\sum_{i=1}^M \zeta_i}{1 - \sum_{i=1}^M \zeta_i} .$$

Proof of Theorem 3.13. Let $\mathcal{P}$ be the partition $\mathcal{P} = (V_1, V_2, \ldots, V_k)$. By the hypothesis that $\mathcal{P}$ is not ε'-FK-regular, we have sets S, T such that

$$\left| e(S,T) - \sum_{i \neq j} d_{ij} |S_i||T_j| \right| > \varepsilon' n^2 .$$

Let us define the following for $i = 1, 2, \ldots, k$:

$$S_i = V_i \cap S, \;\; \bar{S}_i = V_i \backslash S, \;\; T_i = V_i \cap T, \;\; \bar{T}_i = V_i \backslash T .$$

For each $i = 1, 2, \ldots, k$, let us define the following sets as well:

$$V_i^{(1)} = V_i \cap (S \backslash T), \;\; V_i^{(2)} = V_i \cap (T \backslash S), \;\; V_i^{(3)} = V_i \cap (S \cap T), \;\; V_i^{(4)} = V_i \backslash (S \cup T) .$$

Let $\mathcal{R}$ be the partition consisting of all the sets $V_i^{(1)}, V_i^{(2)}, V_i^{(3)}, V_i^{(4)}$ for $i = 1, \ldots, k$. We shall show that ind$(\mathcal{R}) \geq$ ind$(\mathcal{P}) + \varepsilon'^2$.

Let us define $\eta_{i,j} = d(S_i, T_j) - d_{ij}$ for all i, j. We have

$$e(V_i, V_j) = e(S_i, T_j) + e(\bar{S}_i, T_j) + e(S_i, \bar{T}_j) + e(\bar{S}_i, \bar{T}_j) .$$

We can rewrite this as

$$\begin{aligned} d_{ij}|V_i|\,|V_j| = d(S_i,T_j)|S_i|\,|T_j| &+ d(\bar{S}_i,T_j)|\bar{S}_i|\,|T_j| \\ &+ d(S_i,\bar{T}_j)|S_i|\,|\bar{T}_j| + d(\bar{S}_i,\bar{T}_j)|\bar{S}_i|\,|\bar{T}_j|\ . \end{aligned}$$

We also have

$$|V_i|\,|V_j| = |S_i|\,|T_j| + |\bar{S}_i|\,|T_j| + |S_i|\,|\bar{T}_j| + |\bar{S}_i|\,|\bar{T}_j|\ .$$

Using Lemma 3.14 with the above two identities, (setting $N=4$, $M=1$, $\zeta_1 = \frac{|S_i|\,|T_j|}{|V_i|\,|V_j|}$, $\zeta_2 = \frac{|\bar{S}_i|\,|T_j|}{|V_i|\,|V_j|}$, $\zeta_3 = \frac{|S_i|\,|\bar{T}_j|}{|V_i|\,|V_j|}$ and $\zeta_4 = \frac{|\bar{S}_i|\,|\bar{T}_j|}{|V_i|\,|V_j|}$) we get

$$\begin{aligned} \frac{1}{|V_i|\,|V_j|}\left[d^2(S_i,T_j)|S_i|\,|T_j| + d^2(\bar{S}_i,T_j)|\bar{S}_i|\,|T_j| + d^2(S_i,\bar{T}_j)|S_i|\,|\bar{T}_j| + d^2(\bar{S}_i,\bar{T}_j)|\bar{S}_i|\,|\bar{T}_j|\right] \geq \\ d_{ij}^2 + [d_{ij} - d(S_i,T_j)]^2\left[\frac{\frac{|S_i|\,|T_j|}{|V_i|\,|V_j|}}{1-\frac{|S_i|\,|T_j|}{|V_i|\,|V_j|}}\right]\ . \end{aligned}$$

That is,

$$\begin{aligned} d^2(S_i,T_j)|S_i|\,|T_j| + d^2(\bar{S}_i,T_j)|\bar{S}_i|\,|T_j| + d^2(S_i,\bar{T}_j)|S_i|\,|\bar{T}_j| + d^2(\bar{S}_i,\bar{T}_j)|\bar{S}_i|\,|\bar{T}_j| \\ \geq d_{ij}^2|V_i|\,|V_j| + \eta_{i,j}^2\left[\frac{|S_i|\,|T_j|}{1-\frac{|S_i|\,|T_j|}{|V_i|\,|V_j|}}\right] \geq d_{ij}^2|V_i|\,|V_j| + \eta_{i,j}^2|S_i|\,|T_j|\ . \end{aligned} \tag{17}$$

We have for the index of partition $\mathcal{R}$,

$$\begin{aligned} \text{ind}(\mathcal{R}) &= \frac{1}{n(n-1)}\sum_{(i,l_i)\neq(j,l_j)} d^2(V_i^{(l_i)},V_j^{(l_j)})|V_i^{(l_i)}|\,|V_j^{(l_j)}| \\ &\geq \frac{1}{n(n-1)}\sum_{i\neq j}\sum_{l_i,l_j\in\{1,2,3,4\}} d^2(V_i^{(l_i)},V_j^{(l_j)})|V_i^{(l_i)}|\,|V_j^{(l_j)}| \\ &\geq \frac{1}{n(n-1)}\sum_{i\neq j} d^2(S_i,T_j)|S_i|\,|T_j| + d^2(\bar{S}_i,T_j)|\bar{S}_i|\,|T_j| \\ &\qquad + d^2(S_i,\bar{T}_j)|S_i|\,|\bar{T}_j| + d^2(\bar{S}_i,\bar{T}_j)|\bar{S}_i|\,|\bar{T}_j|\ , \end{aligned}$$

where the first inequality follows from the fact that we are dropping some terms from the summation. The second inequality follows from Cauchy-Schwarz, and by observations such as $S_i = V_i^{(1)} \cup V_i^{(3)}$. To see why the second inequality is true, note that we have $S_i = V_i^{(1)} \cup V_i^{(3)}$ and $T_j = V_j^{(2)} \cup V_j^{(3)}$. We can conclude that $d^2(V_i^{(1)}, V_j^{(2)})|V_i^{(1)}|\ |V_j^{(2)}| + d^2(V_i^{(1)}, V_j^{(3)})|V_i^{(1)}|\ |V_j^{(3)}| + d^2(V_i^{(3)}, V_j^{(2)})|V_i^{(3)}|\ |V_j^{(2)}| + d^2(V_i^{(3)}, V_j^{(3)})|V_i^{(3)}|\ |V_j^{(3)}| \geq d^2(S_i, T_j)|S_i|\ |T_j|$ by using Cauchy-Schwarz. Similarly, we can derive the remaining terms in the RHS of the second inequality. We can proceed in the following manner by using (17):

$$\begin{aligned} \operatorname{ind}(\mathcal{R}) &\geq \frac{1}{n(n-1)} \sum_{i \neq j} \left[d_{ij}^2 |V_i|\ |V_j| + \eta_{i,j}^2 |S_i|\ |T_j| \right] \\ &= \operatorname{ind}(\mathcal{P}) + \frac{1}{n(n-1)} \sum_{i \neq j} \eta_{i,j}^2 |S_i|\ |T_j| \\ &\geq \operatorname{ind}(\mathcal{P}) + \frac{\left(\sum_{i \neq j} \eta_{i,j} |S_i|\ |T_j| \right)^2}{n(n-1) \sum_{i \neq j} |S_i|\ |T_j|}\ , \end{aligned}$$

where the last inequality follows by Cauchy-Schwarz. We have

$$\left| \sum_{i \neq j} \eta_{i,j} |S_i|\ |T_j| \right| = \left| \sum_{i \neq j} (e(S_i, T_j) - d_{ij}|S_i|\ |T_j|) \right| = \left| e(S, T) - \sum_{i \neq j} d_{ij} |S_i|\ |T_j| \right| \geq \varepsilon' n^2\ .$$

So we get

$$\operatorname{ind}(\mathcal{R}) \geq \operatorname{ind}(\mathcal{P}) + \frac{(\varepsilon' n^2)^2}{(n(n-1))^2} \geq \operatorname{ind}(\mathcal{P}) + \varepsilon'^2\ .$$

Now we shall show how to get an equitable partition $\mathcal{Q}$, which is a refinement of $\mathcal{P}$, for which the index is at least $\varepsilon'^2/2$ more. We subdivide each vertex class V_i of $\mathcal{P}$ into sets $W_{i,a}$ of size $\lfloor \varepsilon'^2 n/(7k) \rfloor$ or $\lfloor \varepsilon'^2 n/(7k) \rfloor + 1$ in such a way that all but at most three of these sets $W_{i,a}$ is completely contained inside one of $V_i^{(1)}, V_i^{(2)}, V_i^{(3)}$ or $V_i^{(4)}$. W.l.o.g, let these three sets be $W_{i,1}, W_{i,2}$ and $W_{i,3}$. We can partition these three sets further to get a partition $\mathcal{Q}^*$, which is a refinement of $\mathcal{R}$. Since $\mathcal{Q}^*$ is a refinement of $\mathcal{R}$, Cauchy-Schwarz implies that $\operatorname{ind}(\mathcal{Q}^*) \geq \operatorname{ind}(\mathcal{R})$. We shall now show that the indices of $\mathcal{Q}^*$ and $\mathcal{Q}$ are not too far apart. The only parts that differ in these partitions are $W_{i,1}, W_{i,2}$ and $W_{i,3}$, for each i. Also $|W_{i,j}| \leq \lfloor \varepsilon'^2 n/(7k) \rfloor + 1$. We get

$$\operatorname{ind}(\mathcal{Q}^*) - \operatorname{ind}(\mathcal{Q}) \leq \frac{1}{n(n-1)} \sum_{i=1}^{k} 3 \left(\frac{\varepsilon'^2 n}{7k} + 1 \right) n \leq \frac{\varepsilon'^2}{2}\ .$$

Combining, we get

$$\operatorname{ind}(\mathcal{Q}) \geq \operatorname{ind}(\mathcal{Q}^*) - \frac{\varepsilon'^2}{2} \geq \operatorname{ind}(\mathcal{R}) - \frac{\varepsilon'^2}{2} \geq \operatorname{ind}(\mathcal{P}) + \frac{\varepsilon'^2}{2} ,$$

which is what we wanted to prove.

In each refinement step, we split the classes into at most $\lfloor 7/\varepsilon'^2 + 1 \rfloor \leq 8/\varepsilon'^2$ classes $W_{i,a}$. So the new partition $\mathcal{Q}$ has size at most $8/\varepsilon'^2$ the size of $\mathcal{P}$. Also, the construction involves only the breaking up of the sets V_i using S, T. This can be performed in $O(n)$ time. □

We can now prove the main theorem.

Theorem 3.2 (Restated)**.** *Given $\varepsilon > 0$ and an n vertex graph $G = (V, E)$, one can construct in deterministic time $O\left(\frac{1}{\varepsilon^6} n^\omega \log\log n\right)$ an ε-FK-regular partition of G of order at most $2^{10^8/\varepsilon^7}$.*

Proof. If $n \leq 2^{10^8/\varepsilon^7}$, we simply return each single vertex as a separate set V_i, which is clearly ε-FK-regular for any $\varepsilon > 0$. Else, we start with an arbitrary equitable partition of vertices V. Using Corollary 3.9 we can either check that the partition is ε-FK-regular, or obtain a proof (i.e., sets S and T that violate the condition) that the partition is not $\varepsilon^3/1000$-FK-regular. Now using Theorem 3.13 (with $\varepsilon' = \varepsilon^3/1000$), we can refine the partition such that the index increases by at least $(\varepsilon^3/1000)^2/2 = \varepsilon^6/(2 \cdot 10^6)$. Since the index is upper bounded by 1, we would terminate in at most $2 \cdot 10^6/\varepsilon^6$ iterations.

The size of the partition gets multiplied by $8/\varepsilon'^2 = 8 \cdot 10^6/\varepsilon^6$ during each iteration. So the number of parts in the final partition is at most $\left(\frac{8 \cdot 10^6}{\varepsilon^6}\right)^{(2 \cdot 10^6/\varepsilon^6)}$. A quick calculation gives us that

$$\left(\frac{8 \cdot 10^6}{\varepsilon^6}\right)^{(2 \cdot 10^6/\varepsilon^6)} = 2^{\left(\log \frac{8 \cdot 10^6}{\varepsilon^6}\right) \frac{2 \cdot 10^6}{\varepsilon^6}} \leq 2^{\left(\log(8 \cdot 10^6) + \log \frac{1}{\varepsilon^6}\right) \frac{2 \cdot 10^6}{\varepsilon^6}} \leq 2^{10^8/\varepsilon^7} .$$

We need to use Corollary 3.9 a total at most $2 \cdot 10^6/\varepsilon^6$ times, and each use takes $O(n^\omega \log\log n)$ time. So the total running time is $O\left(\frac{1}{\varepsilon^6} n^\omega \log\log n\right)$. □

3.5 Concluding Remarks and Open Problems

We have designed an $\tilde{O}(n^{\omega})$ time deterministic algorithm for constructing an ε-FK regular partition of a graph. It would be interesting to see if one can design an $O(n^2)$ time deterministic algorithm for this problem. We recall that it is known [63] that one can construct an ε-regular partition of a graph (in the sense of Szemerédi) in deterministic time $O(n^2)$. This algorithm relies on a *combinatorial* characterization of ε-regularity using a co-degree condition. Such an approach might also work for ε-FK regularity, though the co-degree condition in this case might be more involved.

We have used a variant of the power iteration method to obtain an $\tilde{O}(n^{\omega})$ time algorithm for computing an approximation to the first eigenvalue of a symmetric matrix. It would be interesting to see if the running time can be improved to $O(n^2)$. Recall that our approach relies on (implicitly) running n power-iterations in parallel, each of which on one of the n standard basis vectors. One approach to design an $\tilde{O}(n^2)$ algorithm would be to show that given an $n \times n$ PSD matrix M, one can find in time $O(n^2)$ a set of $n^{0.1}$ unit vectors such that one of the vectors $\mathbf{v}$ in the set has an inner product at least $1/\mathsf{poly}(n)$ with the first eigenvector of M. If this can indeed be done, then one can replace the fast matrix multiplication algorithm for square matrices that we use in the algorithm, by an algorithm of Coppersmith [29] that multiplies an $n \times n$ matrix by an $n \times n^{0.1}$ matrix in time $\tilde{O}(n^2)$. The modified algorithm would then run in $\tilde{O}(n^2)$.

Designing an $\tilde{O}(n^2)$ algorithm for finding the first eigenvalue of a PSD matrix would of course yield an $\tilde{O}(n^2)$ algorithm for finding an ε-FK regular partition of a graph (via Theorem 3.4). In our case, it is enough to find the first eigenvalue up to a δn additive error. So another approach to getting an $\tilde{O}(n^2)$ algorithm for ε-FK regularity would be to show that in time $\tilde{O}(n^2)$ we can approximate the first eigenvalue up to an *additive* error of δn. It might be easier to design such an $\tilde{O}(n^2)$ algorithm than for the multiplicative approximation discussed in the previous paragraph.

After a preliminary version of this result appeared in RANDOM 2011, we learned that another characterization of FK-regularity had appeared in a paper of Lovász and Szegedy [69], and that one can use this characterization to design an $O(n^\omega)$ algorithm for constructing an ε-FK-regular partition of a graph. However, this characterization is different from the spectral one we obtain here. Furthermore, we are currently working on improving the spectral approach described here in order to design an optimal $O(n^2)$ algorithm for FK-regularity, so we expect the ideas presented here to be useful in future studies.

CHAPTER IV

A WOWZER TYPE LOWER BOUND FOR THE STRONG REGULARITY LEMMA

4.1 Introduction

The regularity lemma of Szemerédi asserts that one can partition every graph into a bounded number of quasi-random bipartite graphs. As we saw in Section 1.2.3, in some applications, one would like to have a strong control on how quasi-random these bipartite graphs are. Alon, Fischer, Krivelevich and Szegedy [6] obtained a powerful variant of the regularity lemma, which allows one to have an *arbitrary* control on this measure of quasi-randomness. However, their proof only guaranteed to produce a partition where the number of parts is given by the Wowzer function, which is the iterated version of the Tower function. We show here that a bound of this type is unavoidable by constructing a graph H, with the property that even if one wants a very mild control on the quasi-randomness of a regular partition, then any such partition of H must have a number of parts given by a Wowzer-type function.

Let us now formally state Szemerédi's regularity lemma. For a graph $G = (V, E)$ and two disjoint vertex sets A and B, we denote by $e_G(A, B)$ the number of edges of G with one vertex in A and one in B. The *density* $d_G(A, B)$ of the pair (A, B) in the graph G is

$$d_G(A, B) = e_G(A, B)/|A||B| \; . \tag{18}$$

That is, $d_G(A, B)$ is the fraction of pairs $(x, y) \in A \times B$ such that (x, y) is an edge of G. For $\gamma > 0$, we say that the pair (A, B) in a graph G is *γ-regular* if for any choice of $A' \subseteq A$ of size at least $\gamma|A|$ and $B' \subseteq B$ of size at least $\gamma|B|$, we have $|d_G(A', B') - d_G(A, B)| \leq \gamma$. Note that a large random bipartite graph is γ-regular for all $\gamma > 0$. Thus we can think of γ as measuring the quasi-randomness of the bipartite graph connecting A and B; the smaller γ is the more quasi-random the graph is. We will sometimes drop the subscript G in the above notations when the graph G we are referring to is clear from context.

Let $\mathcal{Z} = \{Z_1, \dots, Z_k\}$ be a partition of $V(G)$ into k sets. Throughout this chapter, we will only consider partitions into sets Z_i of equal size[1]. We will refer to each $Z \in \mathcal{Z}$ as a *cluster* of the partition $\mathcal{Z}$. The *order* of a partition is the number of clusters it has (k above). We will sometimes use $|\mathcal{Z}|$ to denote the order of $\mathcal{Z}$. We say that a partition $\mathcal{Z} = \{Z_1, \dots, Z_k\}$ *refines* another partition $\mathcal{Z}' = \{Z'_1, \dots, Z'_{k'}\}$ if each cluster of $\mathcal{Z}$ is contained in one of the clusters of $\mathcal{Z}'$.

A partition $\mathcal{Z} = \{Z_1, \dots, Z_k\}$ of $V(G)$ is said to be γ-regular if all but γk^2 of the pairs (Z_i, Z_j) are γ-regular. Szemerédi's regularity lemma can be also formulated in the following manner:

Theorem 4.1 (Szemerédi [93])**.** *For any $\gamma > 0$ and t there is an integer $K = K(t, \gamma)$ with the following property; given a graph G and a partition $\mathcal{A}$ of $V(G)$ of order t, one can find a γ-regular partition $\mathcal{B}$ of $V(G)$ which refines $\mathcal{A}$ and satisfies $|\mathcal{B}| \le K$.*

Let $T(x)$ be the function satisfying $T(0) = 1$ and $T(x) = 2^{T(x-1)}$ for $x \ge 1$. So $T(x)$ is a tower of 2's of height x. Szemerédi's proof of the regularity lemma [93] showed that the function $K(t, \gamma)$ can be bounded from above[2] by $T(1/\gamma^5)$. For a long time it was not clear if one could obtain better upper bounds for $K(t, \gamma)$. Besides being a natural problem, further motivation came from the fact that some fundamental results, such as Roth's Theorem [85, 86], could be proved using the regularity lemma. Hence improved upper bounds for $K(t, \gamma)$ might have resulted in

[1] In some papers partitions of this type are called *equipartitions*.

[2] We note that in essentially any application of Theorem 4.1, one takes t to be (at least) $1/\gamma$ so some papers simply consider the function $K'(\gamma) = K(1/\gamma, \gamma)$. The reason is that one wants to avoid "degenerate" regular partitions into a very small number of parts, where most of the graph's edges will belong to the sets V_i where one has no control on the edge distribution.

improved bounds for several other fundamental problems. In a major breakthrough, Gowers [42] proved that the tower-type dependence is indeed necessary. He showed that for any $\gamma > 0$ there is a graph where any γ-regular partition must have size at least $T(1/\gamma^{1/16})$.

Gowers' lower bound [42] can be stated as saying that if one wants a regular partition of order k, then the best quasi-randomness measure one can hope to obtain is merely $1/\log^*(k)$. Suppose however that for some $f : \mathbb{N} \mapsto (0,1)$, we would like to find a partition of a graph of order k that will be "close" to being $f(k)$-regular. Alon, Fischer, Krivelevich and Szegedy [6] formulated the following notion of being close to $f(k)$-regular.

Definition 4.2 ((ε, f)-regular partition)**.** *Let f be a function $f : \mathbb{N} \mapsto (0,1)$. An (ε, f)-regular partition of a graph G is a pair of partitions $\mathcal{A} = \{V_i \ : \ 1 \le i \le k\}$ and $\mathcal{B} = \{U_{i,i'} \ : \ 1 \le i \le k, 1 \le i' \le \ell\}$ of $V(G)$, where $\mathcal{B}$ is a refinement of $\mathcal{A}$ and the following two conditions hold:*

1. *$\mathcal{B}$ is $f(k)$-regular.*
2. *Say that a pair (V_i, V_j) of clusters of $\mathcal{A}$ is* good *if all but at most $\varepsilon\ell^2$ of pairs $1 \le i', j' \le \ell$ satisfy $|d(U_{i,i'}, U_{j,j'}) - d(V_i, V_j)| < \varepsilon$. Then, at least $(1-\varepsilon)\binom{k}{2}$ of the pairs (V_i, V_j) are good.*

One of the main results of [6] was that given a graph G and *any* function f, one can construct an (ε, f)-regular partition of G of bounded size. This version of the regularity lemma is sometimes referred to as the *strong regularity lemma.* As we have mentioned above, in order to avoid degenerate partitions we will assume henceforth that an (ε, f)-regular partition has order at least $1/\varepsilon$.

Theorem 4.3 (Strong Regularity Lemma [6])**.** *For every $\varepsilon > 0$ and $f : \mathbb{N} \mapsto (0,1)$, there is an integer $S = S(\varepsilon, f)$ such that any graph $G = (V, E)$ has an (ε, f)-regular partition $(\mathcal{A}, \mathcal{B})$ where $1/\varepsilon \le |\mathcal{A}|, |\mathcal{B}| \le S$.*

As we have already seen in Section 1.2.3, the strong regularity lemma is very useful and has been widely applied in several papers [6, 8, 10, 11, 62, 82].

Let $W(x)$ be the function satisfying $W(0) = 1$ and $W(x) = T(W(x-1))$ for $x \geq 1$. So the function W is an iterated version of the tower function $T(x)$. The function W is sometimes referred to as the Wowzer[3] function (for obvious reasons). The proof of Theorem 4.3 in [6] gave a W-type upper bound for the function $S(\varepsilon, f)$ in Theorem 4.3. As we have mentioned above, in some applications of this lemma one uses functions f that go to zero extremely fast. But in some cases, as was the case in [6], one uses moderate functions like $f(x) = 1/x^2$. However, even when the function f is $f(x) = 1/x$, the upper bound given in [6] for the function $S(\varepsilon, f)$ is (roughly) $W(1/\varepsilon)$. Hence it is natural to ask if better bounds can be obtained for such versions of Theorem 4.3. Our main result here is that a W-type dependence is unavoidable even in this case.

Theorem 4.4. *Set $f(x) = 1/x$. For every small enough $\varepsilon \leq c_0$ there is a graph H with the following property: If $(\mathcal{A}, \mathcal{B})$ is an (ε, f)-regular partition of H, and*[4] *$|\mathcal{A}| \geq 1/\varepsilon$, then $|\mathcal{A}| \geq W(\sqrt{\log(1/\varepsilon)}/100)$.*

An interesting aspect of our proof is that it gives the same lower bound even if one considers a much weaker condition than the second condition in Definition 4.2. What we show is that the lower bound of Theorem 4.3 holds even if one wants only $\varepsilon^{1/10}k^2$ of the pairs (V_i, V_j) to be good. Observe that Definition 4.2 asks[5] for $(1-\varepsilon)\binom{k}{2}$ good pairs! In other words, the lower bound holds even if one is interested in having a very weak similarity[6] between the partitions $\mathcal{A}$ and $\mathcal{B}$.

[3] This name was coined by Graham, Rothschild and Spencer [46].

[4] As we have mentioned before, in order to rule out degenerate partitions (such as taking a partition into 1 set) we assume that $|\mathcal{A}| \geq 1/\varepsilon$. A similar assumption was used in [6], where they assume that $f(x) \leq \varepsilon$. These two assumptions are basically equivalent (recall that $f(x) = 1/x$), but the one we use makes the notation somewhat simpler.

[5] We note that the application of Theorem 4.3 in [6] (as well as in most other papers) critically relied on the partition having $(1-\varepsilon)\binom{k}{2}$ good pairs.

[6] Recall the discussion following Definition 4.2.

Another interesting aspect of the proof of Theorem 4.4 is that by resetting the parameters appropriately, one can get W-type lower bounds for (ε, f)-regularity for any function $f : \mathbb{N} \mapsto (0, 1)$ going to zero faster that $1/\log^*(x)$. Observe that this is not a caveat of the proof; when $f(x) = 1/\log^*(x)$, Theorem 4.1 can be formulated as saying that any graph has an (ε, f)-regular partitions of order $T(1/\varepsilon^5)$. Hence, one cannot obtain a W-type lower bound for f of this type. So we see that even if one wants to have a very weak relation between the order of $\mathcal{A}$ and the regularity measure of $\mathcal{B}$ (say, $1/\log\log(k)$) one would still have to use a partition of size given by a W-type function[7].

The ideas we use here in order to prove Theorem 4.4 appear to be useful also for proving W-type lower bounds for the hypergraph regularity lemma [37, 43, 44, 73, 84, 94]. As we explained above, in this case also one is faced with the need to control a measure of quasi-randomness approaching 0, and this seems to be the main reason why the current bounds for this lemma are of W-type.

The rest of the chapter is organized as follows. In the following section we describe the graph H that we use in proving Theorem 4.4. In Section 4.3 we give an overview of the proof, state the two key lemmas that are needed to prove Theorem 4.4 and then derive Theorem 4.4 from them. In Section 4.4 we prove several preliminary lemmas that we would later use in the proofs of the two key Lemmas. In Sections 4.5 and 4.6 we prove the key lemmas stated in Section 4.3.

4.2 A Hard Graph for the Strong Regularity Lemma

In this section we describe the graph H that will have the properties asserted in Theorem 4.4. The description will be somewhat terse; the reader can find in Section 4.3 an overview of the proof of Theorem 4.4, which includes some intuition/motivation for the way we define H.

[7] But in such cases the bound might become $W(\log\log(1/\varepsilon))$ or some other W-type function.

4.2.1 A weighted reformulation of Theorem 4.4

Suppose P is a weighted complete graph, where each edge (x, y) is assigned a weight $d_P(x, y) \in [0, 1]$. For two sets A, B define the weighted density between A, B

$$d_P(A, B) = \sum_{x \in A, y \in B} d_P(x, y)/|A||B| \; . \tag{19}$$

Note that if we think of a graph as a weighted complete graph with 0/1 weights then the above definition coincides with the definition of $d_G(A, B)$ given in (18). Also note that when $A = \{x\}$, $B = \{y\}$ are just two vertices then $d_P(A, B)$ is just the weight $d_P(x, y)$ assigned to (x, y) as above. The following simple claim follows immediately from a standard application of Chernoff's inequality.

Claim 4.5. *Suppose P is a weighted complete graph with weights in $[0, 1]$, and H is a random graph, where each edge (x, y) is chosen independently to be included in H with probability $d_P(x, y)$. Then with probability at least $1/2$ we have*

$$|d_H(A, B) - d_P(A, B)| \leq \zeta \; ,$$

for all sets A, B of size at least $20\zeta^{-2} \log(n)$.

It is clear that we can prove Theorem 4.4 by constructing an arbitrarily large graph, such that the number of vertices n will be much larger than all the constants involved. Hence, by the above claim, we see that in order to prove Theorem 4.4 it is enough to construct a *weighted* graph H satisfying the condition of the theorem with respect to the notion of $d(A, B)$ defined in (19). The reason is that by Claim 4.5, if we have a weighted graph H satisfying Theorem 4.4, then a random graph generated as in Claim 4.5 will satisfy the assertion of of Theorem 4.4 with high probability. Therefore, from this point and *throughout this chapter* we will focus on the construction of a *weighted* graph H satisfying the condition of Theorem 4.4. Hence whenever we talk about $d(A, B)$ we will be referring to the weighted density between A, B as in (19).

4.2.2 A preliminary construction

In this subsection we describe the first step in defining the graph H of Theorem 4.4. This graph will be a variant of the graph used by Gowers in [42]. We start with the following definition.

Definition 4.6 (Balanced Partitions). *Let M be an integer and suppose we have a sequence $(A_i, B_i)_{i=1}^m$ of (not necessarily distinct) partitions of $[M]$. We call this sequence of partitions* balanced *if for any distinct $j, j' \in [M]$, the number of $1 \leq i \leq m$ for which j and j' lie in the same set of the partition (A_i, B_i) is at most $3m/4$.*

The following claim appears in [42]. For completeness, we will reproduce a simple proof later on in this chapter (see Section 4.4).

Claim 4.7. *Let $\phi(m) = 2^{\lceil m/16 \rceil}$. Then for every $m \geq 1$ there exists a sequence of m balanced partitions of $\phi(m)$.*

Let $T^{\phi}(x)$ be the function satisfying $T^{\phi}(0) = 1$ and $T^{\phi}(x) = T^{\phi}(x-1)\phi(T^{\phi}(x-1))$ for $x \geq 1$, where $\phi(x) = 2^{\lceil m/16 \rceil}$ is the function defined in Claim 4.7. It is not hard to see that T^{ϕ} it a tower-type function, and that in fact $T^{\phi}(x) \geq T(\lfloor x/2 \rfloor)$.

Let us define a sequence of integers as follows. We set

$$w(1) = \lfloor \log \log(1/\varepsilon) \rfloor \;, \tag{20}$$

and define inductively

$$w(x+1) = \lfloor \log \log(T^{\phi}(w(x))) \rfloor \;. \tag{21}$$

It is also not hard to see that $w(x)$ has a W-type dependence on x. Specifically we will later (see Section 4.4) observe that:

Claim 4.8. *For every integer $x \geq 1$, we have $w(x) \geq W(\lfloor x/2 \rfloor)$.*

We now turn to define a graph G, which we will later modify in order to get the actual graph H that will satisfy the assertion of Theorem 4.4. In order to define G we will first define a sequence of partitions of the vertex set of G. For simplicity we will identify the n vertices of G with the integers $[n]$. So let $n \in \mathbb{N}$ and set $s = w(\frac{1}{48}\sqrt{\log(1/\varepsilon)})$, where $w(x)$ is the function defined in (21). We set $m_0 = 1$ and for $1 \le r \le s$, let $m_r = m_{r-1}\phi(m_{r-1})$. For each $0 \le r \le s$, let $X_1^{(r)}, X_2^{(r)}, \ldots, X_{m_r}^{(r)}$ be a partition of $[n]$ into m_r intervals of integers of equal size[8]. We will later refer to this partition as *canonical partition* $\mathcal{P}_r$. Thus at level r, we have a canonical partition $\mathcal{P}_r$ consisting of m_r clusters. So $\mathcal{P}_0$ is just the entire vertex set of G. Note that using the notation we introduced above we have

$$|\mathcal{P}_r| = m_r = T^{\phi}(r) \ . \tag{22}$$

A crucial observation that will be used repeatedly in this chapter is that for every $r < r'$, partition $\mathcal{P}_{r'}$ refines partition $\mathcal{P}_r$.

We finally arrive at the actual definition of G. We will start with the graph G where each pair (x, y) has weight 0. We will then go over the partitions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_s$ one after the other, and in each case increase the weight between some of the pairs (x, y).

Consider some $r \ge 1$ and focus on $\mathcal{P}_r$ and $\mathcal{P}_{r-1}$. Let us simplify the notation a bit and set $m = m_{r-1}$, $M = \phi(m)$ and $m_r = Mm$. So m is the number of clusters of $\mathcal{P}_{r-1}$, M is the number of clusters of $\mathcal{P}_r$ inside each cluster of $\mathcal{P}_{r-1}$, and mM is the number of clusters of $\mathcal{P}_r$. Let us use $X_1, \ldots, X_m$ to denote the m clusters of $\mathcal{P}_{r-1}$. Also, for each $1 \le i \le m$ we use $X_{i,1}, \ldots, X_{i,M}$ to denote the M clusters of $\mathcal{P}_r$ inside X_i. Now, for each $1 \le i \le m$, let $(A'_{i,j}, B'_{i,j})_{j=1}^{m}$ be a sequence of balanced partitions of $[M]$. Such a collection exists since $M = \phi(m)$ so Claim 4.7 can be used here. One can think of each of these partitions as partitioning the clusters of $\mathcal{P}_r$ within cluster X_i.

[8] We assume that n is such that it can be divided into equal sized parts of size m_r for all $0 \le r \le s$.

Let $A_{i,j} = \cup_{t \in A'_{i,j}} X_{i,t}$ and $B_{i,j} = \cup_{t \in B'_{i,j}} X_{i,t} = X_i \backslash A_{i,j}$. We now update the weights of G as follows: If $(x, y) \in X_i \times X_j$, then we increase $d_G(x, y)$ by $4^{-r}/4^{\sqrt{\log(1/\varepsilon)}}$ if and only if $(x, y) \in A_{i,j} \times A_{j,i}$ or $(x, y) \in B_{i,j} \times B_{j,i}$. We will later refer several times to the following observation.

Fact 4.9. *For any $x, y \in V(G)$ we have $d_G(x, y) \leq 4^{-\sqrt{\log(1/\varepsilon)}}$.*

4.2.3 Adding *Traps* to G

We will now need to modify the graph G defined above in order to obtain the graph H from Theorem 4.4. To this end we will need to define certain quasi-random graphs. Let $b' < b$ and consider two of the canonical partitions $\mathcal{P}_{b'}$ and $\mathcal{P}_b$ defined in the previous subsection. Suppose $\mathcal{P}_b$ has order m_b and let V be a set of m_b vertices, where we identify vertex $i \in V$ with cluster $X_i \in \mathcal{P}_b$. Note that with this interpretation in mind, one can think of a cluster $U \in \mathcal{P}_{b'}$ as a subset of vertices $U' \subseteq V$, where vertex j belongs to U' if and only if cluster $X_j \in \mathcal{P}_b$ is a subset of U. It follows that for every $b' < b$, partition $\mathcal{P}_{b'}$ defines a natural partition of V into $m_{b'}$ subsets $U_1^{b'}, \ldots, U_{m_{b'}}^{b'}$ corresponding to its $m_{b'}$ clusters.

We now arrive at a critical definition. We will use $e(R, R')$ to denote the number of edges in a graph with one vertex in R and another in R', where edges in $R \cap R'$ are counted twice[9].

Definition 4.10 (Trap)**.** *Let $\mathcal{P}_b$, m_b, V and the partitions $U_1^{b'}, \ldots, U_{m_{b'}}^{b'}$ be as above. Let $\mathcal{O} = (V, E)$ be an m_b-vertex graph on V. Then $\mathcal{O}$ is said to be a* trap *if it satisfies the following two conditions:*

- *For every pair of sets $R, R' \subseteq V(\mathcal{O})$ of size $\lceil \sqrt{m_b}/4 \rceil$ we have*

$$\left| e(R, R') - \frac{1}{2}|R||R'| \right| \leq \frac{1}{4}|R||R'| \; .$$

[9] Note that this definition is compatible with the definition of $e(A, B)$ we used earlier, where we assumed that the sets A, B are disjoint.

- *For every $b' < b$, for every $1 \leq i, j \leq m_{b'}$, every choice of $200 \leq k \leq \log(m_b)$, every choice of $R \subseteq U_i^{b'}$ of size k^6 and every choice of $R' \subseteq U_j^{b'}$ of size $\lceil |U_j^{b'}|/k \rceil$, we have*
$$\left| e(R, R') - \frac{1}{2}|R||R'| \right| \leq \frac{1}{k^2}|R||R'| \; .$$

We will later prove the following (see Section 4.4).

Claim 4.11. *There is a constant C, such that for every $m > C$, there exists a trap on m vertices.*

We are now ready to describe the modifications needed to turn G into the graph H. We do the following for every integer $1 \leq g \leq \frac{1}{48}\sqrt{\log(1/\varepsilon)}$; let $b = w(g)$ be the integer defined in (21), let m_b be the order of $\mathcal{P}_b$ and let $\mathcal{O}_b = (V, E)$ be[10] a trap on a vertex set V of size m_b. Recall that we identify vertex $i \in V$ with cluster $X_i \in \mathcal{P}_b$. We now modify G as follows; for every pair of clusters (X_i, X_j), if $(i, j) \in E(\mathcal{O}_b)$ we increase by 4^{-g} the weight of every pair of vertices $(x, y) \in X_i \times X_j$. If $(i, j) \notin E(\mathcal{O}_b)$ we do not increase the weight of (x, y). Let us state the following fact to which we will later refer.

Fact 4.12. *The smallest weight used when placing a trap in H is $4^{-\frac{1}{48}\sqrt{\log(1/\varepsilon)}}$.*

Later on in this chapter we will say that we have *placed* a trap on partition $\mathcal{P}_b$ if b is one of the integers $w(1), \ldots, w(\frac{1}{48}\sqrt{\log(1/\varepsilon)})$. If a trap was placed on $\mathcal{P}_b$ and (i, j) is an edge of the graph $\mathcal{O}_b$ that was used in the previous paragraph, then we will say that the pair (X_i, X_j) *belongs* to the trap placed on $\mathcal{P}_b$. Also, if $b = w(g)$, then we will refer to the trap placed on $\mathcal{P}_b$ as the g^{th} trap placed in H. Finally, if $(x, y) \in X_i \times X_j$ and (X_i, X_j) belong to the trap placed on $\mathcal{P}_{w(g)}$ then we will say that (x, y) *received* an extra weight of 4^{-g} from the g^{th} trap placed in H.

[10] Note that since we only ask Theorem 4.4 to hold for small enough ε, we can assume that ε is small enough so that already $m_{w(1)} = T^{\phi}(w(1))$ would be larger than C, thus allowing us to pick a trap via Claim 4.11 (where $w(1)$ is defined in (20)).

Using the above jargon, we can thus say that in order to obtain the graph H from the graph G we do the following for every $1 \leq g \leq \frac{1}{48}\sqrt{\log(1/\varepsilon)}$; setting $b = w(g)$, we place the g^{th} trap on partition $\mathcal{P}_b$, by increasing the weight of (x, y) by 4^{-g} if and only if $(x, y) \in X_i \times X_j$ and (X_i, X_j) belong to the trap.

Let us draw some distinction between the way we assigned weights to edges in G and the way we have done so when modifying G to obtain H. When defining G we looked at *each* of the partitions $\mathcal{P}_r$, and for *every* $X_i, X_j \in \mathcal{P}_{r-1}$ added weight $4^{-r}/4^{\sqrt{\log(1/\varepsilon)}}$ only to *some* of the pairs $(x, y) \in X_i \times X_j$. More specifically, we considered the partitions of $X_i = A_{i,j} \cup B_{i,j}$ and $X_j = A_{j,i} \cup B_{j,i}$ and only added the weight $4^{-r}/4^{\sqrt{\log(1/\varepsilon)}}$ when either $(x, y) \in A_{i,j} \times A_{j,i}$ or $(x, y) \in B_{i,j} \times B_{j,i}$. When adding the traps, we have only added weights to *some* of the partitions $\mathcal{P}_b$, that is, those for which $b = w(g)$ for some $1 \leq g \leq \frac{1}{48}\sqrt{\log(1/\varepsilon)}$. Moreover, when placing a trap on $\mathcal{P}_b$ we added weight 4^{-g} only to pairs (x, y) connecting *some* of the pairs (X_i, X_j) (those that belong to the trap). Finally, for each such pair (X_i, X_j) we either added more weights to all the pairs $(x, y) \in X_i \times X_j$ or to none of them.

Another important distinction is the following; suppose $b = w(g)$. Then in G, the weight that was added to $\mathcal{P}_b$ was $4^{-b}/4^{\sqrt{\log(1/\varepsilon)}}$ while the weight we added when placing a trap on $\mathcal{P}_b$ is 4^{-g}. Since w is a W-type function we see that the weights assigned in G to a specific partition $\mathcal{P}_b$ are extremely small compared to those assigned to $\mathcal{P}_b$ when placing a trap on it (assuming a trap was placed on $\mathcal{P}_b$).

We also observe that for every pair of vertices (x, y) of H, the total weight it can receive from all the traps we placed is bounded by $1/4 + 1/16 + \ldots < 1/3$. We also recall Fact 4.9 stating that the total weight assigned to a pair (x, y) in G is bounded by $1/4^{\sqrt{\log(1/\varepsilon)}}$. This means that $d_H(x, y) \leq 1$, as needed for the application of Claim 4.5.

4.3 Proof Overview, Key Lemmas and Proof of Theorem 4.4

Our goal in this section is fourfold; give an overview of the proof of Theorem 4.4, describe the main intuition behind the construction of H, state the two key lemmas that will be used to prove Theorem 4.4 and finally derive Theorem 4.4 from these two lemmas.

Perhaps the best way to approach our construction of H is to first consider the proof of Theorem 4.3 in [6]. For simplicity, let us consider the case $f(x) = 1/x$; we start by taking $\mathcal{A}_1$ to be an arbitrary partition of G of order $1/\varepsilon$, and then apply Theorem 4.1 in order to find a $1/|\mathcal{A}_1|$-regular partition, $\mathcal{B}_1$, of G that refines $\mathcal{A}_1$. Note that by definition, $\mathcal{A}_1$ and $\mathcal{B}_1$ satisfy the first condition of Definition 4.2, so if they also satisfy the second, then we are done. If they do not, then we set $\mathcal{A}_2$ to be $\mathcal{B}_1$ and use Theorem 4.1 to find a $1/|\mathcal{A}_2|$-regular partition, $\mathcal{B}_2$, of G which refines $\mathcal{A}_2$. Note that $\mathcal{A}_2$ and $\mathcal{B}_2$ satisfy the first property, so if they satisfy the second we are done. The process thus goes on till we end up with a pair of partitions $\mathcal{A}_i$, $\mathcal{B}_i$ that satisfy the second condition. The main argument in [6] shows that this process must stop after (about) $1/\varepsilon$ steps with a pair $\mathcal{A}_i$, $\mathcal{B}_i$ that satisfies the second condition, and also (by definition) the first condition. Since the above proof applies Theorem 4.1 repeatedly, where each time we take $1/\gamma$ to be the order of the previous partition, the bound we obtain is of W-type.

Of course, if we want to have any chance of proving Theorem 4.4, we need to come up with a graph for which the *proof* of Theorem 4.3 will produce a partition of W-size. Given the overview of this proof described above, the graph H needs to have two properties: (1) For every $\gamma > 0$, any γ-regular partition of H has size given by a tower-type function; (2) one needs to iteratively apply Theorem 4.1 a super-constant[11] number of times in order to get two partitions $\mathcal{A}$ and $\mathcal{B}$ satisfying the

[11] To be precise, in order to get a W-type lower bound the number of iterations needs to be larger

second condition of Definition 4.2. The first property will guarantee that each time we apply Theorem 4.1 we get a tower-type increase in the size of $\mathcal{A}_i$ while the second condition will guarantee that we will have to repeat this sufficiently many times.

Let us describe how to get a graph satisfying property (1) mentioned above. Recall that Gowers showed [42] that for every γ there exists a graph with the property that any γ-regular partition has a size $T(1/\gamma^{1/16})$. It is not hard to see that by a minor "tweak" of his construction[12] one can get a *single* graph that works for all γ bounded away from 0. This is basically[13] the graph G we defined in Subsection 4.2.2. For completeness let us describe the intuition behind Gowers' construction. Let us explain why the partitions $\mathcal{P}_r$ used in the construction of G cannot be used as γ-regular partitions of G. Recall that at each iteration, we take every pair of sets $X_i, X_j \in \mathcal{P}_{r-1}$ split them as $X_i = A_{i,j} \cup B_{i,j}$ and $X_j = A_{j,i} \cup B_{j,i}$ and increase the weight between $A_{i,j}, A_{j,i}$ and $B_{i,j}, B_{j,i}$. So, in some sense, each partition $\mathcal{P}_r$ is used in order to rule out the possibility of using the previous partition $\mathcal{P}_{r-1}$ as a γ-regular partition. We note that when one comes about to actually prove that no other (small) partition can be γ-regular one relies critically on the fact that the weights assigned to the partitions $\mathcal{P}_r$ in G decrease exponentially (as a function of r). This makes sure that any irregularity found in level r cannot be canceled by weights assigned to levels $r' > r$.

Let us describe how to get a graph satisfying property (2) mentioned above. Recall that G was defined over a sequence of partitions $\mathcal{P}_r$. Suppose we want to make sure that two specific partitions in this sequence $\mathcal{P}_r$ and $\mathcal{P}_{r'}$, with $\mathcal{P}_{r'}$ refining $\mathcal{P}_r$, will not satisfy the second property of Definition 4.2. Then we can do the following; we

than $W^{-1}(1/\varepsilon)$.

[12] In fact, we will be tweaking the construction of Gowers [42] that gives a slightly weaker lower bound of $T(\log(1/\gamma))$, and is much simpler to analyze. Since we are trying to prove W-type lower bounds it makes little difference if we are iterating the function $T(x)$ or $\log(T(x))$.

[13] If we were only interested in getting a graph that for all $\gamma > 0$ had only γ-regular partitions of Tower-size, then we could have used the weights 4^{-r} instead of $4^{-r}/4^{\sqrt{\log(1/\varepsilon)}}$ like we do.

take a random graph $\mathcal{O}$ whose vertices are the clusters of $\mathcal{P}_{r'}$, and for every edge $(i', j') \in E(\mathcal{O})$ increase the weight of all pairs $(x, y) \in U_{i'} \times U_{j'}$, where $U_{i'}, U_{j'} \in \mathcal{P}_{r'}$. This is just the trap we used in Subsection 4.2.3. Since we use a random graph, we expect *all* pairs of clusters (X_i, X_j) of $\mathcal{P}_r$ to not be good (in the sense of Definition 4.2) since close to half of the clusters $(U_{i'}, U_{j'})$ with $U_{i'} \subseteq X_i, U_{j'} \subseteq X_j$, will get an extra weight while the other half will not. Now it is not hard to see that for this to work we do not actually have to put the trap on $\mathcal{P}_{r'}$; it is enough to do that on some partition $\mathcal{P}_b$ with $r \leq b \leq r'$ them. Since we will make sure that a γ-regular partition must be huge, in order to satisfy the first condition of Definition 4.2 one would have to pick two partitions $\mathcal{P}_{r'}$, $\mathcal{P}_r$ with r' being much larger than r. Therefore, in order to make sure that all pairs $\mathcal{P}_{r'}$, $\mathcal{P}_r$ will fail the second condition, it is enough to place the traps only on very few partitions $\mathcal{P}_b$, where by few we mean that there will be a tower-type jump between their indices.

So with one serious caveat, if one wants to construct an (ε, f)-regular partition by taking $\mathcal{A}$ and $\mathcal{B}$ to be two of the canonical partitions $\mathcal{P}_r$,$\mathcal{P}_{r'}$, then one is forced to take two partitions that refine the last trap we have placed in H. The reason is that by property (1) the integers r and r' must be very far apart, and the way we have placed the traps will guarantee that there will be a trap in between them that will then make sure that they do not satisfy the second property of Definition 4.2. The caveat we are referring to is the fact that once we have added the traps to G, we have destroyed the critical feature of the graph G, which is that the weights decrease exponentially (recall the observation we made above and the discussion at the end of Subsection 4.2.3). Hence, it is no longer true that once we find a discrepancy in some partition $\mathcal{P}_r$, this discrepancy cannot be canceled by lower levels. In terms of analyzing Gowers' example, it might be the case that some pairs that were not γ-regular in G, might become γ-regular in H. Actually, there *will* be such pairs. This might completely ruin our ability to prove the H has only γ-regular partitions of tower-size.

We overcome the above problem by proving that it cannot happen *very often*. Namely, since the trap we have added originates from a random graph, then at least on average we expect it to contribute the same density to all pairs of vertex sets. So *on average*, we do not expect a trap to cancel a discrepancy caused by partitions that are refined by it. This is of course only true on average. To turn this into a deterministic statement, we formulate a condition that holds in random graphs, and show that if too many pairs that were supposed not to be γ-regular somehow turn out to be γ-regular, then we get a violation of the property we assume the trap to satisfy. Turning this intuition into formality is probably the most challenging part of this chapter. One of the main reasons is that we cannot run this argument over all the pairs; instead we need to somehow "pack" them together and then argue about each of these packaged pairs. See Lemmas 4.25 and 4.26.

We now turn to the key lemmas of this chapter. To state them we will need to define the notion of β-refinement. We briefly mention that this notion is crucial in overcoming another assumption we have used in the above discussion, that one is trying to construct an (ε, f)-regular partitions by using only the canonical partitions $\mathcal{P}_r$. Using the notion of β-refinement we will show that one actually has to *approximately* use only such partitions.

Let $0 \leq \beta < 1/2$. Given two sets Z and X, we write $Z \subset_\beta X$, to denote the fact that $|Z \cap X| \geq (1-\beta)|Z|$. We will sometimes also say that X β-contains Z or that Z is β-contained in X to refer to the fact that $Z \subset_\beta X$. Note that since we assume that $\beta < 1/2$, there can be at most one set X that β-contains a set Z. Given two partitions $\mathcal{P} = \{X_1, \ldots, X_m\}$ and $\mathcal{Z} = \{Z_1, \ldots, Z_k\}$ of $V(H)$ and $\beta > 0$, we shall say that $\mathcal{Z}$ is a β-refinement of $\mathcal{P}$ if for at least $(1-\beta)k$ values of t, there exists i such that $Z_t \subset_\beta X_i$. Observe that if $\beta = 0$, then β-refinement coincides with the standard notion of one partition refining another one, that we discussed earlier.

In what follows, when we refer to the graph H we mean the graph H defined in the previous section. We now state the two key lemmas we will prove later on in this chapter. Getting back to the intuitive discussion above, one can think of the first lemma as formalizing condition (1) mentioned above, which we wanted H to satisfy.

Lemma 4.13. *Let $f(x) = 1/x$. Suppose $\mathcal{A}$ and $\mathcal{B}$ form an (ε, f)-regular partition of H. If $|\mathcal{A}| = k \geq 1/\varepsilon$ then $\mathcal{B}$ is an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_{2\log\log k}$.*

Note that if $\beta < 1/2$ and partition $\mathcal{A}$ is a β-refinement of $\mathcal{P}_r$ then the order of $\mathcal{A}$ is at least as large as the order of $\mathcal{P}_r$. Hence the above lemma says (implicitly) that partition $\mathcal{B}$, which must be $1/k$-regular, must have order as large as that of $\mathcal{P}_{2\log\log k}$. Recalling (22), this means that $|\mathcal{B}| \geq T^{\phi}(\log\log k)$. We note however, that knowing that $\mathcal{B}$ must have tower size is not enough for our proof to work. We actually need to know that $\mathcal{B}$ is a good refinement of partition $\mathcal{P}_{2\log\log k}$. This is needed in order to show that if a trap was placed between $\mathcal{A}$ and $\mathcal{B}$ then they will indeed fail to satisfy the second property of Definition 4.2. This is exactly where the notion of β-refinement becomes useful, as we state in the second key lemma, that formalizes property (2) mentioned above that we wanted H to satisfy.

Lemma 4.14. *Suppose $\mathcal{A}$, $\mathcal{B}$ are two partitions of H with the following properties*

- *$\mathcal{B}$ is a refinement of $\mathcal{A}$.*
- *$|\mathcal{A}| = k$ and H has a trap on a canonical partition $\mathcal{P}_b$ whose order is at least k^2.*
- *$\mathcal{B}$ is an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_b$.*

Then $\mathcal{A}$ and $\mathcal{B}$ do not satisfy the second condition of Definition 4.2. In particular they do not form an (ε, f)-regular partition of H.

We end this section with the derivation of Theorem 4.4 from Lemma 4.13 and Lemma 4.14.

Proof of Theorem 4.4. Suppose $\mathcal{A}$ and $\mathcal{B}$ form an (ε, f)-regular partition of H, where $|\mathcal{A}| = k \geq 1/\varepsilon$. Let m_s denote the order of $\mathcal{P}_s$, which is the largest partition on which we have placed a trap. Recall that $s = w(\frac{1}{48}\sqrt{\log(1/\varepsilon)})$ and that $m_s \geq s$ (In fact, $m_s = T^{\phi}(s)$). Hence, by Claim 4.8 we have $m_s \geq W(\frac{1}{96}\sqrt{\log(1/\varepsilon)})$. Therefore, if $k \geq \sqrt{m_s}$ we are done since $\sqrt{W(\frac{1}{96}\sqrt{\log(1/\varepsilon)})} > W(\frac{1}{100}\sqrt{\log(1/\varepsilon)})$ (with a lot of room to spare).

We can thus assume that $|\mathcal{A}| = k \leq \sqrt{m_s}$, and choose b to be the *smallest* index of a partition $\mathcal{P}_b$, on which we have placed a trap satisfying $|\mathcal{P}_b| \geq k^2$. If we could show that $\mathcal{B}$ forms an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_b$, then an application of Lemma 4.14 would give that $\mathcal{A}$ and $\mathcal{B}$ do not form an (ε, f)-regular partition of H, which would be a contradiction. Now, Lemma 4.13 tells us that $\mathcal{B}$ is an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_{2\log\log k}$. Note that if $\mathcal{B}$ is an $\varepsilon^{1/5}$-refinement on $\mathcal{P}_{2\log\log k}$ then it is also an $\varepsilon^{1/5}$-refinement of any partition that is refined by $\mathcal{P}_{2\log\log k}$. In other words, it is enough[14] that we show that $b \leq 2\log\log(k)$.

Suppose first that $b = w(1)$, that is, the first trap of size at least k^2 is the first trap placed in H. Then recalling (20) and the fact that $k \geq 1/\varepsilon$, we have

$$b = w(1) = \lfloor \log\log(1/\varepsilon) \rfloor \leq 2\log\log(k) \ ,$$

as needed. Suppose now that $b = w(g+1)$ for some $g \geq 1$ and that the trap with largest order smaller than k^2 was placed on $\mathcal{P}_{b'}$ where $b' = w(g)$. Then recalling (21) we see that $b = \lfloor \log\log(T^{\phi}(b')) \rfloor$. We also recall (22) stating that $|\mathcal{P}_{b'}| = T^{\phi}(b')$. We thus infer that

$$T^{\phi}(b') = |\mathcal{P}_{b'}| \leq k^2 \ ,$$

implying that

$$b = \lfloor \log\log(T^{\phi}(b')) \rfloor \leq \log\log(k^2) \leq 2\log\log(k) \ ,$$

thus completing the proof. □

[14] Recall that each partition $\mathcal{P}_r$ is a refinement of all the partitions $\mathcal{P}_{r'}$ with $r' \leq r$.

As one can see from our proof of Theorem 4.4, what we show is not only that an (ε, f)-regular partition must be large, but that the only way to get such a partition it to basically take $\mathcal{A}$ and $\mathcal{B}$ to be refinements of partition $\mathcal{P}_s$ in H. Recall that we started this section by saying that one should design H in a way that will make sure that at least the proof of Theorem 4.3 will produce a large partition. The fact that the only way to get an (ε, f)-regular partition is to take partition $\mathcal{P}_s$, can be interpreted as saying that the only way to *prove* Theorem 4.3 is to go through the process described at the beginning of this section.

4.4 Some Preliminary Lemmas

In this section we prove some simple lemmas that will be used later on in this chapter. But we start with proving the claims that were stated without proof in the previous sections. From this point on, when we write something like $x \leq_{(20)} y$, we mean that the fact that $x \leq y$ follows from the facts stated in equation (20). As the reader will inevitably notice, we will be very loose in many of the proofs. The main reason is that as we are dealing with W-type and Tower-type functions, many "improvements" have absolutely no difference even on the quantitative bounds one obtains. Hence, we opted for statements that are simpler to state and apply.

Proof of Claim 4.7. First, notice that for any $m \geq 1$, we can choose $M = 2$. Indeed, we can simply repeat the partition $A_i = \{1\}, B_i = \{2\}$, a total of m times to get m partitions where there is no i for which (distinct) j, j' appear in the same set. So the claim holds for $1 \leq m \leq 16$.

Suppose now that $m \geq 17$, set $M = 2^{\lceil m/16 \rceil}$ and consider a randomly generated sequence $(A_i, B_i)_{i=1}^{m}$ of partitions of $[M]$ obtained as follows; for each $1 \leq i \leq m$ and each $1 \leq j \leq M$ we assign element j to A_i with probability $1/2$ (all mM choices being independent). Fix a pair of distinct elements $j, j' \in [M]$. Clearly the number of i such that j, j' belong to the same class in (A_i, B_i) is distributed as the binomial random variable $B(m, 1/2)$. Hence, we get from a standard application of Chernoff's inequality that the probability that the number of these i is larger than $3m/4$ is bounded by $e^{-m/6}$. Hence, the probability that some pair of distinct $j, j' \in [M]$ belong to the same part in more than $3m/4$ of the partitions is bounded by $\binom{M}{2}e^{-m/6} < 1$ so the required sequence of partitions exists. $\square$

Proof of Claim 4.8. Let us start by proving that

$$T^{\phi}(x) \geq T(\lfloor x/2 \rfloor) \; , \tag{23}$$

as we have previously claimed. We first notice that when $x \geq 256$ we have $2^{x/16} \geq 16x$, implying that in this case we have

$$\phi(\phi(t)) \geq 2^{2^{t/16}/16} \geq 2^t \; . \tag{24}$$

Now, one can verify that (23) holds when $1 \leq x \leq 10$ and that $T(x) \geq 256$ when $x \geq 4$. Thus, when $x \geq 11$, we have

$$T^{\phi}(x) \geq \phi(\phi(T^{\phi}(x-2))) \geq_{(23)} \phi(\phi(T(\lfloor x/2 \rfloor - 1))) \geq_{(24)} 2^{T(\lfloor x/2 \rfloor -1)} = T(\lfloor x/2 \rfloor) \; .$$

We now recall (20) which implies that since we can assume that ε is small enough, we can also assume that $w(1)$ is large enough. In particular we have $w(1) \gg W(1) = T(1) = 2$. Let us denote $\hat{T}(t) = \lfloor \log\log(T^{\phi}(t)) \rfloor$. So $w(i)$ is just $\hat{T}$ iterated i times with $w(1) = \lfloor \log\log(1/\varepsilon) \rfloor$. Now we shall show that for any large enough t, $\hat{T}(\hat{T}(t)) > T(t)$. Using induction, it would follow that for all $i \geq 1$, $w(i) > W(\lfloor i/2 \rfloor)$, thus completing the proof. Now

$$\begin{aligned}
\hat{T}(\hat{T}(t)) &= \lfloor \log\log(T^{\phi}(\lfloor \log\log(T^{\phi}(t))\rfloor))\rfloor \\
&\geq \frac{1}{4}\log\log\left(T\left(\frac{1}{4}\log\log\left(T\left(t/4\right)\right)\right)\right) \\
&\geq \frac{1}{4}T\left(\frac{1}{4}T\left(t/4-2\right)-2\right) \\
&\geq \frac{1}{4}T\left(\frac{1}{5}T\left(\frac{t}{5}\right)\right) \\
&\geq T(t)\ ,
\end{aligned}$$

where in the first inequality we apply (23), in the second we use the fact that $\log\log(T(x)) = T(x-2)$, and the last holds for all large enough t. □

We now turn to the proof of Claim 4.11. Recall that given two sets of vertices R, R', which are not necessarily disjoint, we used $e(R, R')$ to denote the number of edges connecting a vertex in R to a vertex in R', where an edge belonging to $R \cap R'$ is counted twice.

Claim 4.15. *There is a constant C, such that if $m = m_b \geq C$ and $\mathcal{O}$ is a random graph from $G(m, 1/2)$, then with probability at least $3/4$ it satisfies the first condition of a trap (as stated in Definition 4.10).*

Proof. Fix two sets R, R' of size $r = \lceil \sqrt{m}/4 \rceil$. Given distinct i, i' let $z_{i,i'}$ be the indicator for the event that $(i, i') \in E(\mathcal{O})$, and $z_{R,R'} = \sum_{i \in R, i' \in R'} z_{i,i'}$. Then,

$$\frac{3r^2}{8} \leq \binom{r}{2} \leq \mathbb{E}[z_{R,R'}] = \mathbb{E}[e(R, R')] = \frac{1}{2}\left(r^2 - |R \cap R'|\right) \leq \frac{r^2}{2}\ ,$$

for all large enough m. Now observe that $z_{R,R'}$ is a sum of at least $\binom{r}{2}$ indicators $z_{i,i'}$ and each $z_{i,i'}$ can change the value of $z_{R,R'}$ by at most 2. We thus get from a standard application of Chernoff's inequality that

$$\mathbb{P}\left[\left|e(R, R') - \frac{1}{2}r^2\right| \geq \frac{1}{4}r^2\right] \leq \mathbb{P}\left[|z_{R,R'} - \mathbb{E}[z_{R,R'}]| \geq \frac{1}{8}r^2\right] \leq e^{-\frac{r^2}{100}}\ .$$

Hence the probability that there is any pair of sets R, R' satisfying $|e(R,R') - \frac{1}{2}r^2| > \frac{1}{4}r^2$ is at most

$$\binom{m}{r}^2 2^{-\frac{1}{100}r^2} \leq m^{\sqrt{m}} e^{-m/1600} \ll 1/4 \; ,$$

for all large enough m. □

Claim 4.16. *There is a constant C, such that if $m = m_b \geq C$ and $\mathcal{O}$ is a random graph from $G(m, 1/2)$, then with probability at least $3/4$, it satisfies the second condition of a trap (as stated in Definition 4.10).*

Proof. Let us start by considering the case $b' = b - 1$. Suppose $U_1, \ldots, U_{m_{b-1}}$ is the partition of $V(\mathcal{O})$ induced by the partition $\mathcal{P}_{b-1}$ (as discussed prior to Definition 4.10). Now recall (see Subsection 4.2.2) that the integers m_b satisfy the relation

$$m = m_b = m_{b-1}\phi(m_{b-1}) = m_{b-1}2^{\lceil m_{b-1}/16 \rceil} \; .$$

This means that

$$\log(m) \leq m_{b-1} \leq 17\log(m) \; , \tag{25}$$

so the size of the sets U_i, which we will denote by h_{b-1}, satisfies

$$m/17\log(m) \leq h_{b-1} = m/m_{b-1} \leq m/\log(m) \; . \tag{26}$$

Fix now two sets U_i, U_j, an integer $200 \leq k \leq \log(m)$, a subset $R \subseteq U_i$ of size k^6 and a subset $R' \subseteq U_j$ of size $\lceil h_{b-1}/k \rceil$. Given distinct i, i' with $i \in R$ and $i' \in R'$ let $z_{i,i'}$ be the indicator for the event that $(i, i') \in E(\mathcal{O})$, and $z_{R,R'} = \sum_{i \in R, i' \in R'} z_{i,i'}$. Then

$$\begin{aligned} \frac{|R||R'|}{2} \geq \mathbb{E}[z_{R,R'}] = \mathbb{E}[e(R,R')] &= \frac{1}{2}\left(|R||R'| - |R \cap R'|\right) \\ &\geq \frac{1}{2}|R||R'| - \frac{1}{2}|R| \\ &\geq \left(\frac{1}{2} - \frac{1}{2k^2}\right)|R||R'| \; . \end{aligned}$$

where in the last inequality we use the facts that $k \leq \log(m)$, that $|R'| = h_{b-1} \geq_{(26)} m/17\log(m)$ and that we can pick m to be large enough so that $|R'| \geq k^2$.

Note that $z_{R,R'}$ is a sum of at least $|R|(|R'|-|R|) \geq |R||R'|/2$ indicators $z_{i,i'}$ (we are using the fact that $|R| \ll |R'|$). Since each of them can change $z_{R,R'}$ by at most 2, we get from Chernoff's inequality, the fact that $k \geq 200$ and the estimate for $\mathbb{E}[z_{R,R'}]$ from the previous paragraph that

$$\begin{aligned}
\mathbb{P}\left[\left|e(R,R') - \frac{1}{2}|R||R'|\right| \geq \frac{1}{k^2}|R||R'|\right] &\leq \mathbb{P}\left[|z_{R,R'} - \mathbb{E}[z_{R,R'}]| \geq \frac{1}{2k^2}|R||R'|\right] \\
&\leq e^{-\frac{|R||R'|}{64k^4}} \\
&\leq e^{-kh_{b-1}/64} \\
&\leq e^{-2h_{b-1}} \,.
\end{aligned}$$

Now, there are $m_{b-1}^2 = O(\log^2(m))$ ways to pick the sets U_i, U_j, $O(\log(m))$ ways to choose k, $\binom{h_{b-1}}{k^6}$ ways to pick R and $\binom{h_{b-1}}{h_{b-1}/k}$ ways to pick R'. Overall, we get from a union bound that the probability that some choice of U_i, U_j, k, R and R' will violate the second condition of Definition 4.10 is bounded by

$$O(\log^3 m)\binom{h_{b-1}}{k^6}\binom{h_{b-1}}{h_{b-1}/k}e^{-2h_{b-1}} \leq m^{2k^6}(ek)^{h_{b-1}/k}e^{-2h_{b-1}} \leq m^{2\log^6(m)}e^{-h_{b-1}} \,, \quad (27)$$

where in the first inequality we use the inequality $\binom{n}{k} \leq (en/k)^k$ and in the second the fact that $k \leq \log(m)$.

Let us now consider an arbitrary $b' < b$. Note that since $m_{b'} \leq m_{b-1}$, we still have $m_{b'} \leq 17\log(m)$. Hence there are still only $O(\log^2(m))$ many ways to choose the sets $U_i^{b'}, U_j^{b'}$. This means that the upper bound obtained in (27) for the probability of partition $\mathcal{P}_{b-1}$ violating the condition applies to any given partition $\mathcal{P}_{b'}$, with h_{b-1} replaced by $h_{b'}$. But since $h_{b'} \geq h_{b-1}$ the bound in (27) still holds.

We finally recall (22) stating that $m_b = T^{\phi}(b)$. As we noted in (23) we have $T^{\phi}(b) > T(\lfloor b/2 \rfloor)$. Hence the number of $b' < b$ we need to consider is only $O(\log^*(m))$. So combining this fact with the discussion in the previous paragraph we get that the probability of any partition $\mathcal{P}_{b'}$ violating the second condition of Definition 4.10 is bounded by

$$m^{3\log^6(m)}e^{-h_{b-1}} \ll 1/4$$

where we apply the fact that $h_{b-1} \geq m/17\log(m)$, stated in (26). □

Proof of Claim 4.11. Follows immediately from Claims 4.15 and 4.16. □

We will now prove two lemmas that will somewhat streamline the application of the properties of traps later on in this chapter. Both lemmas will rely on the observation stated in Lemma 4.17 below. In what follows, we use $v_S \in \mathbb{R}^n$, with $S \subseteq [n]$ to denote the vector whose i^{th} entry is $1/|S|$ when $i \in S$ and 0 otherwise. Let $\mathcal{V}_k = \{v_S : S \subseteq [n], |S| = k\}$.

Lemma 4.17. *If $x \in [0, 1/k]^n$ and $\sum x_i = 1$, then x is a convex combination of the vectors of $\mathcal{V}_k$.*

Before we prove this lemma, we need a standard theorem from linear programming theory, which we state without proof. A polyhedron $P \subseteq \mathbb{R}^n$ is the set of points satisfying a finite number of linear inequalities. P is *bounded* if there is a constant C such that $\|x\| \leq C$ for all $x \in P$. Finally, a point $x \in P$ is said to be a *vertex* of P if it cannot be represented as a proper convex combination of points $x', x'' \in P$.

Theorem 4.18 ([14])**.** *For every bounded polyhedron $P \subseteq \mathbb{R}^n$ and $x \in P$, the point x can be written as a convex combination of the vertices of P.*

Proof of Lemma 4.17. Consider the polyhedron

$$P = \left\{ x \ : \ \sum_i x_i = 1, \text{ and } 0 \leq x_1, \ldots, x_n \leq 1/k \right\} .$$

Notice that for all $x \in P$, we have $\|x\| \leq 1$. Let $\mathcal{V}$ be the set of vertices of P. By Theorem 4.18, we have that any $x \in P$ is a convex combination of $\mathcal{V}$. So we need to show that[15] $\mathcal{V} \subseteq \mathcal{V}_k$.

Suppose $u \in \mathcal{V}$. If all its entries are either 0 or $1/k$ it obviously belongs to $\mathcal{V}_k$. So suppose that u has an entry $u_i \in (0, 1/k)$. Then there exists at least one

[15] We clearly have $\mathcal{V}_k \subseteq \mathcal{V}$ but this direction is not needed.

more entry $u_j \in (0, 1/k)$, because otherwise the entries cannot sum to 1. Let $\varepsilon_u = \frac{1}{2}\min\{u_i, u_j, 1/k - u_i, 1/k - u_j\}$. Let e_i denote the canonical basis vector where the ith entry is 1 and all the other entries are 0. Similarly define e_j. Let $u' = u + \varepsilon_u e_i - \varepsilon_u e_j$ and $u'' = u - \varepsilon_u e_i + \varepsilon_u e_j$. It can be checked that both $u', u'' \in P$ and that $u' + u'' = 2u$. So u can be written as the convex combination of two other vectors in P, which means that u is not a vertex of P. □

We now turn to prove two lemmas. The first one will help us in applying the first property of traps in proving Lemma 4.14, while the second one will help us in applying the second property of traps in proving Lemma 4.13.

Lemma 4.19. *Suppose $\mathcal{O}$ is the graph that was used when defining the trap on partition $\mathcal{P}_b$ (so $|V(\mathcal{O})| = m_b$ and we can assume that $\mathcal{O}$ satisfies the first condition of Definition 4.10). Let Q be the adjacency matrix of $\mathcal{O}$, and suppose $x, y \in [0,1]^{m_b}$ satisfy $\sum x_i = \sum y_i = g \geq \sqrt{m_b}/2$. Then we have*

$$\left| x^T Q y - \frac{1}{2} g^2 \right| \leq \frac{1}{4} g^2 \ .$$

Proof. The vectors x/g and y/g satisfy the condition of Lemma 4.17 with $k = \lceil \sqrt{m_b}/4 \rceil$. Hence we can express x/g and y/g as convex combinations of the vectors of $\mathcal{V}_k$ as $x/g = \sum_R a_R v_R$ and $y/g = \sum_{R'} b_{R'} v_{R'}$. Observe further that $(v_R)^T Q v_{R'} = e(R, R')/|R||R'|$. Since $|R| = |R'| = k = \lceil \sqrt{m_b}/4 \rceil$ and we assume that $\mathcal{O}$ satisfies the first condition of being a trap, we can infer that for any R and R' we have

$$1/4 \leq (v_R)^T Q v_{R'} \leq 3/4 \ . \tag{28}$$

We can thus infer from (28) and the fact that $\sum_R a_R v_R$ and $\sum_{R'} b_{R'} v_{R'}$ are convex combinations that

$$
\begin{aligned}
(x/g)^T Q(y/g) &= \left(\sum_R a_R v_R\right)^T Q \left(\sum_{R'} b_{R'} v_{R'}\right) \\
&= \sum_{R,R'} a_R b_{R'} (v_R)^T Q v_{R'} \\
&\le \frac{3}{4} \sum_{R,R'} a_R b_{R'} \\
&= \frac{3}{4} \,,
\end{aligned}
$$

implying that $x^T Qy \le \frac{3}{4}g^2$. An identical argument gives $x^T Qy \ge \frac{1}{4}g^2$, which completes the proof. □

Lemma 4.20. *Suppose $\mathcal{O}$ is the graph that was used when defining the trap placed on partition $\mathcal{P}_b$ (so $|V(\mathcal{O})| = m_b$ and we can assume that $\mathcal{O}$ satisfies the second condition of Definition 4.10). Let Q be the adjacency matrix of $\mathcal{O}$. Let $b' < b$, set $m = m_{b'}$ and let $X_1, \ldots, X_m$ be the partition of $V(\mathcal{O})$ induced[16] by $\mathcal{P}_{b'}$. Suppose each of the sets X_i has size h and let X_i, X_j be two of these sets. Suppose δ and $y, x \in [0,1]^{m_b}$ satisfy the following conditions:*

1. $1/\log(m_b) < \delta < 1/200$.
2. *The vector y has only non-zero entries in X_i and x has only non-zero entries in X_j.*
3. *For each $1 \le p' \le m_b$ we have $y_{p'}/(\sum_p y_p) < \delta^6$.*
4. $\sum_{p=1}^{m_b} x_p > 2\delta h$.

Then, setting $g_1 = \sum_p y_p$ and $g_2 = \sum_p x_p$, we have

$$
\left| y^T Q x - \frac{1}{2} g_1 g_2 \right| \le 2\delta^2 g_1 g_2 \,. \tag{29}
$$

[16] This was defined explicitly just before Definition 4.10. Since we are identifying the clusters of $\mathcal{P}_b$ with the vertices of $\mathcal{O}$ we can also identify these clusters with the indices of the adjacency matrix Q. Hence, since we think of X_i as a subset of vertices of $\mathcal{O}$, we can say (as we will in item 2) that an index of a vector $x \in [0,1]^{m_b}$ belongs to X_i.

Proof. Put $k = \lfloor 1/\delta \rfloor$. Then item (1) of the lemma guarantees that $200 \leq k \leq \log(m_b)$. Item (3) of the lemma guarantees that the vector y/g_1 satisfies the condition of Lemma 4.17 with respect to k^6. Hence we can write $y/g_1 = \sum_R a_R v_R$ using the vectors of $\mathcal{V}_{k^6}$. Moreover, since item (2) guarantees that y has only non-zero entries in X_i we know that in the convex combination $\sum_R a_R v_R$ we have only $R \subseteq X_i$. Observe now that item (2) guarantees that x has only non-zero entries in X_j. Item (4) of the lemma guarantees that the vector x/g_2 satisfies the condition of Lemma 4.17 with respect to $\lceil h/k \rceil$. Hence we can write $x/g_2 = \sum_{R'} b_{R'} v_{R'}$ using the vectors of $\mathcal{V}_{\lceil h/k \rceil}$. Again, we know that in this convex combination we are only using sets $R' \subseteq X_j$.

Now, $(v_R)^T Q v_{R'} = e(R, R')/|R||R'|$. Hence, if $|R| = k^6$ and $|R'| = \lceil h/k \rceil$ and $R \subseteq X_i$, $R' \subseteq X_j$, then we can use the assumption that $\mathcal{O}$ satisfies the second condition of being a trap, to conclude that

$$\left| (v_R)^T Q v_{R'} - \frac{1}{2} \right| \leq 1/k^2 \leq 2\delta^2 \ . \tag{30}$$

We can thus infer from (30) and the facts that $\sum_R a_R v_R$ and $\sum_{R'} b_{R'} v_{R'}$ are convex combinations that

$$\begin{aligned} (y/g_1)^T Q (x/g_2) &= \left(\sum_R a_R v_R \right)^T Q \left(\sum_{R'} b_{R'} v_{R'} \right) \\ &= \sum_{S,T} a_R b_{R'} (v_R)^T Q v_{R'} \\ &\leq (1/2 + 2\delta^2) \sum_{R,R'} a_R b_{R'} \\ &= (1/2 + 2\delta^2) \end{aligned}$$

implying that $y^T Q x \leq (1/2 + 2\delta^2) g_1 g_2$. An identical argument gives $y^T Q x \geq (1/2 - 2\delta^2) g_1 g_2$, which completes the proof. □

4.5 Proof of Lemma 4.14

Suppose $\mathcal{A} = \{V_i : 1 \leq i \leq k\}$ and $\mathcal{B} = \{U_{i,i'} : 1 \leq i \leq k, 1 \leq i' \leq \ell\}$ (so $|\mathcal{B}| = k\ell$). We will say that a pair of sets (V_i, V_j) is *bad* if there are two sets $C_1, C_2 \subseteq [\ell] \times [\ell]$, each of size at least $\varepsilon\ell^2$ such that $|d(U_{i,i_1}, U_{j,j_1}) - d(U_{i,i_2}, U_{j,j_2})| \geq 2\varepsilon$ for every $(i_1, j_1) \in C_1$ and $(i_2, j_2) \in C_2$. Note that if (V_i, V_j) is bad then it cannot be good in the sense Definition 4.2. Hence, to show that $\mathcal{A}$ and $\mathcal{B}$ fail to satisfy the second condition of Definition 4.2 it is enough to show that there are at least $\varepsilon\binom{k}{2}$ bad pairs (V_i, V_j). As we mentioned after the statement of Theorem 4.4, we will actually show that there at least $(1 - 2\varepsilon^{1/10})\binom{k}{2}$ bad pairs.

A set $U_{i,i'}$ is called *useful* if there is an $X \in \mathcal{P}_b$ such that $U_{i,i'} \subset_{\varepsilon^{1/5}} X$. If $U_{i,i'}$ is not useful, we call it *useless*. A set V_i is called *useful* if it contains[17] less than $\varepsilon^{1/10}\ell$ useless sets $U_{i,i'}$. If V_i is not useful, we call it *useless*. Observe that there can be at most $\varepsilon^{1/10}k$ useless sets V_i, as otherwise $\mathcal{B}$ would not be an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_b$, which would contradict the third assumption of the lemma. Hence, there are at least $(1 - 2\varepsilon^{1/10})\binom{k}{2}$ pairs of useful sets (V_i, V_j). By the previous paragraph it is enough to show that every such pair is bad.

So for the rest of the proof, let us fix a pair of useful sets (V_i, V_j). Let us assume that ε is small enough so that $\varepsilon^{1/5} < 1/2$. Given a useful set $U_{i,i'} \subset_{\varepsilon^{1/5}} X \in \mathcal{P}_b$, we let $X_{\mathcal{P}_b}(U_{i,i'})$ denote this (unique) cluster in $\mathcal{P}_b$ that $\varepsilon^{1/5}$-contains $U_{i,i'}$. We will later prove the following claim:

Claim 4.21. *If V_i and V_j are both useful, then there are $D_1, D_2 \subseteq [\ell] \times [\ell]$ satisfying the following:*

- *D_1 and D_2 have size at least $\frac{1}{32}\ell^2$.*
- *For every $(i_1, j_1) \in D_1$ both U_{i,i_1} and U_{j,j_1} are useful and the pair $(X_{\mathcal{P}_b}(U_{i,i_1}), X_{\mathcal{P}_b}(U_{j,j_1}))$ belongs to the trap placed on $\mathcal{P}_b$.*
- *For every $(i_2, j_2) \in D_2$ both U_{i,i_2} and U_{j,j_2} are useful and the pair $(X_{\mathcal{P}_b}(U_{i,i_2}), X_{\mathcal{P}_b}(U_{j,j_2}))$ does not belong to the trap placed on $\mathcal{P}_b$.*

[17] Recall that each V_i is the union of ℓ sets $U_{i,i'}$.

In the next subsection we prove the lemma assuming Claim 4.21, in the subsection following it we will prove this claim.

4.5.1 Proof of Lemma 4.14 via Claim 4.21

Let α be the weight added to H by the trap that was placed on $\mathcal{P}_b$. Let D_1, D_2 be the subsets of $[\ell] \times [\ell]$ guaranteed by Claim 4.21. Take any pair $(i_1, j_1) \in D_1$ and let $X_1 = X_{\mathcal{P}_b}(U_{i,i_1})$ and $X_2 = X_{\mathcal{P}_b}(U_{j,j_1})$. Since $(i_1, j_1) \in D_1$ we know that the pair (X_1, X_2) was assigned an extra weight of α by the trap placed on $\mathcal{P}_b$. Now consider the traps with weight larger than α, that is, the traps that were placed on partitions $\mathcal{P}'$ that are refined by $\mathcal{P}_b$. Note that (X_1, X_2) might get an extra weight from a subset of these traps[18]. But since H contains only $\frac{1}{48}\sqrt{\log(1/\varepsilon)}$ many traps, the number of ways to choose the subset of the traps with weight larger than α from which (X_1, X_2) get an extra weight is bounded by $2^{\frac{1}{48}\sqrt{\log(1/\varepsilon)}} \ll \frac{1}{32\varepsilon}$. Hence D_1 must have a subset of pairs of size at least $\varepsilon\ell^2$, denoted D_1', and set of weights W_1 (all larger than α) with the following property; if $\alpha' > \alpha$ and $\mathcal{P}'$ is the partition on which the trap with weight α' was placed then for any $(i_1, j_1) \in D_1'$ the pair $(X_{\mathcal{P}'}(U_{i,i_1}), X_{\mathcal{P}'}(U_{j,j_1}))$ belongs to the trap on $\mathcal{P}'$ if and only if $\alpha' \in W_1$. We can also define D_2' and W_2 in the same manner.

We now claim that we can take C_1 and C_2 (the sets showing that (V_i, V_j) is bad) to be the sets D_1' and D_2'. First, as noted above, both D_1' and D_2' have size at least $\varepsilon\ell^2$. So to finish the proof we will have to show that for every $(i_1, j_1) \in D_1'$ and $(i_2, j_2) \in D_2'$ we have

$$|d(U_{i,i_1}, U_{j,j_1}) - d(U_{i,i_2}, U_{j,j_2})| \geq 2\varepsilon \; . \tag{31}$$

Let α' be the largest weight that belongs to exactly one of the sets W_1 and W_2. Assume without loss of generality that $\alpha' \in W_1$ and $\alpha' \notin W_2$. If there is no such

[18] More precisely, if X_1 and X_2 are subsets of the same cluster $X' \in \mathcal{P}'$, then they will never get an extra weight from the trap placed on $\mathcal{P}'$. If they belong to different clusters $X_1', X_2' \in \mathcal{P}'$, then they will receive an extra weight only if (X_1', X_2') belong to the trap placed on $\mathcal{P}'$.

weight (that is, $W_1 = W_2$) then set $\alpha' = \alpha$. We now recall Fact 4.12 which tells us that

$$\alpha' \geq 4^{-\frac{1}{48}\sqrt{\log(1/\varepsilon)}} \; . \tag{32}$$

Let $\mathcal{P}'$ be the partition on which the trap with weight α' was placed. Since traps with weight at least α are placed on partitions that are refined by $\mathcal{P}_b$, we see that if a set $U_{i,i'}$ is useful with respect to $\mathcal{P}_b$ it must also be useful with respect to $\mathcal{P}'$. This means that for each pair $(i_1, j_1) \in D'_1$ the trap at $\mathcal{P}'$ increases $d(U_{i,i_1}, U_{j,j_1})$ by at least

$$\alpha' \left(1 - \varepsilon^{1/5}\right)^2 \geq \alpha'(1 - 2\varepsilon^{1/5}) \geq 0.99\alpha' \; .$$

Similarly, for each pair $(i_2, j_2) \in D'_2$ the trap at $\mathcal{P}'$ increases $d(U_{i,i_2}, U_{j,j_2})$ by at most

$$2\alpha'\varepsilon^{1/5} \leq 0.01\alpha' \; .$$

Hence, disregarding for a moment all the other weights that can be assigned to these sets in H, we see that all the pairs in $(i_1, j_1) \in D'_1$ are such that $d(U_{i,i_1}, U_{j,j_1}) \geq 0.99\alpha'$ while all $(i_2, j_2) \in D'_2$ are such that $d(U_{i,i_2}, U_{j,j_2}) \leq 0.01\alpha'$. We will now show that this discrepancy is (essentially) maintained even when considering the entire graph H.

First, recall that by Fact 4.9 the total weight assigned to any pair of vertices of H in the graph G is bounded by $1/4^{\sqrt{\log(1/\varepsilon)}}$. Hence, recalling (32), we see that even after taking into account these weights, we have $d(U_{i,i_2}, U_{j,j_2}) \leq 0.02\alpha'$ for any $(i_2, j_2) \in D'_2$. Let us now consider the contribution of the weights coming from traps that were assigned a weight smaller than α'. Since these weights are $\alpha'/4, \alpha'/16, ...$ their sum is bounded by $\alpha'/3$, so after taking these weights into account we still have $d(U_{i,i_2}, U_{j,j_2}) \leq 0.36\alpha'$ for any $(i_2, j_2) \in D'_2$. Let us now consider the contribution coming from traps with weight more than α'. Consider any trap with weight $\alpha'' > \alpha'$ that was placed on a partition $\mathcal{P}''$. Recall that by definition of W_1, W_2 and by our choice of α', either the extra weight α'' was added to all pairs $(X_{\mathcal{P}''}(U_{i,i'}), X_{\mathcal{P}''}(U_{j,j'}))$

with $(i',j') \in D'_1 \cup D'_2$ or to none of them. Since all the sets U_{i,i_1} and U_{j,j_1} are useful we see that for each pair $(i_1,j_1) \in D'_1$ the pair (U_{i,i_1}, U_{j,j_1}) gets from the trap at $\mathcal{P}''$ a total weight at least

$$\alpha'' \left(1-\varepsilon^{1/5}\right)^2 \geq \alpha''(1-2\varepsilon^{1/5}) \ .$$

Set w to be the sum of the weights in W_1 that are larger than α'. Then the above discussion implies that for each $(i_1,j_1) \in D'_1$ we have

$$d(U_{i,i_1}, U_{j,j_1}) \geq (1-2\varepsilon^{1/5})w + 0.99\alpha' \geq w + 0.99\alpha' - 2\varepsilon^{1/5} \ . \tag{33}$$

Consider now a pair $(i_2,j_2) \in D'_2$; If a weight $\alpha'' \geq \alpha'$ belongs to W_2 then it can contribute to $d(U_{i,i_2}, U_{j,j_2})$ a weight of at most α'', hence such weights contribute to $d(U_{i,i_2}, U_{j,j_2})$ a total weight of at most[19] w. As to weights $\alpha'' > \alpha'$ that do not belong to W_2, we see that since U_{i,i_2} and U_{j,j_2} are useful, they can increase $d(U_{i,i_2}, U_{j,j_2})$ by at most $2\alpha''\varepsilon^{1/5}$. As the total sum of weights of all traps is at most 1, this extra contribution is bounded by $2\varepsilon^{1/5}$. All together, we see that for every $(i_2,j_2) \in D'_2$

$$d(U_{i,i_2}, U_{j,j_2}) \leq w + 0.36\alpha' + 2\varepsilon^{1/5}. \tag{34}$$

Recalling (32), we see that $4\varepsilon^{1/5} < 0.1\alpha'$. Hence, (33) and (34) imply that

$$d(U_{i,i_1}, U_{j,j_1}) - d(U_{i,i_2}, U_{j,j_2}) > 0.2\alpha' >_{(32)} 2\varepsilon$$

for every choice of $(i_1,j_1) \in D'_1$ and $(i_2,j_2) \in D'_2$. This establishes (31), thus completing the proof.

4.5.2 Proof of Claim 4.21

Let us start with observing that since V_i is assumed to be useful, it contains (more than) $\frac{1}{2}\ell$ useful sets $U_{i,i'}$. Let V'_i be the union of $\frac{1}{2}\ell$ such sets, and define V'_j is a similar way. From now on we will focus on V'_i and V'_j and their subsets $U_{i,i'}$ and $U_{j,j'}$

[19] Recall that by choice of α' the sets W_1 and W_2 contain the same weights larger than α'.

so we will only be talking about sets $U_{i,i'}$ and $U_{j,j'}$ that are useful. Recall that for any useful set $U_{i,i'}$ there is a (unique) set $X_{\mathcal{P}_b}(U_{i,i'}) \in \mathcal{P}_b$ such that $U_{i,i'} \subset_{\varepsilon^{1/5}} X_{\mathcal{P}_b}(U_{i,i'})$.

Suppose $\mathcal{P}_b$ has m clusters and recall that we defined the trap on $\mathcal{P}_b$ using an m-vertex graph $\mathcal{O}$ satisfying the first condition of Definition 4.10. That is (u, v) is an edge of $\mathcal{O}$ if and only if (X_u, X_v) belongs to the trap on $\mathcal{P}_b$. Define a vector $x \in [0, 1]^m$ by setting $x_u = |V'_i \cap X_u|/|X_u|$. Define $y \in [0, 1]^m$ similarly by setting $y_u = |V'_j \cap X_u|/|X_u|$. Recall that each of the sets V_i contains a $1/k$-fraction of the vertices H (since $|\mathcal{A}| = k$) so $|V'_i|$ contains a $1/2k$-fraction of the vertices of H. Since $\mathcal{P}_b$ has order m (so there are m sets X_u) and we assume that $m \geq k^2$ (in the second item of Lemma 4.14) we infer that

$$\sum_u x_u = \sum_u y_u = \frac{m}{2k} \geq \sqrt{m}/2\ . \tag{35}$$

If we take Q to be the adjacency matrix of $\mathcal{O}$, then by (35) we can apply Lemma 4.19 (with $g = m/2k$) to infer that

$$\frac{1}{4}(m/2k)^2 \leq x^T Q y \leq \frac{3}{4}(m/2k)^2\ . \tag{36}$$

Given a set $U_{i,i'}$ we define a vector $x^{i'}$ by setting $x^{i'}_u = |U_{i,i'} \cap X_u|/|X_u|$. Similarly given a set $U_{j,j'}$ we define a vector $y^{j'}$ by setting $y^{j'}_u = |U_{j,j'} \cap X_u|/|X_u|$. Observe that since V'_i is the union of the sets $U_{i,i'}$ we have $x = \sum_{i'} x^{i'}$ where the sum ranges over all the $\ell/2$ indices i' for which $U_{i,i'} \subseteq V'_i$. Similarly $y = \sum_{j'} y^{j'}$ where the sum ranges over all the $\ell/2$ indices j' for which $U_{j,j'} \subseteq V'_j$. Hence, we get from (36) that

$$\frac{1}{4}(m/2k)^2 \leq \sum_{i',j'} (x^{i'})^T Q y^{j'} \leq \frac{3}{4}(m/2k)^2\ . \tag{37}$$

Consider now any pair i', j' in the above sum. Let $X_u = X_{\mathcal{P}_b}(U_{i,i'})$ and $X_v = X_{\mathcal{P}_b}(U_{j,j'})$. Recall that $U_{i,i'}$ contains a $1/k\ell$ fraction of $V(H)$ while the sets X_u contains a $1/m$ fraction of $V(H)$. This means that

$$\sum_u x^{i'}_u = m/k\ell\ ,$$

hence

$$0 \le (x^{i'})^T Q y^{j'} \le m^2/k^2\ell^2 \ . \tag{38}$$

More importantly, since $|U_{i,i'} \cap X_u| \ge \left(1 - \varepsilon^{1/5}\right)|U_{i,i'}|$ we have

$$x_u^{i'} = |U_{i,i'} \cap X_u|/|X_u| \ge \left(1 - \varepsilon^{1/5}\right) m/k\ell \ , \tag{39}$$

and since $|U_{j,j'} \cap X_v| \ge \left(1 - \varepsilon^{1/5}\right)|U_{j,j'}|$ we have

$$y_v^{j'} = |U_{j,j'} \cap X_v|/|X_v| \ge \left(1 - \varepsilon^{1/5}\right) m/k\ell \ . \tag{40}$$

Suppose now that (X_u, X_v) belong to the trap placed on $\mathcal{P}_b$, that is, that $Q_{u,v} = 1$. We then get from (38), (39) and (40) that

$$0.99m^2/k^2\ell^2 \le \left(1 - \varepsilon^{1/5}\right)^2 m^2/k^2\ell^2 \le (x^{i'})^T Q y^{j'} \le m^2/k^2\ell^2 \ . \tag{41}$$

Suppose now that (X_u, X_v) does not belong to the trap placed on $\mathcal{P}_b$, that is, that $Q_{u,v} = 0$. We then get from (38), (39) and (40) that

$$0 \le (x^{i'})^T Q y^{j'} \le 2\varepsilon^{1/5} m^2/k^2\ell^2 \le 0.01m^2/k^2\ell^2 \ . \tag{42}$$

We thus see from (42) that the total to contribution to (37) of pairs (i', j') for which (X_u, X_v) does not belong to the trap is bounded by $(\ell/2)^2 \cdot 0.01m^2/k^2\ell^2 = 0.01(m/2k)^2$. Combining (37), (41) and (42) it thus must be the case that there are at least

$$\frac{\frac{1}{4}(m/2k)^2 - 0.01(m/2k)^2}{m^2/k^2\ell^2} \ge \frac{1}{32}\ell^2 \ ,$$

pairs (i', j') for which (X_u, X_v) belongs to the trap placed on $\mathcal{P}_b$. Hence we can take D_1 to be the collection of these pairs. Finally, we see from (37), (41) and (42) that the number of pairs (i', j') for which (X_u, X_v) belongs to the trap on $\mathcal{P}_b$ cannot be larger than

$$\frac{\frac{3}{4}(m/2k)^2}{0.99m^2/k^2\ell^2} \le \frac{31}{32}\ell^2 \ ,$$

so we can take D_2 to be the collection of pairs (i', j') that do not belong to D_1. We thus complete the proof of Claim 4.21.

4.6 Proof of Lemma 4.13

We will prove Lemma 4.13 by first performing a series of reductions that will culminate in Lemma 4.26. We will then spend most of this section proving Lemma 4.26. Let us first derive Lemma 4.13 from the following lemma:

Lemma 4.22. *Suppose $\gamma \leq \varepsilon$ and $\mathcal{Z} = \{Z_1, \ldots, Z_k\}$ is a γ-regular partition of H. Assume*

- $r < \frac{\log(1/\gamma)}{10\sqrt{\log(1/\varepsilon)}}$
- $\gamma^{1/4} \leq \beta \leq 1/100$

Then, if $\mathcal{Z}$ is a β-refinement of $\mathcal{P}_{r-1}$ it is also an 8β-refinement of $\mathcal{P}_r$.

Proof that Lemma 4.22 implies Lemma 4.13. By the definition of (ε, f)-regularity, we get that if $|\mathcal{A}| = k$ then $\mathcal{B}$ must be $\frac{1}{k}$-regular. Since $k \geq 1/\varepsilon$ we have $1/k \leq \varepsilon$. Since $\mathcal{B}$ is a refinement of $\mathcal{P}_0$ (recall that $\mathcal{P}_0$ is just the entire vertex set of H), it is in particular a $(1/k)^{1/4}$-refinement of $\mathcal{P}_0$. Hence, starting with $\beta = (1/k)^{1/4}$ we can repeatedly apply Lemma 4.22 (with $\gamma = 1/k$) as long as

$$r \leq \frac{\sqrt{\log(k)}}{10} \leq \frac{\log k}{10\sqrt{\log(1/\varepsilon)}} \tag{43}$$

and

$$8^r/k^{1/4} \leq 1/100 \; . \tag{44}$$

Taking $r = 2\log\log(k)$, we thus make sure that both (43) and (44) hold[20] with a lot of room to spare. Hence, after these $r = 2\log\log k$ applications of Lemma 4.22 we get that $\mathcal{B}$ must be an $8^{2\log\log k}/k^{1/4}$-refinement of $\mathcal{P}_{2\log\log k}$. Since

$$8^{2\log\log k}/k^{1/4} \leq 1/k^{1/5} \leq \varepsilon^{1/5} \; ,$$

we get that $\mathcal{B}$ is indeed an $\varepsilon^{1/5}$-refinement of $\mathcal{P}_{2\log\log k}$. □

[20] Recall that $k \geq 1/\varepsilon$. Since Theorem 4.4 allows us to assume that ε is sufficiently small, we can assume that k is large enough so that $2\log\log k < \frac{\sqrt{\log(k)}}{10}$ and that $8^{2\log\log k}/k^{1/4} \leq 1/100$.

Let us now continue with the proof of Lemma 4.22. So throughout the rest of this section we assume all the facts that are stated in the lemma. Suppose $\mathcal{P}_{r-1} = \{X_i : 1 \le i \le m\}$ and $\mathcal{P}_r = \{X_{i,i'} : 1 \le i \le m, 1 \le i' \le M\}$. Recall the sets $A_{i,j}, B_{i,j}$ that were used in the construction of the graph G in Subsection 4.2.2. With respect to these, we make the following definition:

Definition 4.23. *A pair of sets (Z_t, Z_u) is said to be β-*helpful *if*

1. *There are $1 \le i, j \le m$ such that $Z_t \subset_\beta X_i$ and $Z_u \subset_\beta X_j$ (we are* not *requiring $i \neq j$).*
2. *We have $\min(|Z_t \cap A_{i,j}|, |Z_t \cap B_{i,j}|) \ge \beta^2 |Z_t|$.*

We will need the following lemma, restated from [42].

Lemma 4.24. ([42]) *Let M be an integer and let $(A_j, B_j)_{j=1}^m$ be a sequence of balanced partitions of $[M]$. Let $0 < \zeta \le 1/2$ and let $\eta, \xi > 0$ be such that*

$$(1-\eta)(1-4\xi) > 1 - \zeta + \zeta^2 \, . \tag{45}$$

Then for every sequence $\lambda = (\lambda_1, \ldots, \lambda_M)$ such that $\lambda_{i'} \ge 0$ for every i', $\|\lambda\|_1 = 1$ and $\|\lambda\|_\infty < 1-\zeta$, there are at least ηm values of j for which $\min(\sum_{i' \in A_j} \lambda_{i'}, \sum_{i' \in B_j} \lambda_{i'}) > \xi$.

Lemma 4.25. *Suppose $\mathcal{Z}$ is a β-refinement of $\mathcal{P}_{r-1}$. Then, if $Z_t \subset_\beta X_i$ for some i, but there is no i' for which $Z_t \subset_{8\beta} X_{i,i'}$, then there are at least $2\beta m$ sets X_j such that each of these sets X_j β-contains at least $\frac{k}{2m}$ sets Z_u such that (Z_t, Z_u) are β-helpful.*

Proof. Let $Z_t \subset_\beta X_i$ and suppose that there is no $1 \le i' \le M$ for which $Z_t \subset_{8\beta} X_{i,i'}$. Write $\lambda_{i'}$ for $|Z_t \cap X_{i,i'}|/|Z_t|$. Then $\lambda_{i'} \ge 0$ for all i', $\|\lambda\|_1 \ge 1 - \beta$ (since $Z_t \subset_\beta X_i$) and $\|\lambda\|_\infty \le 1 - 8\beta$ (since we assume that there is no i' for which $Z_t \subset_{8\beta} X_{i,i'}$). Set $\zeta = 7\beta/(1-\beta) < 1/2$ and note that we have

$$(1-6\beta)(1-8\beta^2) > 1 - 6\beta - 8\beta^2 > 1 - \zeta + \zeta^2 \, , \tag{46}$$

where in the second inequality we use the fact that $\beta < 1/100$. Define the vector $\lambda' = \lambda/\|\lambda\|_1$. Then $\|\lambda'\|_1 = 1$ and

$$\|\lambda'\|_\infty \leq (1-8\beta)/\|\lambda\|_1 \leq (1-8\beta)/(1-\beta) = 1-\zeta \ . \tag{47}$$

Since $(A'_{i,j}, B'_{i,j})_{j=1}^m$ are balanced partitions of $[M]$, we can apply Lemma 4.24 to the vector λ' (with $\eta = 6\beta$ and $\xi = 2\beta^2$), and conclude that there are at least $6\beta m$ values of j, for which $\min(\sum_{i' \in A'_{i,j}} \lambda'_{i'}, \sum_{i' \in B'_{i,j}} \lambda'_{i'}) > 2\beta^2$. Recalling that $\lambda' = \lambda/\|\lambda\|_1$ and that $\|\lambda\|_1 \geq 1-\beta$ this means that for each such j we have $\min(\sum_{i' \in A'_{i,j}} \lambda_{i'}, \sum_{i' \in B'_{i,j}} \lambda_{i'}) > 2\beta^2(1-\beta) > \beta^2$. Notice that by the construction of the sets $A_{i,j}, B_{i,j}$, (that is $A_{i,j} = \cup_{i' \in A'_{i,j}} X_{i,i'}$ and $B_{i,j} = \cup_{i' \in B'_{i,j}} X_{i,i'}$) and by the definition of λ, these j's satisfy

$$\min(|Z_t \cap A_{i,j}|, |Z_t \cap B_{i,j}|) \geq \beta^2 |Z_t| \ , \tag{48}$$

that is, they satisfy the second condition of being β-helpful. This means that if a set Z_u is β-contained in X_j then (Z_t, Z_u) is β-helpful. So to finish the proof, we need to show that out of the $6\beta m$ values of j that satisfy (48), at least $2\beta m$ are such that X_j β-contains at least $k/2m$ sets Z_u. Hence, it is enough to show that $\mathcal{P}_{r-1}$ has at most $4\beta m$ sets X that β-contain less than $k/2m$ sets $Z \in \mathcal{Z}$.

Call a vertex $v \in V(H)$ *bad* if it either belongs to a set $Z \in \mathcal{Z}$ that is not β-contained in any $X \in \mathcal{P}_{r-1}$ or if it belongs to $Z \setminus X$ where $Z \subset_\beta X$. Note that since we assume that $\mathcal{Z}$ is a β-refinement of $\mathcal{P}_{r-1}$ then the fraction of H's vertices that are bad is bounded by 2β. Suppose now that there are more than $4\beta m$ sets X that β-contain less than $k/2m$ sets Z. Recall that each set X contains a $1/m$-fraction of vertices of H, while each Z contains a $1/k$-fraction. Therefore, if X has less than $k/2m$ sets Z that are β-contained in it, then half of its vertices belong to sets Z that are either β-contained in another set X' or that are not β-contained in any set. Hence, if $\mathcal{P}_{r-1}$ has more than $4\beta m$ such sets X, then more than 2β-fraction of H's vertices would be bad which is impossible. □

The main part of the proof of Lemma 4.22 will be the proof of the following lemma

Lemma 4.26. *Suppose $Z \in \mathcal{Z}$ and $X_i, X_j \in \mathcal{P}_{r-1}$. Suppose $Z \subset_\beta X_i$ and there are $\frac{k}{2m}$ sets $Z_u \subset_\beta X_j$ such that (Z, Z_u) is β-helpful. Then at least $\frac{k}{4m}$ of the sets Z_u are such that (Z, Z_u) is not γ-regular.*

We first derive Lemma 4.22 from Lemmas 4.25 and 4.26.

Proof of Lemma 4.22. By Lemma 4.25 we know that if $Z_t \subset_\beta X_i$ for some i, but there is no i' for which $Z_t \subset_{8\beta} X_{i,i'}$, then there is $S_t \subseteq [m]$ of size at least $2\beta m$ such that for any $j \in S_t$, the set X_j β-contains at least $k/2m$ sets Z_u for which (Z_t, Z_u) is β-helpful. By Lemma 4.26, each of these sets X_j β-contains at least $k/4m$ sets Z_u such that (Z_t, Z_u) is not γ-regular. Hence, all together (that is, when considering all the sets X_j where $j \in S_t$) there are at least $\beta k/2$ sets Z_u such that (Z_t, Z_u) is not γ-regular. Hence, since $\beta^2 > \gamma$ and we assume that $\mathcal{Z}$ is γ-regular, there cannot be more than $2\beta k$ sets Z_t as above.

Since we assume that for at least $(1-\beta)k$ of the sets Z_t there is a set X_i such that $Z_t \subset_\beta X_i$, it follows that for at least $(1-3\beta)k > (1-8\beta)k$ of the sets Z_t there exists an X_i and i' such that $Z_t \subset_{8\beta} X_{i,i'}$, which means that $\mathcal{Z}$ is an 8β-refinement of $\mathcal{P}_r$. □

In the next subsections we complete the proof of Lemma 4.22 by proving Lemma 4.26.

4.6.1 Setting the stage for the proof of Lemma 4.26

We start by setting some notation and observing some relations between the parameters involved. We remind the reader again that we will be assuming the conditions of Lemma 4.22. Also, hereafter we focus only on the $k/2m$ sets $Z_u \subset_\beta X_j$ such that (Z, Z_u) are β-helpful, namely the sets in the statement of Lemma 4.26.

Let us set $A = Z \cap A_{i,j}$ and $B = Z \cap B_{i,j}$. Also for each of the sets $Z_u \subset_\beta X_j$, if $|Z_u \cap A_{j,i}| \geq |Z_u \cap B_{j,i}|$ we set $W_u = Z_u \cap A_{j,i}$, otherwise we set $W_u = Z_u \cap B_{j,i}$. Since we assume that all the pairs (Z, Z_u) are β-helpful and that $\beta \geq \gamma^{1/4}$ we can deduce that

$$\min(|A|, |B|) \geq \beta^2 |Z| \geq \gamma^{1/2} |Z| \;, \tag{49}$$

and for all u we have

$$|W_u| \geq (1 - \beta)|Z_u|/2 \geq |Z_u|/4 \;. \tag{50}$$

Let $\mathcal{P}_{r_1}, ..., \mathcal{P}_{r_f}$ be the canonical partitions that refine $\mathcal{P}_{r-1}$ and on which we have placed a trap. For each $1 \leq \ell \leq f$, let α_ℓ be the weight[21] that was added to H when placing a trap on partition $\mathcal{P}_{r_\ell}$. Recall that H contains $\frac{1}{48}\sqrt{\log(1/\varepsilon)}$ many traps so

$$f \leq \frac{1}{48}\sqrt{\log(1/\varepsilon)} \;. \tag{51}$$

Also recall that by Fact 4.12 we have that all weights $\alpha_1, \ldots, \alpha_f$ satisfy

$$\alpha_1, \ldots, \alpha_f \geq 4^{-\frac{1}{48}\sqrt{\log(1/\varepsilon)}} \;. \tag{52}$$

Set

$$\delta = \frac{4^{-r}}{4^{\sqrt{\log(1/\varepsilon)}}} \;, \tag{53}$$

and recall that δ is the extra weight we have added to some of the pairs (x, y) in G when considering partition $\mathcal{P}_{r-1}$. Since in Theorem 4.4 we can assume that ε is sufficiently small, we get from (51), (52) and (53) that

$$\delta \ll \frac{1}{f}, \alpha_1, \ldots, \alpha_f \;. \tag{54}$$

We also observe that since $\gamma \leq \varepsilon$, and Lemma 4.22 assumes that $r \leq \frac{\log(1/\gamma)}{10\sqrt{\log(1/\varepsilon)}}$ we get from (53) that

$$\gamma^{1/3} \ll \delta \;. \tag{55}$$

[21] So recalling the way we have defined H in Subsection 4.2.3, we get that if $r_\ell = b = w(g)$ then $\alpha_\ell = 4^{-g}$.

We now define a set $A' \subseteq A$ using the following iterative process. We first set $A_0 = A$. If each of the clusters $X \in \mathcal{P}_{r_1}$ is such that $|A_0 \cap X| < \delta^6 |A_0|$, then the process ends with $A' = A_0$. If there is a cluster $X \in \mathcal{P}_{r_1}$ such that $|A_0 \cap X| \geq \delta^6 |A_0|$ then we set $A_1 = |A_0 \cap X|$, and continue to the next phase. If each of the clusters $X \in \mathcal{P}_{r_2}$ is such that $|A_1 \cap X| < \delta^6 |A_1|$, then the process ends with $A' = A_1$. If there is a cluster $X \in \mathcal{P}_{r_2}$ such that $|A_1 \cap X| \leq \delta^6 |A_1|$ then we set $A_2 = |A_2 \cap X|$ and move to the next phase. So the process either stops at some level $\mathcal{P}_{r_t}$ in which none of the clusters of $\mathcal{P}_{r_t}$ contains more than a δ^6-fraction of A_{t-1}, or it goes all the way to $\mathcal{P}_{r_f}$.

Let us make two important observations about A'. First, if the process stops at level $\mathcal{P}_{r_t}$ (where $t \leq f$) then for any $t' > t$ we have $|A' \cap X| < \delta^6 |A'|$ for all $X \in \mathcal{P}_{r_{t'}}$. This follows from the fact that $\mathcal{P}_{r_{t'}}$ refines $\mathcal{P}_{r_t}$. Therefore, A' has the property, that for each partition $\mathcal{P}_{r_t}$ the set A' is either contained in a single cluster $X \in \mathcal{P}_{r_t}$ or none of the clusters contains more than a δ^6-fraction of A'.

The second observation is that at each iteration the process picks a subset A_i satisfying $|A_i| \geq \delta^6 |A_{i-1}|$. Since we have at most f iterations, we get that the final set A' we end up with satisfies

$$|A'| \geq \delta^{6f}|A| = \left(\frac{4^{-r}}{4\sqrt{\log(1/\varepsilon)}}\right)^{6f} |A| \geq_{(51)} \left(\frac{4^{-r}}{4\sqrt{\log(1/\varepsilon)}}\right)^{\frac{6}{48}\sqrt{\log(1/\varepsilon)}} |A| \geq \varepsilon^{1/4}\gamma^{1/4}|A| \geq \gamma |Z| \,, \tag{56}$$

where the third inequality relies on the assumption of Lemma 4.22 that $r \leq \frac{\log(1/\gamma)}{10\sqrt{\log(1/\varepsilon)}}$ and the last uses (49) and the fact that $\gamma \leq \varepsilon$. We now use the same process to pick a set $B' \subseteq B$ satisfying the same properties discussed above, and whose size also satisfies

$$|B'| \geq \gamma |Z| \,. \tag{57}$$

Take one of the sets $W = W_u$ and assume without loss of generality that $W \subseteq A_{j,i}$. Recall that by $d_G(A', W)$ and $d_G(B', W)$ we denote the densities between these sets in the graph G, that is, before adding the traps to obtain the final graph H. First note that since A', B' both belong to $X_i \in \mathcal{P}_{r-1}$ and $W \subseteq X_j$, we infer that exactly the same weight was added in G to $d(A', W)$ and $d(B', W)$ by the partitions $\mathcal{P}$ that are refined by $\mathcal{P}_{r-1}$. Now recall that we put weight δ between all the edges connecting a vertex in $A_{i,j}$ and a vertex in $A_{j,i}$ and that we did not do so for edges connecting a vertex in $B_{i,j}$ and a vertex in $A_{j,i}$. Since $A' \subseteq A_{i,j}$, $B' \subseteq B_{i,j}$ and $W \subseteq A_{j,i}$ this means that $\mathcal{P}_{r-1}$ creates a discrepancy of δ between $d(A', W)$ and $d(B', W)$. Now recall that the weights assigned by G to the partitions $\mathcal{P}$ that refine $\mathcal{P}_{r-1}$ are $\delta/4, \delta/4^2, \delta/4^3, \ldots$. Since the sum of these weights is at most $\delta/3$ we get that

$$|d_G(A', W) - d_G(B', W)| \geq \frac{2}{3}\delta \geq_{(55)} \gamma \ . \tag{58}$$

It thus follows from (50) (56), (57) and (58) that if we had not added the traps to G, we would have thus concluded that *every* β-helpful pair (Z, Z_u) is not γ-regular. So to finish the proof we need to show that a large number of these β-helpful pairs are not γ-regular in H as well.

For $1 \leq \ell \leq f$ we let $d_\ell(A, B)$ be the weight added to $d(A, B)$ by the trap placed on $\mathcal{P}_{r_\ell}$. We thus have the following claim:

Claim 4.27. *If (Z, Z_u) is γ-regular, then there is $1 \leq \ell \leq f$ for which*

$$|d_\ell(A', W_u) - d_\ell(B', W_u)| > 4\delta^2 \ . \tag{59}$$

Proof. Recall that since both $A', B' \subseteq X_i \in \mathcal{P}_{r-1}$ and $W_u \subseteq X_j \in \mathcal{P}_{r-1}$ we get that $d_H(A', W_u)$ and $d_H(B', W_u)$ get the same weight from each of the traps placed on partitions $\mathcal{P}_{r'}$ that are refined by $\mathcal{P}_{r-1}$ (that includes the case that a trap was placed on $\mathcal{P}_{r-1}$). This means that a discrepancy between $d_H(A', W_u)$ and $d_H(B', W_u)$ can come either from $d_G(A', W_u)$ and $d_G(B', W_u)$ or from traps placed on partitions

$\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$. Thus, if (59) does not hold for all $1 \leq \ell \leq f$ then we would have

$$\begin{aligned}
|d_H(A', W_u) - d_H(B', W_u)| &= \left| d_G(A', W_u) - d_G(B', W_u) + \sum_{\ell=1}^{f} (d_\ell(A', W_u) - d_\ell(B', W_u)) \right| \\
&\geq |d_G(A', W_u) - d_G(B', W_u)| - \sum_{\ell=1}^{f} |(d_\ell(A', W_u) - d_\ell(B', W_u))| \\
&\geq \frac{2}{3}\delta - 4f\delta^2 \geq_{(54)} \frac{1}{3}\delta \geq_{(55)} \gamma \ ,
\end{aligned}$$

where in the second inequality we use (58). Recalling (50), (56) and (57) we thus infer that (Z, Z_u) is not γ-regular which is a contradiction. □

Assume that for each u for which (Z, Z_u) is γ-regular, we set ℓ_u to be the *smallest* integer for which (59) holds. In the following subsection we prove Lemma 4.26 via Claim 4.28 (stated below) and in the subsection following it we prove this claim thus completing the proof of Lemma 4.26.

Claim 4.28. *If (Z, Z_u) is γ-regular, then either A' or B' satisfies the following two conditions (we write the condition with respect to A'):*

- *There is no $X \in \mathcal{P}_{r_{\ell_u}}$ such that $A' \subseteq X$.*
- $|d_{\ell_u}(A', W_u) - \frac{1}{2}\alpha_{\ell_u}| > 2\delta^2$.

4.6.2 Proof of Lemma 4.26 via Claim 4.28

Once again, let us recall that given $Z \subset_\beta X_i$ and X_j we are focusing only the $k/2m$ sets $Z_u \subset_\beta X_j$ such that (Z, Z_u) are β-helpful. We need to show that at least $k/4m$ of the sets Z_u are such that (Z, Z_u) is not γ-regular.

Suppose to the contrary that there are $k/4m$ sets Z_u for which (Z, Z_u) is γ-regular. Then by Claim 4.28, for such Z_u either A' or B' satisfies the two conditions of Claim 4.28. Suppose without loss of generality that in at least $k/8m$ of these cases the set is A'. Also, suppose without loss of generality that out of these $k/8m$ cases, in at least $k/16m$ we have $d_{\ell_u}(A', W_u) > \alpha_{\ell_u}/2 + 2\delta^2$. Finally, since there are only f traps, we get that there must be an integer $1 \leq \ell \leq f$ for which there at least $k/16mf$ sets W_u for which the above holds and $\ell_u = \ell$. So for each of these sets we have

$$d_\ell(A', W_u) > \frac{1}{2}\alpha_\ell + 2\delta^2 \ . \tag{60}$$

For what follows we set S to be the collection of $k/16mf$ values of u for which (60) holds and such that $\ell_u = \ell$.

We now make a simple observation that relates $d_\ell(A', W_u)$, the graph $\mathcal{O}_{r_\ell}$ that was used to define the trap which was placed on level $\mathcal{P}_{r_\ell}$ and the way in which A' and W are "spread" over the clusters of $\mathcal{P}_{r_\ell}$. Let m_{r_ℓ} denote the number of clusters of $\mathcal{P}_{r_\ell}$ (which is also the number of vertices of $\mathcal{O}_{r_\ell}$). Let us use Y_p to denote the clusters of $\mathcal{P}_{r_\ell}$. Suppose X_i and X_j each contain h clusters of $\mathcal{P}_{r_\ell}$.

Let $x^a \in [0,1]^{m_{r_\ell}}$ be the vector satisfying $x^a_p = |A' \cap Y_p|/|Y_p|$ for every $1 \leq p \leq m_{r_\ell}$. Similarly, let $x^u \in [0,1]^{m_{r_\ell}}$ be the vector satisfying $x^u_p = |W_u \cap Y_p|/|Y_p|$ for every $1 \leq p \leq m_{r_\ell}$. If we take Q to be the adjacency matrix of $\mathcal{O}_{r_\ell}$ then

$$d_\ell(A', W_u) = \frac{(x^a)^T(\alpha_\ell Q)x^u}{(\sum_p x^a_p)(\sum_p x^u_p)} \ . \tag{61}$$

Our plan now is to show that the information we have gathered thus far contradicts Lemma 4.20. Let us start setting the stage for applying this lemma. First, as partition $\mathcal{P}_b$ in Lemma 4.20 we will take partition $\mathcal{P}_{r_\ell}$. So we are using m_{r_ℓ} as m_b in Lemma 4.20.

Second, as partition $\mathcal{P}_{b'}$ in Lemma 4.20 we will take partition $\mathcal{P}_{r-1}$. Note that here and in Lemma 4.20 we use m to denote the number of clusters in partitions $\mathcal{P}_{r-1}$ and $\mathcal{P}_{b'}$ and that we use $X_1, \ldots, X_m$ to name the m clusters of both partitions. As δ in Lemma 4.20 we use the same δ used here, that is $\delta = 4^{-r}/4^{\sqrt{\log(1/\varepsilon)}}$ as defined in (53). We clearly have $\delta < 1/200$. Also, to satisfy the first condition of Lemma 4.20 we need to make sure that $\delta > 1/\log(m_{r_\ell})$, or equivalently that

$$m_{r_\ell} =_{(22)} T^\phi(r_\ell) \geq_{(23)} T(\lfloor r_\ell/2 \rfloor) > 2^{4^{r+\sqrt{\log(1/\varepsilon)}}} =_{(53)} 2^{1/\delta} \ , \tag{62}$$

So we need to verify the second inequality. Recall that $r_\ell \geq r$ since we are only considering traps that were placed on partitions refining $\mathcal{P}_{r-1}$. Recalling (20) we also have $r_\ell \geq \log\log(1/\varepsilon)$ since the first trap was placed on a partition with this index. It is easy to see that these two facts imply that the second inequality in (62) indeed holds.

As the vector y in Lemma 4.20 we will take the vector x^a defined above, and as the vector x we take $\sum_{u \in S} x^u$ with S the set defined just after equation (60). Note that since $A' \subseteq X_i$ and for all u we have $W_u \subseteq X_j$, these vectors satisfy the second condition of Lemma 4.20.

Now, by Claim 4.28 there is no cluster[22] $X \in \mathcal{P}_{r_\ell}$ such that $A' \subseteq X$. By the process we have used to define A', this means that each of the clusters of $X \in \mathcal{P}_{r_\ell}$ contains no more than a δ^6-fraction of the vertices of A'. This means that the vector x^a defined above satisfies the third item of Lemma 4.20.

Finally, observe that each of the sets Y_p contains a $1/mh$-fraction of H's vertices[23] while each set Z_u takes a $1/k$-fraction. We thus get from (50) that the sum of entries of each of the vectors x^u is at least $mh/4k$. Since we assume that there are at least $k/16mf$ sets W_u, we infer that the sum of entries of x is at least $h/64f \geq_{(54)} 2\delta h$. Hence x satisfies the fourth condition of Lemma 4.20.

Since we assume that each of the sets W_u satisfies (60), we can use the formulation of (61) to infer that

$$(x^a)^T Q x^u > (1/2 + 2\delta^2) \left(\sum_p x_p^a \right) \left(\sum_p x_p^u \right) = \left(1/2 + 2\delta^2\right) g_1 g_2^u \ , \qquad (63)$$

where we set $g_1 = \sum_p x_p^a$ and $g_2^u = \sum_p x_p^u$. Now set $g_2 = \sum_p x_p = \sum_u g_2^u$. Summing

[22] Recall that we assume that $\ell_u = \ell$ for the set W_u with $u \in S$. See the discussion at the beginning of this subsection.

[23] Since each X_i contains a $1/m$ fraction of H's vertices and we assumed that X_i is partitioned into h sets Y_p.

over all vectors x^u, and applying (63) we have

$$(x^a)^T Qx = (x^a)^T Q \left(\sum_u x^u \right) > \left(1/2 + 2\delta^2\right) g_1 \sum_u g_2^u = \left(1/2 + 2\delta^2\right) g_1 g_2 \ ,$$

which contradicts (29) in Lemma 4.20.

4.6.3 Proof of Claim 4.28

We recall that we use α_ℓ to denote the weight added to H when placing a trap on partition $\mathcal{P}_{r_\ell}$, and that for a set W_u we defined ℓ_u just before Claim 4.28.

Claim 4.29. *Set $\alpha = \alpha_{\ell_u}$. If $|d_{\ell_u}(A', W_u) - d_{\ell_u}(B', W_u)| \geq 0.4\alpha$ then (Z, Z_u) is not γ-regular.*

Proof. Recall that ℓ_u was chosen to be the smallest integer for which (59) holds. Hence

$$\left| \sum_{\ell=1}^{\ell_u - 1} d_\ell(A', W_u) - d_\ell(B', W_u) \right| \leq 4f\delta^2 \leq_{(54)} \frac{1}{100}\alpha \ .$$

The assumption of the lemma thus gives

$$\left| \sum_{\ell=1}^{\ell_u} d_\ell(A', W_u) - d_\ell(B', W_u) \right| \geq 0.39\alpha \ .$$

Since the weights assigned to traps with weight smaller than α are given by $\alpha/4, \alpha/16, \ldots$, after taking into account all the traps placed on $\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$ we still have

$$\left| \sum_{\ell=1}^{f} d_\ell(A', W_u) - d_\ell(B', W_u) \right| \geq 0.05\alpha \ . \tag{64}$$

As we have noted in the proof of Claim 4.27, the only traps that can create a discrepancy between $d_H(A', W_u)$ and $d_H(B', W_u)$ are those placed on $\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$. Hence we can disregard the traps that were placed on partitions refined by $\mathcal{P}_{r-1}$, that is partitions other than $\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$. Thus, (64) holds even when considering *all* the traps placed in H. Finally, by Fact 4.9 the total weight assigned to edges in G is at most $1/4^{\sqrt{\log(1/\varepsilon)}} \leq_{(52)} 0.01\alpha$. We thus conclude that

$$|d_H(A', W_u) - d_H(B', W_u)| \geq 0.04\alpha >_{(52)} \varepsilon \geq \gamma \ .$$

Recalling (50), (56) and (57) we can deduce that (Z, Z_u) is not γ-regular. □

Claim 4.30. *If there is a cluster $X \in \mathcal{P}_{r_\ell}$ such that $A' \subseteq X$ and*

$$\delta^2 \leq d_\ell(A', W_u) \leq \alpha_\ell - \delta^2 \ , \tag{65}$$

then (Z, Z_u) is not γ-regular[24].

Proof. Let us define the vectors x^a and x^u as we have done just before equation (61). Let us also use the terminology used when defining these vectors. So $X = Y_q$ for some $Y_q \subseteq X_i$ implying that $x^a_q = |A'|/|Y_q|$ and all the other entries of x^a are 0. Suppose $Y_1, \ldots, Y_h$ are the clusters of $\mathcal{P}_{r_\ell}$ within X_j. Let $\mathcal{O}_{r_\ell}$ be the graph used when placing the trap on $\mathcal{P}_{r_\ell}$, let $v_q \in V(\mathcal{O})$ be the vertex corresponding to cluster Y_q and let $u_1, \ldots, u_h$ be the vertices corresponding to $Y_1, \ldots, Y_h$. Finally set $N = \{p : (v_q, u_p) \in E(\mathcal{O})\}$ to be the indices of the vertices $u_1, \ldots, u_h$ that are neighbors of v_q in $\mathcal{O}$. Then by (61) and (65) we have

$$\delta^2 \leq \frac{\alpha_\ell \sum_{p \in N} x^u_p}{\sum_{p=1}^h x^u_p} \leq \alpha_\ell - \delta^2 \ ,$$

implying that

$$\delta^2 \leq \frac{\sum_{p \in N} x^u_p}{\sum_{p=1}^h x^u_p} \leq 1 - \delta^2 \ .$$

This means that if we take $W^1 = W_u \cap (\bigcup_{p \in N} Y_p)$ then

$$\delta^2 |W_u| \leq |W^1| \leq (1 - \delta^2)|W_u| \ . \tag{66}$$

Let $W^2 = W_u \setminus W^1$ and note that it satisfies (66) as well. A critical observation now is that our choice of N implies that for all $p \in N$ the pair (Y_q, Y_p) belongs to the trap placed on $\mathcal{P}_{r_\ell}$ and for all $p \notin N$ the pair (Y_q, Y_p) does not belong to this trap. This means that $d_\ell(A', W^1) = \alpha_\ell$ while $d_\ell(A', W^2) = 0$.

We will now show that we can find $W' \subseteq W^1$ and $W'' \subseteq W^2$, satisfying $|W'| \geq \varepsilon^{1/10}|W^1|$, $|W''| \geq \varepsilon^{1/10}|W^2|$ and

$$|d_H(A', W') - d_H(A', W'')| \geq \gamma \ . \tag{67}$$

[24] Note that in this claim we are not assuming that $\ell = \ell_u$. That is, the claim is true for all $1 \leq \ell \leq f$. However, we will only apply it with $\ell = \ell_u$.

Recalling (56), this will imply that (Z, Z_u) is not γ-regular as the fact that $|W'| \geq \varepsilon^{1/10}|W^1|$ means that

$$|W'| \geq \varepsilon^{1/10}|W^1| \geq_{(66)} \varepsilon^{1/10}\delta^2|W_u| \geq_{(50)} \frac{1}{4}\varepsilon^{1/10}\delta^2|Z_u| \geq \frac{1}{4}\gamma^{1/10}\delta^2|Z_u| \geq_{(55)} \gamma|Z_u| \ ,$$

where in the fourth inequality we use the fact that $\gamma \leq \varepsilon$. A similar derivation would show that $|W''| \geq \gamma|Z_u|$.

So we are left with picking the sets W' and W''. Let us focus on W'. Consider some $1 \leq \ell' < \ell$. Since we assume that A' is contained is one of the clusters of $\mathcal{P}_{r_\ell}$ there must be a cluster $Y'_q \in \mathcal{P}_{r_{\ell'}}$ such that $A' \subseteq Y'_q$. Take some $p \in N$ and let $Y'_p \in \mathcal{P}_{r_{\ell'}}$ be the cluster containing Y_p. So we see that for each pair (Y_q, Y_p), either all the vertices $(x, y) \in Y_q \times Y_p$ get an extra weight of $\alpha_{\ell'}$ from that trap or none of them do (depending on whether (Y'_q, Y'_p) belongs to the trap placed on $\mathcal{P}_{r_{\ell'}}$). So for each pair (Y_q, Y_p) there is a subset $S_p \subseteq [\ell - 1]$ representing those traps from which (Y_q, Y_p) got an extra weight. Recall now that H contains only $\frac{1}{48}\sqrt{\log(1/\varepsilon)}$ many traps, so there are (much) less than $1/\varepsilon^{1/10}$ ways to pick a set $S_p \subseteq [\ell - 1]$. So there must be a subset $N' \subseteq N$ such that $S_p = S_{p'}$ for all $p, p' \in N'$ and such that $|W^1 \cap \bigcup_{p \in N'} Y_p| \geq \varepsilon^{1/10}|W^1|$. We now take $W' = W^1 \cap \bigcup_{p \in N'} Y_p$ and take S' to be the subset of $[\ell - 1]$ that is common to all $p \in N'$. Recapping the above, we see that if $\ell' \in S'$ then $d_{\ell'}(A', W') = \alpha_{\ell'}$ and if $\ell' \notin S'$ then $d_{\ell'}(A', W') = 0$. We can now define W'' and S'' in a similar way, such that if $\ell' \in S''$ then $d_{\ell'}(A', W'') = \alpha_{\ell'}$ and if $\ell' \notin S''$ then $d_{\ell'}(A', W'') = 0$.

If $S' = S''$ set $\alpha = \alpha_\ell$, otherwise, let ℓ' be the smallest index that appears in exactly one of the sets S' and S'' and set $\alpha = \alpha_{\ell'}$. Let us now compare $d_H(A', W')$ and $d_H(A', W'')$. By our choice of α, the traps with weight larger than α have the same contribution to both $d_H(A', W')$ and $d_H(A', W'')$. Using again the way we chose α we get that

$$\left| \sum_{\ell=1}^{\ell'} d_\ell(A', W') - d_\ell(A', W'') \right| = \alpha \ .$$

Now observe that the total weight added by traps with weight smaller than α is bounded by $\alpha/4 + \alpha/16... < \alpha/3$ so after taking into account all traps $\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$ there is still a discrepancy of at least

$$\left|\sum_{\ell=1}^{f} d_\ell(A', W') - d_\ell(A', W'')\right| \geq \alpha/2 \ .$$

As in previous proofs, we do not need to consider the weight coming from traps not placed on $\mathcal{P}_{r_1}, \ldots, \mathcal{P}_{r_f}$ (that is, traps placed on partitions refined by $\mathcal{P}_{r-1}$) since $A' \subseteq X_i \in \mathcal{P}_{r-1}$ and $W_u \subseteq X_j \in \mathcal{P}_{r-1}$. Finally, by Fact 4.9 the total weight assigned to edges in G is bounded by $1/4^{\sqrt{\log(1/\varepsilon)}} \leq_{(52)} \alpha/4$, so after taking into account all the weights assigned to (A', W') and (A', W'') in H we still have

$$|d_H(A', W') - d_H(A', W'')| \geq \alpha/4 \geq_{(52)} \varepsilon \geq \gamma \ .$$

This proves (67) thus completing the proof. □

Claim 4.31. *If there is a cluster $X \in \mathcal{P}_{r_{\ell_u}}$ such that $A' \subseteq X$ and a cluster $Y \in \mathcal{P}_{r_{\ell_u}}$ such that $B' \subseteq Y$ then (Z, Z_u) is not γ-regular.*

Proof. If either A' or B' satisfies (65) then Claim 4.30 implies that (Z, Z_u) is not γ-regular. So suppose both do not satisfy (65). Now note $d_{\ell_u}(A', W_u), d_{\ell_u}(B', W_u) \leq \alpha_{\ell_u}$ since α_ℓ is the maximum weight a pair of sets can get from the trap placed on $\mathcal{P}_{r_\ell}$. Recall that ℓ_u is an integer for which (59) holds hence one of the sets (say A') satisfies $0 \leq d_{\ell_u}(A', W_u) \leq \delta^2$ while the other satisfies $\alpha_{\ell_u} - \delta^2 \leq d_{\ell_u}(B', W_u) \leq \alpha_{\ell_u}$. But this means that

$$|d_{\ell_u}(A', W_u) - d_{\ell_u}(B', W_u)| \geq \alpha_{\ell_u} - 2\delta^2 \geq_{(54)} \alpha_{\ell_u}/2 \ ,$$

so (Z, Z_u) is not γ-regular by Claim 4.29. □

We are now ready to complete the proof of Claim 4.28. We know from Claim 4.31 that one of the sets A' or B' must satisfy the first requirement of the claim. Suppose it is A'. If A' also satisfies the second item then we are done, so suppose it does not.

If B' also satisfies the first requirement of the claim, then since ℓ_u is chosen to satisfy (59) and since we assume that A' does not satisfy the second requirement of the lemma, we get that B' must satisfy the second requirement and we are done.

So suppose now that the B' does not satisfy the first item. If $\delta^2 \le d_{\ell_u}(B', W_u) \le \alpha_{\ell_u} - \delta^2$ then by Claim 4.30 (Z, Z_u) is not γ-regular, which contradicts the assumption of Claim 4.28 that (Z, Z_u) is γ-regular. Finally, if either $d_{\ell_u}(B', W_u) \ge \alpha_{\ell_u} - \delta^2$ or $d_{\ell_u}(B', W_u) \le \delta^2$ we can combine this with the assumption that A' does not satisfy the second requirement of the claim to get that

$$|d_{\ell_u}(A', W_u) - d_{\ell_u}(B', W_u)| \ge \frac{1}{2}\alpha_{\ell_u} - 3\delta^2 >_{(54)} 0.4\alpha_{\ell_u} .$$

Claim 4.29 then implies that (Z, Z_u) is not γ-regular which again contradicts the assumption of Claim 4.28.

CHAPTER V

SIMULATION OF COUNTING TURING MACHINES

5.1 Introduction

The Turing machine is the most fundamental model of computation. Introduced by Alan Turing in 1936 [102], almost all of Theoretical Computer Science as we know it today has been built on top of the basic building block that is the Turing machine.

The simplest Turing machine just contains an infinitely long tape, a head that reads the tape and a state control that controls the movement of the Turing tape according to the symbols read. The tape serves as the carrier for the input, a storage device and (if necessary) as an output device. The basic model of the Turing machine is deterministic, in that, the output and the computation process of the Turing machine is determined only by the input given to the Turing machine. Even though there are more complicated variants of Turing machines, it could be shown that they are all equivalent in computation power.

Usually, in computer science, one describes algorithms in pseudocode or a simple programming language, such as C++. The relevance of Turing machines is captured by the Church-Turing thesis, which asserts that one can encode any "reasonable" algorithm into a Turing machine algorithm. That is, any algorithm that could be described by pseudocode or a conventional programming language can be encoded so that a Turing machine could be made to run this algorithm.

In this chapter, we observe randomized algorithms from a different perspective: we view them as algorithms performed by randomized Turing machines. In this setting, the derandomization of a randomized Turing machine amounts to performing a deterministic simulation of it. The basic ability required for simulating a randomized

Turing machine is the ability to count the number of accepting computations. We study the following problem in this chapter: how fast can a deterministic Turing machine count the number of accepting computations of a randomized/nondeterministic Turing machine? We exploit the fact that the Turing machine operations are very structured and hence a simulation algorithm should be able to exploit this structure.

A key feature of our algorithms is that they make no assumption about the kind of problem that the Turing machine is attempting to solve/compute. Our results only rely on the structure of the Turing machine and the manner in which a computation is performed.

5.1.1 Simulation of Turing Machines

How fast can we deterministically simulate a nondeterministic Turing machine (NTM)? This is one of the fundamental problems in theoretical computer science. Of course, the famous $\mathsf{P} \neq \mathsf{NP}$ conjecture, as most believe, would answer that we cannot hope to simulate nondeterministic Turing machines very fast. However, the best known result to date is the famous theorem of Paul, Pippenger, Szemerédi, and Trotter [78] that $\mathsf{NTIME}(O(n))$ is not contained in $\mathsf{DTIME}(o((n \log^* n)^{1/4}))$. This is a beautiful result, but it is a long way from the current belief that the deterministic simulation of a nondeterministic Turing machine *should* in general take exponential time.

We look at NTM simulations from the opposite end: rather than seeking better lower bounds, we ask how far can one improve the upper bound? We suspect even the following could be true:

$$\text{For any } \varepsilon > 0, \qquad \mathsf{NTIME}(t(n)) \subseteq \mathsf{DTIME}(2^{\varepsilon t(n)}).$$

To our knowledge, this does not contradict any of the current strongly held beliefs. This interesting question has been raised before, see e.g., [36].

For a given nondeterministic Turing machine (NTM), counting the number of accepting computation paths is a more difficult problem in general. If we can count the number of accepting computation paths, we can check if the count is nonzero or zero, thereby determining if the NTM accepts or not. So counting the number of accepting computation paths is at least as hard as simulating an NTM. Moreover, the complexity class #P captures the complexity of counting for decision problems in NP. The computational power of #P is highlighted by a celebrated result of Toda [99]. Toda showed that a polynomial time machine with a #P oracle can perform any computation in the polynomial hierarchy.

We prove that we can deterministically count the number of accepting paths of a k-tape NTM N in time

$$a^{kt/2} \cdot f(\cdot) \ ,$$

where a is the alphabet size, and t is the running time of N. The function f grows much slower than $a^{kt/2}$ and so does not contribute significantly to the running time. Our main theorem is:

Theorem 5.1. *The number of accepting computations of any k-tape NTM N with time complexity $t(n)$ can be computed by a DTM M in time*

$$a^{kt(n)/2} H_N^{\sqrt{t(n)}\log t(n)} \cdot q^2 \mathsf{poly}(\log q, k, t(n), a),$$

where a is the alphabet size and q is the number of states of N and H_N is a constant that depends only on a.

The ability to count the number of accepting computations immediately implies the ability to simulate probabilistic classes, like PP. In [103], van Melkebeek and Santhanam had shown a simulation of probabilistic time machines in deterministic time $o(2^t)$. However, their model restricted the nondeterministic choices available. Our model is more general and considers all the choices available, i.e., the choices in tape movement, written alphabet and next state.

Our bound has two key improvements. First, all nondeterminism arising from the choice of the next state or tape head movements is subsumed into the factor

$H_N^{\sqrt{t(n)}\log t(n)}$ with much smaller time dependence, compared to the main exponential term. Second, while N may write any of $S = a^{kt(n)}$ strings nondeterministically on its k tapes, our simulator needs to search only $\sqrt{S}$ of that space. Thus, we search the NTM graph in the *square-root* of its size.

There is no general deterministic procedure that can search a graph of size S in $\sqrt{S}$ time, even if the graph has a simple description. Hence to prove our theorem we must use the special structure of the graph: we must use that the graph arises from an NTM. We use several simple properties of the operation of Turing tapes and the behavior of guessing to reduce the search time by the square root.

5.1.2 Some related work

The only separation of nondeterministic from deterministic time known is $\mathsf{DTIME}(n) \neq \mathsf{NTIME}(n)$ proved in [78], which is also specific to the multi-tape Turing machine model. It is also known that nondeterministic two-tape machines are more powerful than deterministic one-tape machines [59], and non-deterministic multi-tape machines are more powerful than deterministic multi-tape machines with additional space bound [60]. Limited nondeterminism was analyzed in [36], which showed that achieving it for certain problems implies a general subexponential simulation of nondeterministic computation by deterministic computation. In [103] an unconditional simulation of time-$t(n)$ probabilistic multi-tape Turing machines Turing machines operating in deterministic time $o(2^t)$ is given.

For certain NP-complete problems, improvements over exhaustive search that involve the constant in the exponent were obtained in [13], [16], [89], and [96], while [53] and [74] also found NP-complete problems for which exhaustive search is not the quickest solution. Williams [105] showed that having such improvements in all cases would collapse other complexity classes. Drawing on [103], Williams [105] showed that the exponent in the simulation of NTM by DTM can be reduced by a multiplicative factor smaller than 1. The NTMs there are allowed only the string-writing form of nondeterminism, but may run for more steps; since the factor is not close to 1/2, the result in [105] is incomparable with ours.

5.2 Model & Problem Statement

Given a nondeterministic Turing machine (NTM) N, let $t = t(n)$ be the time complexity for inputs of size n. We assume that $t(n)$ is *time-constructible* and *space-constructible.* A function $f : \mathbb{N} \to \mathbb{N}$ is called *time-constructible* if there exists a Turing machine M that given a string 1^n consisting of n ones as input, outputs the binary representation of $f(n)$ in $O(f(n))$ time. Similarly, f is called *space-constructible* if there exists a Turing machine M that given the string 1^n, outputs the binary representation of $f(n)$, while using only $O(f(n))$ space. Throughout this chapter, we will use q for the number of states, k for the number of tapes, and a for the alphabet size of N. Our question is, in terms of a, k, q, what is the most efficient way in which a deterministic Turing machine (DTM) can count the number of accepting computations of N? Let us first see two straightforward approaches.

Tracing the computation tree: This is the standard method, the one that is the most straightforward. Here we trace down each computation path of the NTM N from the starting configuration till it halts. We keep count of the number of accepting paths. Since we do not limit N to be binary-branching, individual nodes of the tree may have degree as high as $v = a^k 3^k q$, where the "3" allows each head on each tape to move left, right, or stationary. This leads to the following proposition.

Proposition 5.2. *The number of accepting computations of any NTM N with time complexity $t(n)$ can be computed by a DTM M in time $c(N)^{t(n)}$, where $c(N)$ is a constant depending on N.*

An upper bound for $c(N)$ is given by the maximum degree of the computation tree, v, which depends on q as well as k and a. There is thus a factor q^t in the running time of M. It would be our goal to eliminate such a factor.

Traversing the configuration graph: Here we show that we can eliminate q^t factor by looking at the configuration graph of N.

A *configuration* of a Turing machine is an encoding of the current state, the tape contents, and current position of the tape heads. Configurations form a directed graph where there are directed edges from a configuration to a valid successor configuration, with sources being the initial configurations I_x on given inputs x and sinks being accepting configurations I_a (perhaps including non-accepting halting configurations too). When N uses at most space s on any one tape, the number of nodes in the graph (below I_x) is at most

$$qa^{ks}s^k.$$

Notice that $s \leq t$ holds trivially, where t is the running time of N. By using a modified configuration graph and a variant of the Breadth First Search algorithm, we get the following proposition.

Proposition 5.3. *The number of accepting computations of any NTM N with time complexity $t(n)$ can be computed by a DTM M in time $q^2(3at)^k a^{kt(n)}\mathsf{poly}(\log q, k, t(n), a)$.*

Proof. We consider the following modified configuration graph $\widetilde{C}$: the nodes are pairs (I, p), where I is a configuration of the NTM N and p is an integer $0 \leq p \leq t$. By the above bound, this graph has at most $S = qa^{kt}t^k \cdot (t+1)$ nodes. There is a directed edge from (I, p) to (I', p') if and only if I' is a valid successor configuration for I in the NTM N and $p' = p + 1$. Notice that $\widetilde{C}$ is a directed acyclic graph, and that for any two nodes $(I, p), (I', p') \in V(\widetilde{C})$ all paths from (I, p) to (I', p') are of the same length. This follows from the fact that all the paths have to be of length $p' - p$. One can use a variant of Breadth First Search in $\widetilde{C}$ to keep track of the number of shortest paths to each node from the starting node $(I_x, 0)$. By construction of $\widetilde{C}$, each path is a shortest path, and this gives the number of shortest paths from $(I_x, 0)$ to each node. We use a look up table for simulating the transition function of N.

At the end, we have the number of paths leading to each node. We go through all the nodes, and sum up the number of paths to all the nodes corresponding to accepting configurations of N.

The dominant term in the running time comes from the sorting we need to perform. This is

$$O(Sv \cdot \log(Sv) \cdot \log S) = q^2(3at)^k a^{kt} \mathsf{poly}(\log q, k, t, a).$$

□

Notice that the dependence on q is at most q^2, not q^t. Proposition 5.2 and Proposition 5.3 present the tradeoff between space and time. For tracing the tree, we need to store only the current path from the root and some local information, but we need to spend more time in re-computing nodes that are reached by multiple paths. In Proposition 5.3, we avoid this redundant expansion at the expense of storing the whole list of visited nodes.

Theorem 5.4. *The number of accepting computations of any NTM N can be computed by a DTM M in time $c(N)^{t(n)}$, where $t(n)$ is the time complexity of N and the constant $c(N)$ depends on the alphabet size a and the number of tapes k of N, but is* ***independent*** *of q.*

Proof. We define *weak trace* as the move labels on an accepting path in the computation tree, but *omitting* the next-state information. There are only $(a^k 3^k)^t$ such potential witnesses to enumerate. We call a path "compatible with the weak trace y" if it adds states $q_0, \ldots, q_t$ to the parts specified by y to make a legal computation. Below, we show that for each of these weak traces we can compute the number of compatible accepting computations in time $q^2 a^{2k} 3^k \mathsf{poly}(\log q, a, k, t)$.

For each step j in the computation, define Q_j to be the set of pairs (q, c), where q is a state N that can be in at step j on some full path that is compatible with y, and c is the number of compatible full paths that lead to q at step j. We also keep track of the number of compatible accepting computation paths.

Initially $Q_0 = \{ (q_0, 1), (q_A, 0) \}$, where q_0 is the start state of N and q_A is the accepting state. Given Q_{j-1}, to compute Q_j, we first get the entry $(q_A, c_A) \in Q_{j-1}$. and add (q_A, c_A) to Q_j. For each pair (r, c) in Q_{j-1}, we look up all possible next states r' in a pre-computed lookup table based on the transition relation of N. For each compatible transition of r to r' at the jth step, we add (r', c) to Q_j where c is the second entry from (r, c) in Q_{j-1}. If the element (r', c) already exists in Q_j, we simply add a duplicate copy. This is done to preserve the count.

After computing each Q_j, M needs to sort and combine the duplicate states in Q_j. All of the pairs $(r', c^i_{r'}) \in Q_j$ are replaced by one pair $(r', \sum_i c^i_{r'})$ where the second entry is the sum of all the second entries. This achieves a two-fold purpose. First, the second entry of r' is the number of compatible paths leading to r'. Second, this helps maintain the sets Q_j bounded in size. The simulation finally goes through Q_t, and at the end the second entry in $(q_A, c_A) \in Q_t$ gives the number of accepting computations compatible to the given weak trace.

The deterministic counting machine M has $k + 3$ tapes. The first k tapes are meant to simulate the tapes of the NTM N. The next tape contains the transition function of the NTM as a lookup table. The remaining two tapes left alone for the bookkeeping.

The lookup table rows are indexed by the current state, the k symbols being currently read, the k symbols that would be written and k directions (left, right or stay) in which the tape heads shall move. The entries contain: (i) all the possible states that N can move to, and (ii) the number of distinct compatible ways to reach each state corresponding to the indexed state, symbols read and written and directions moved. The lookup table is stored in a serial fashion in a single tape. There are $q(3a^2)^k$ rows and each row could have at most q entries. The cost of a look up is upper bounded by $q(3a^2)^k \cdot [k \log(3a^2) + \log q] + q \log q$.

After the lookups, we sort and combine duplicates from a set (of pairs) that could be potentially q^2 in size. This takes $q^2 \log q$ comparisons, where each comparison costs $\log q$, yielding a running time of $q^2 \log^2 q$. Multiplying the whole expression by t, we get that the running time per weak trace is

$$[q(3a^2)^k \cdot [k \log(3a^2) + \log q] + q \log q + q^2 \log^2 q] \cdot t,$$

which can be upper bounded by

$$h(a, q, k, t) = q^2 a^{2k} 3^k \mathsf{poly}(\log q, a, k, t).$$

We run through all the possible weak traces and obtain the number of compatible accepting computations for each one. We add the number over all weak traces to get the distinct number of accepting computation paths. Notice that each computation path is accounted for by exactly one weak trace, so there is no over-counting.

The overall running time is $(3^k a^k)^t$ multiplied by the function h. The factor h is majorized by $(1 + \delta)^t$ for any $\delta > 0$ as t becomes sufficiently large, which happens as inputs x to N become sufficiently large. The whole time is thus bounded by $(3^k a^k + \delta')^t$, where $\delta' = 3^k a^k \delta$. Note that δ' is independent of q and can likewise be made arbitrarily small when a and k are fixed. Hence the total computation time is asymptotically bounded by $c(N)^{t(n)}$ where $c(N)$ is independent of q. □

We improve on this idea in the next section by using block traces, which are more succinct witnesses, carrying more information. As a consequence, we would need to enumerate a smaller number of them, resulting in a faster simulation.

5.3 Block-Trace Simulation

We introduce the idea of block traces, where we break down computations of the NTM N into "blocks" of d steps, where d will be specified later. Let us start with the following definitions.

Definition 5.5. *A* segment of size d *for a k-tape NTM N with alphabet of size a is a sequence of 4-tuples*

$$\tau = [(r_1, f_1, \ell_1, u_1), \ldots, (r_k, f_k, \ell_k, u_k)],$$

where for each tape j, $1 \leq j \leq k$:

- $r_j \in \{0, \ldots, d\}$ *stands for the maximum number of cells to the right of its starting position the tape head will ever be over the next d steps,*
- $f_j \in \{0, \ldots, d - r_j\}$ *is the number of cells left of the position of r_j that the tape head ends up after the d-th step, and*
- $\ell_j \in \{1, \ldots, d\}$ *is the number of distinct cells that shall be changed over the next d steps on tape j. For a given r_j and f_j we have the bound $\ell_j \leq d + 1 - \min\{r_j, f_j\}$.*
- u_j *is a string of length ℓ_j, which is interpreted as the final contents of those cells.*

Definition 5.6. *A* block trace of block-size d*, for an NTM N, is a sequence of segments of size d.*

Definition 5.7. *An accepting full path is* compatible *with a block trace if the latter has $\lceil t/d \rceil$ blocks where t is the total number of steps in the path, and in every block each 4-tuple (r_j, f_j, ℓ_j, u_j) correctly describes the head locations after the corresponding d steps of the full path,* and *every character in u_j is the correct final content of its cell after the d steps.*

Every accepting computation path gives rise to a block trace with which it is compatible. The above definition includes all the possible head movements of N over the next d steps. The following immediate, but critical lemma validates the correctness of this method of counting.

Lemma 5.8. *Two different block trace witnesses give rise a disjoint set of computation paths.*

Proof. Every computation path has a corresponding block trace witness. So it is enough to show that this witness is unique. This follows from the definition of block trace witness. For a given fixed computation path, at each time instance and for each tape j, the values r_j, f_j, ℓ_j, u_j are fixed as in definition 5.5. So the lemma follows. □

The running time of the resulting simulation is a consequence of the following lemmas. Lemma 5.9 bounds the number of potential block trace witnesses for a given d. The proof of Lemma 5.10 follows by generalizing the ideas in Theorem 5.4. The main distinction is that we are dealing with a block trace witness, i.e., segments of size d each. The structure of the lookup table needs to be modified accordingly. Lemma 5.10 thus gives us an algorithm to count the number of accepting computations compatible with each block trace witness. Theorem 5.11 combines these two lemmas to obtain an algorithm to count the number of accepting computations of the NTM N.

Lemma 5.9. *The number B of valid segments is at most $(32a^d)^k$. Hence the number of potential block trace witnesses is at most $B^{t/d} = a^{kt}32^{kt/d}$.*

Proof. We first bound the number of 4-tuples per each tape. We note that for ℓ cells affected for a particular segment, there are a^ℓ possible strings u. We sum over all the possible values of ℓ – ranging from d to 1. Direct calculation gives us that for $\ell = d$, there are at most 6 possible sets of (r, f), for $\ell = d - 1$ at most 14, etc. An upper bound for the number of possible sets for $\ell = d + 1 - i$ is given by $6, 14, 24, \ldots$

for $i = 1, 2, 3, \ldots$. This can be simply written as $i^2 + 5i$. A total number of distinct 4-tuples is upper bounded by

$$\sum_{\ell=d}^{1}[(d+1-\ell)^2 + 5(d+1-\ell)]a^\ell = a^d \cdot \sum_{i=1}^{d}(i^2+5i)/a^{i-1} \leq 32a^d,$$

where the last inequality follows by the worst case value $a = 2$. Since we have k tapes, we obtain $B \leq (32a^d)^k$. (In fact, we can get $B \leq (C_a a^d)^k$ where $C_a \longrightarrow 6$ as $a \longrightarrow \infty$, but this tighter counting does not result in any notable improvement in the eventual simulation.) □

Lemma 5.10. *The number of accepting computations that are compatible with a given block trace witness can be calculated by a deterministic Turing machine in time* $q^2 a^{3kd}\mathsf{poly}(\log q, k, t, a, d)$.

Proof. We generalize the ideas in Theorem 5.4. We are given a block trace witness, i.e., t/d segments of size d each. The idea is to maintain the set Q_i of pairs (r, c). Here r is a state that the machine N on input x can possibly be in, after the i-th segment of d steps in some computation path, and c is the number of distinct compatible computation paths leading to r at the time instance $i \cdot d$. We precompute a lookup table T_d whose rows are indexed by the following information:

- An initial state p entering the segment of d steps.
- Strings w_j of length at most $2d - 1$ indicating the true contents in the cells surrounding the head on tape j. The cases where a segment of cells on the right or left are blank (through never having been visited before) are handled by adjoining integers b_j indicating such cells.
- The string u_j and integers r_j, f_j for each tape j, representing a segment in a block trace.

The lookup table entries contain pairs (r, c) where r is a state that can be reached from p in a manner compatible with the lookup table index and c the number of compatible paths to reach p from r. The lookup table is the d-length segment equivalent of the lookup table in Theorem 5.4. There are $qa^{(3d-1)k}d^2$ rows of the table, the length of each index in binary being thus asymptotic to $\log_2 q + (3d-1)k\log_2 a + 2\log_2 d$. The cost of each lookup is thus upper bounded by $qa^{3kd}d^2(\log q + 3kd\log a + 2\log d) + q\log q$. Similar to Theorem 5.4, we transfer the values in the pair corresponding to the accepting state q_A from Q_{i-1} to Q_i for each i. We stop the computation when we reach Q_t. The second entry in the pair corresponding to q_A in Q_t gives the number of distinct compatible computation paths leading to an acceptance.

By including the time for sorting the states, and multiplying by the running time of t/d segments, we get

$$[qa^{3kd}d^2(\log q + 3kd\log a + 2\log d) + q\log q + q^2\log^2 q]\cdot t/d,$$

which is upper bounded by

$$q^2a^{3kd}\mathsf{poly}(\log q, k, t, a, d).$$

□

Theorem 5.11. *The number of accepting computation paths of a nondeterministic k-tape TM with q states and alphabet size a can be computed by a multi-tape deterministic TM in time*

$$a^{kt}C_N^{\sqrt{t}}\cdot q^2\mathsf{poly}(\log q, k, t, a),$$

where C_N is a constant that depends only on a and k.

Proof. This follows from Lemmas 5.9 and 5.10. The deterministic machine tries out all the possible block witnesses, with a running time

$$q^2a^{kt+3kd}32^{kt/d}\mathsf{poly}(\log q, k, t, a, d).$$

The machine keeps track of number of compatible accepting paths for each witness and then adds them up to get the total number of accepting computations. Lemma

5.8 ensures that there is no overcounting, i.e., each computation path is captured by exactly one block trace witness. We can choose d to be such that these the product of the two factors a^{3kd} and $32^{kt/d}$ are minimized. Direct calculation gives us that this happens when $d = \sqrt{5t/(3\log_2 a)}$. Setting $C_N = 2^{k\sqrt{15\log_2 a}}$, we get a running time of

$$a^{kt}C_N^{\sqrt{t}} \cdot q^2\mathsf{poly}(\log q, k, t, a).$$

□

5.4 Main Theorem

Our goal in this section is to reduce the exponent of the computation time by half. The algorithm that searches the configuration graph, discussed in Section 5.2, needs a running time of $q^2(3at)^k a^{kt}\mathsf{poly}(\log q, k, t, a)$. The block trace method requires a running time of $a^{kt}C_N^{\sqrt{t}} \cdot q^2\mathsf{poly}(\log q, k, t, a)$. The time bounds seem similar, but the approaches are quite different, we shall combine these two approaches to get the reduction in running time.

In the graph search method, the dominating part in the running time is caused by the number of configurations. There are at most $qa^{kt}t^k$ of them. If the NTM used only a tape space of $kt/2$ over all the k tapes, then the number of configurations would be reduced to $qa^{kt/2}t^k$. This would lead to a computation that requires $q^2(3at)^k a^{kt/2}\mathsf{poly}(\log q, k, t, a)$ time.

But when the NTM computations use more than $kt/2$ tape space, we will use the block trace method to exploit an interesting property of the Turing machines. We make the following observation: the last time we visit a location in the NTM tape, we need not write any character there. This is because the tape head would not be reading from that position later. If the NTM visits at least $kt/2$ locations on all tapes together, then each of these $kt/2$ locations is visited once for a last time. For the block traces, we do not need to have a symbol to write down, if we are visiting a tape location for a last time. We could potentially save on a factor of $a^{kt/2}$ on the running time. This brings down the main factor in the running time in Theorem 5.11 to $a^{kt/2}$ as well.

We need the following definition and the subsequent Lemma 5.13 before getting to the main theorem. Lemma 5.14 ensures that we do not over-count the number of accepting computations and is immediate from the definition of directional traces.

Definition 5.12. *A* directional segment of size *d for a k-tape NTM N with alphabet size a is a segment of size d, omitting the strings u_j, that is*

$$\tau = [(r_1, f_1, \ell_1), \dots, (r_k, f_k, \ell_k)],$$

where r_j, f_j, ℓ_j are defined as in Definition 5.5.

A directional trace *of block size d, is a sequence of directional segments of size d.*

Lemma 5.13. *The number of segments of block size d is upper bounded by d^3. The number of potential directional trace witnesses is at most $(d^3)^{t/d}$.*

Proof. The calculations are similar to those in the proof of Lemma 5.9. The difference here is that we do not need to count the number of possible strings u for each tape. This bounds the number of directional segments to $\sum_{i=1}^{d}(i^2+5i) = \frac{1}{3}d(d+1)(d+8) \leq d^3$ per tape, for $d \geq 6$. Since we have k tapes, the bound is d^{3k}. The bound on directional traces follows. □

Lemma 5.14. *Two different directional trace witnesses give rise a disjoint set of computation paths. In other words, every computation path corresponds to a unique directional trace witness.*

We are now ready to prove the main theorem.

Theorem 5.1 (Restated)**.** *The number of accepting computations of any k-tape NTM N with time complexity $t(n)$ can be computed by a DTM M in time*

$$a^{kt(n)/2} H_N^{\sqrt{t(n)}\log t(n)} \cdot q^2 \mathsf{poly}(\log q, k, t(n), a),$$

where a is the alphabet size and q is the number of states of N and H_N is a constant that depends only on a.

Proof. We assume that we know an upper bound $t = t(n)$ as a function of the input length n. (If not, we run the computations for $t = 1, 2, 3, \cdots$, and this will introduce a multiplicative factor $t(t-1)/2$, which is $\mathsf{poly}(t)$ anyway.)

The counting simulation, performed by the DTM M, consists of three parts. First, preprocessing the directional traces. Second, running the block trace computation for those traces that have tape usage $\geq kt/2$. And third, running the graph search computation restricting the tape usage to $kt/2$.

1. This is the preprocessing stage. Here the DTM M lists down all the possible directional traces. There are $d^{3t/d}$ such traces by Lemma 5.13. For $d = \sqrt{5t/(3\log_2 a)}$, as optimized in Theorem 5.11, we get that the number of traces is $(\sqrt{t})^{O(\sqrt{t})}$ or $H_N^{\sqrt{t}\log t}$, where H_N depends only on a.

   The machine M calculates the total tape usage of N for each directional trace. In particular, M decides if the total tape usage is $\leq kt/2$ or $\geq kt/2$. Also, for each tape location M calculates the time of the last visit to that location. This data is stored in a lookup table, in another tape of M. All these operations can be performed in time $\mathsf{poly}(k, t)$ per directional trace.

2. The block trace simulation is performed for the directional traces where the total tape usage is $\geq kt/2$. For a given directional trace, all the block traces that match the (r, f, ℓ) parts are generated, but with a small difference. For those time instances for which the tape head is visiting a location for the last time, the block trace is generated with a ␣ character in the corresponding location. The preprocessed data from the directional traces is used to determine if the visit is a last one for the tape location.

   There are at least $kt/2$ locations visited for the last time, so the number of corresponding block traces is $\leq a^{kt/2}$. So the total number of relevant directional traces here is upper bounded by $H_N^{\sqrt{t}\log t} a^{kt/2}$.

The running time in the Lemma 5.10 holds essentially by the following observation. The lookup table could be expanded (slightly) to accommodate one more symbol in the alphabet, the '␣' symbol. The set of states that are possible in the lookup table after a doing block trace move with a '␣' is obtained by treating the '␣' symbol as a wildcard. To accommodate for the ␣ symbol, for every state r reachable, the lookup table provides the pair (r, c) where c is the number of distinct ways to reach r compatible with the given directional segment. The number c also changes (from the lookup table in Lemma 5.10) because of the wildcard nature of the ␣ symbol.

The running time of this stage is $a^{kt/2} H_N^{\sqrt{t}\log t} \cdot q^2 \mathsf{poly}(\log q, k, t, a)$.

3. The directional trace can be discarded for the cases when the total tape usage is $\leq kt/2$. For all such cases combined, just one call to the graph search simulation is sufficient. The deterministic machine M keeps track of the configurations, and rejects a branch as soon as the tape usage exceeds $kt/2$. This gives a running time of $a^{kt/2} q^2 (3at)^k \mathsf{poly}(\log q, k, t, a)$.

The theorem follows by observing that every accepting computation path of the NTM N, is captured by exactly one of the two simulations, the block trace or the graph search method. The total number of accepting computations is simply the sum of the numbers given by the two methods. The running time is

$$T(n) = a^{kt/2} H_N^{\sqrt{t}\log t} \cdot q^2 \mathsf{poly}(\log q, k, t, a).$$

□

5.4.1 Uniform Simulation

We remark that we could convert the computation in the proof of Theorem 5.1 into a uniform computation performed by a universal Turing machine. This Turing Machine would take the description of the NTM N, along with its input x as arguments. This can be done because the description of N is used only in computing the lookup tables. An application of [80] on the universal machine would yield a 2-tape machine that computes the number of accepting paths of N, and runs essentially in the same time as the machine constructed in the proof of Theorem 5.1. We describe it formally below:

Theorem 5.15. *We can construct a deterministic two-tape Turing machine that given any k-tape NTM N running in time $t(n)$ and binary string x of length n as input, simulates $t(n)$ steps of $N(x)$ in time*

$$a^{kt(n)/2} H_N^{\sqrt{t(n)}\log t(n)} \cdot q^4 \mathsf{poly}(\log q, k, t(n), a), \tag{68}$$

where a is the alphabet size of N, q is the number of states of N, and H_N is a constant that depends only on a and k.

Proof. We first extend Theorem 5.1 to show that we can build a $k+3$-tape deterministic TM M, which takes in the description of the NTM N and an input string x as input, and performs a uniform simulation in the following running time as in Theorem 5.1.

$$t'(n) = a^{kt/2} H_N^{\sqrt{t}\log t} \cdot q^2 \mathsf{poly}(\log q, k, t, a).$$

To build M we need to show how to perform each action in Theorem 5.1 with a universal TM. We go through the three parts, as listed in the proof of Theorem 5.1, and explain how each of the parts can be performed. Like in Theorem 5.1 we assume that we know an upper bound $t = t(n)$ as a function of the input length n.

1. The preprocessing stage is a set of calculations which are independent of the machine N. This can be performed with no knowledge of the transition function of N.

2. The block trace simulation requires the lookup table T_d which provides the successor state to each state as in Lemma 5.10. Once the DTM has this table, it can perform the simulation. This can be computed from the description of the NTM N, which contains the description of the transition function δ of N.

3. Here we need to perform the graph search simulation. We require the ability to compute the configuration(s) which are successor(s) to a given configuration. This too can be computed from the description of the transition function of the NTM N.

Notice that the running time remains the same as that in Theorem 5.1, up to a polynomial factor. The number of tapes that is required is $k+3$ as in Theorem 5.1, we need k tapes to recreate the tapes of N, one store the lookup table and two for other computations.

Now we apply the Hennie-Stearns construction [49] to M to obtain the required 2-tape DTM which simulates M. Here the second tape serves only to copy "blocks" on the first tape to adjacent locations. The 2-tape TM thus runs in time at worst $O(t'(\log t' + |M|))$. Using the above expression for t', we get that the running time is at most

$$t' \cdot O(kt/2 \log a + \sqrt{t} \log t \log H_N + \text{lower terms}) + t' \cdot |M|$$

where $|M|$ is the program size of M. It is $O(t'(\log t' + |M|))$ not $O(t' \log t' \cdot |M|)$ because the part of the second tape storing the program needs to be consulted only once for each step simulated. The multiplier inside the $O(\dots)$ is absorbed into the poly term of (68), so we are left only to bound and absorb the term $t' \cdot |M|$, The proof of Theorem 5.1 constructs the program size $|M|$ of M to be $O(|N| + kt \log H_N)$ plus lower-order terms. This can be observed by the following argument: The machine M needs to keep track of the basic operations of N, plus it has to keep track of the counters for directional and block traces, for which $O(kt \log H_N)$ is an upper bound. The program size of N, i.e. $|N|$ is given by approximately $a^{2k}3^kq^2$. The multiplier $kt \log H_N$ of t' is likewise absorbed into the poly term, leaving just $|N| \approx a^{2k}3^kq^2$ to deal with. The first part converts the multiplier q^2 into q^4, while the rest can be absorbed into the $H_N^{\sqrt{t(n)}\log t(n)}$ term, by increasing H_N slightly. □

5.5 Implications and Possible Extensions

We have shown techniques by which we can deterministically search the computation tree and count the number of accepting computations of an NTM in time square root of the size of the graph. It would be interesting to see if one could use these techniques to push the running time even lower. Also, it would be interesting to see any lower bounds for the problem.

5.5.1 Simulating Probabilistic Classes

One consequence of being able to count the number of accepting computations exactly is that we could deterministically simulate some randomized complexity classes. We use the following definition of a probabilistic Turing machine and prove the following theorem, almost immediately.

Definition 5.16. *A probabilistic Turing machine is a TM that makes choices, possibly at each step, based on probabilities assigned to each of the choices. We say that a probabilistic TM P accepts a string x, if it accepts x with probability at least* $1/2$.

Theorem 5.17. *A probabilistic k-tape TM P with q states and alphabet size a can be simulated by a multi-tape deterministic TM in time*

$$a^{kt(n)/2}H_N^{\sqrt{t(n)}\log t(n)} \cdot q^2\mathsf{poly}(\log q, k, t(n), a),$$

where $t(n)$ is the running time of N and H_N is a constant depending only on a.

Proof. Given a probabilistic machine P that generates random coins for its computation, one can think of the corresponding nondeterministic Turing machine N, that makes nondeterministic choices for the random coins. For a given input x, P would decide on acceptance based on the number of random choices that lead to acceptance. In terms of N, this translates to the number of different nondeterministic choices that lead to acceptance. □

The above theorem implies a simulation of probabilistic classes in the same running time. We define the complexity class PP below.

Definition 5.18 ([41])**.** *A language L is said to be in the class* Probabilistic Polynomial Time *(denoted by PP) if it can be decided by a probabilistic Turing machine that runs in polynomial time. An alternative characterization is that a language L is in PP if there is a nondeterministic polynomial-time Turing machine N such that x is in L if and only if $M(x)$ has more accepting than rejecting paths.*

Once we define PP as above, the following corollary is immediate.

Corollary 5.19. *Consider a language $L \in \mathsf{PP}$. Let L be decided by a k-tape probabilistic TM with q states and alphabet size a that runs in time $t(n)$. Then L can be simulated in time*

$$a^{kt(n)/2} H_N^{\sqrt{t(n)}\log t(n)} \cdot q^2 \mathsf{poly}(\log q, k, t(n), a).$$

Van Melkebeek and Santhanam [103] gave an unconditional simulation of time-$t(n)$ probabilistic multi-tape Turing machines by Turing machines operating in deterministic time $o(2^t)$. They showed that the exponent in the simulation of probabilistic TM can be reduced by a multiplicative factor smaller than 1 (as compared to our factor of 1/2). Moreover, the class PP contains the classes BPP and BQP. Hence our simulations imply a faster simulation of these classes also.

5.5.2 Polynomial Hierarchy and Alternating TMs

By Toda's theorem [99], we have that the entire polynomial hierarchy (PH)is contained in $\mathsf{P}^{\#\mathsf{P}}$. But we cannot conclude that we have an $\widetilde{O}(a^{kt/2})$ time simulation for classes in PH. This is because Toda's theorem involves a blow-up of the running time when converting a problem in say, Σ_2 to $\#\mathsf{P}$. This negates the advantage that we gain by halving the exponent.

This leads us to a further open question. It would be interesting to see if we can simulate any of the classes in PH by $\#\mathsf{P}$ in the same time bound. This, combined with our counting algorithm, would lead to a faster simulation of the classes in PH. Alternatively, we could try to simulate a time-$t(n)$ alternating TM, for instance a Σ_2-machine A, directly by iterating our uniform simulation for NTM's. This seems to work if the two phases of A are divided neatly into $t(n)/2$ steps each, but encounters a problem if A is existential for $t(n)(1-\varepsilon)$ steps in some computation paths and existential for only $\varepsilon t(n)$ steps in others.

REFERENCES

[1] Agrawal, M., Kayal, N., and Saxena, N., "PRIMES is in P," *Annals of Mathematics*, vol. 160, no. 2, pp. 781–793, 2004.

[2] Ajtai, M., Komlos, J., and Szemerédi, E., "Deterministic simulation in LOGSPACE," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, (New York, NY, USA), pp. 132–140, ACM, 1987.

[3] Alon, N., "Eigenvalues and expanders," *Combinatorica*, vol. 6, pp. 83–96, 1986. 10.1007/BF02579166.

[4] Alon, N., Coja-Oghlan, A., Hàn, H., Kang, M., Rödl, V., and Schacht, M., "Quasi-randomness and algorithmic regularity for graphs with general degree distributions," *SIAM J. Comput.*, vol. 39, pp. 2336–2362, April 2010.

[5] Alon, N., Duke, R. A., Lefmann, H., Rödl, V., and Yuster, R., "The algorithmic aspects of the regularity lemma," *J. Algorithms*, vol. 16, pp. 80–109, 1994.

[6] Alon, N., Fischer, E., Krivelevich, M., and Szegedy, M., "Efficient testing of large graphs," *Annual IEEE Symposium on Foundations of Computer Science*, vol. 0, p. 656, 1999.

[7] Alon, N. and Naor, A., "Approximating the cut-norm via Grothendieck's inequality," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, STOC '04, (New York, NY, USA), pp. 72–80, ACM, 2004.

[8] Alon, N. and Shapira, A., "A characterization of the (natural) graph properties testable with one-sided error," in *Proc. of FOCS 2005*, pp. 429–438, 2005.

[9] Alon, N., Shapira, A., and Stav, U., "Can a Graph Have Distinct Regular Partitions?," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 1, pp. 278–287, 2009.

[10] Alon, N. and Stav, U., "What is the furthest graph from a hereditary property?," *Random Struct. Algorithms*, vol. 33, pp. 87–104, August 2008.

[11] Avart, C., Rödl, V., and Schacht, M., "Every monotone 3-graph property is testable," *Siam Journal on Discrete Mathematics*, vol. 21, pp. 73–92, 2007.

[12] Bansal, N. and Williams, R., "Regularity lemmas and combinatorial algorithms," in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, (Washington, DC, USA), pp. 745–754, IEEE Computer Society, 2009.

[13] Beigel, R. and Eppstein, D., "3-coloring in time $O(1.3289^n)$," *J. Algorithms*, vol. 54, no. 2, pp. 168–204, 2005.

[14] Bertsimas, D. and Tsitsiklis, J., *Introduction to Linear Optimization.* Athena Scientific, 1st ed., 1997.

[15] Bhatia, R., *Matrix Analysis.* New York: Springer-Verlag, 1997.

[16] Björklund, A., "Determinant sums for undirected hamiltonicity," in *FOCS '10: Proceedings of the 51st annual symposium on Foundations of Computer Science*, IEEE, 2010.

[17] Bollobás, B., *Random graphs.* No. 73 in Cambridge studies in advanced mathematics, Cambridge University Press, 2 ed., 2001.

[18] Borgs, C., Chayes, J. T., Lovász, L., Sós, V. T., and Vesztergombi, K., "Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing," *Advances in Mathematics*, vol. 219, no. 6, pp. 1801 – 1851, 2008.

[19] Butler, S., "Relating singular values and discrepancy of weighted directed graphs," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, (New York, NY, USA), pp. 1112–1116, ACM, 2006.

[20] Cheeger, J., "A lower bound for the smallest eigenvalue of the Laplacian," *Problems in Analysis*, pp. 195–199, 1970.

[21] Chung, F. R. K. and Graham, R. L., "Quasi-random set systems," *Journal of The American Mathematical Society*, vol. 4, pp. 151–151, 1991.

[22] Chung, F. R. K. and Graham, R. L., "Quasi-random tournaments," *Journal of Graph Theory*, vol. 15, no. 2, pp. 173–198, 1991.

[23] Chung, F. R. K., "Quasi-random classes of hypergraphs," *Random Structures and Algorithms*, vol. 1, pp. 363–382, August 1990.

[24] Chung, F. R. K. and Graham, R. L., "Quasi-random hypergraphs," *Random Structures and Algorithms*, vol. 1, pp. 105–124, 1990.

[25] Chung, F. R. K. and Graham, R. L., "Sparse quasi-random graphs," *Combinatorica*, vol. 22, no. 2, pp. 217–244, 2002.

[26] Chung, F. R. K., Graham, R. L., and Wilson, R. M., "Quasi-random graphs," *Combinatorica*, vol. 9, pp. 345–362, 1989.

[27] Conlon, D. and Fox, J., "Bounds for graph regularity and removal lemmas," 2011.

[28] Cooper, J. N., "A permutation regularity lemma," *Electr. J. Comb.*, vol. 13, no. 1, 2006.

[29] Coppersmith, D., "Rapid multiplication of rectangular matrices," *SIAM J. Computing*, vol. 11, pp. 467–471, 1982.

[30] Coppersmith, D. and Winograd, S., "Matrix multiplication via arithmetic progressions," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, (New York, NY, USA), pp. 1–6, ACM, 1987.

[31] Dellamonica, D., Kalyanasundaram, S., Martin, D., Rödl, V., and Shapira, A., "A deterministic algorithm for the Frieze-Kannan regularity lemma," in *APPROX/RANDOM 2011* (Goldberg, L. A., Jansen, K., Ravi, R., Rolim, J. D. P., and Rubinfeld, R., eds.), vol. 6845 of *Lecture Notes in Computer Science*, pp. 495–506, Springer, 2011.

[32] DeMillo, R. A. and Lipton, R. J., "A probabilistic remark on algebraic program testing.," *Information Processing Letters*, vol. 7, no. 4, pp. 193–195, 1978.

[33] Duke, R. A., Lefmann, H., and Rödl, V., "A fast approximation algorithm for computing the frequencies of subgraphs in a given graph," *SIAM J. Comput.*, vol. 24, no. 3, pp. 598–620, 1995.

[34] Dyer, M., Frieze, A., and Kannan, R., "A random polynomial-time algorithm for approximating the volume of convex bodies," *J. ACM*, vol. 38, pp. 1–17, January 1991.

[35] Erdős, P. and Rényi, A., "On random graphs, I," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[36] Feige, U. and Kilian, J., "On limited versus polynomial nondeterminism," *Chicago J. Theoret. Comput. Sci.*, pp. Article 1, approx. 20 pp. (electronic), 1997.

[37] Frankl, P. and Rödl, V., "Extremal problems on set systems," *Random Struct. Algorithms*, vol. 20, pp. 131–164, March 2002.

[38] Frieze, A. and Kannan, R., "The regularity lemma and approximation schemes for dense problems," *Annual IEEE Symposium on Foundations of Computer Science*, vol. 0, p. 12, 1996.

[39] Frieze, A. and Kannan, R., "Quick approximation to matrices and applications," *Combinatorica*, vol. 19, pp. 175–220, 1999.

[40] Frieze, A. and Kannan, R., "A simple algorithm for constructing Szemerédi's regularity partition," *Electr. J. Comb*, vol. 6, p. pp. (electronic)., 1999.

[41] Gill, III, J. T., "Computational complexity of probabilistic turing machines," in *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, (New York, NY, USA), pp. 91–95, ACM, 1974.

[42] Gowers, W. T., "Lower bounds of tower type for Szemerédi's uniformity lemma," *Geometric And Functional Analysis*, vol. 7, pp. 322–337, 1997.

[43] Gowers, W. T., "Quasirandomness, counting and regularity for 3-uniform hypergraphs," *Comb. Probab. Comput.*, vol. 15, pp. 143–184, January 2006.

[44] Gowers, W. T., "Hypergraph regularity and the multidimensional Szemerédi theorem," *Annals of Mathematics*, vol. 166, pp. 897–946, 2007.

[45] Gowers, W. T., "Quasirandom groups," *Comb. Probab. Comput.*, vol. 17, pp. 363–387, May 2008.

[46] Graham, R., Rothschild, B., and Spencer, J., *Ramsey Theory.* Wiley, 2nd ed., 1990.

[47] Green, B. and Tao, T., "An arithmetic regularity lemma, an associated counting lemma, and applications," in *An Irregular Mind* (Tóth, G. F., Katona, G. O. H., Lovász, L., Pálfy, P. P., Recski, A., Stipsicz, A., Szász, D., Miklós, D., Bárány, I., Solymosi, J., and Sági, G., eds.), vol. 21 of *Bolyai Society Mathematical Studies*, pp. 261–334, Springer Berlin Heidelberg, 2010.

[48] Griffiths, S., "Quasi-random oriented graphs," 2011.

[49] Hennie, F. C. and Stearns, R. E., "Two-tape simulation of multitape turing machines," *J. ACM*, vol. 13, no. 4, pp. 533–546, 1966.

[50] Homer, S. and Selman, A. L., *Computability and complexity theory.* Texts in Computer Science, New York: Springer-Verlag, 2001.

[51] Hoory, S., Linial, N., and Wigderson, A., "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, pp. 439–561, 2006.

[52] Hopcroft, J., Paul, W. J., and Valiant, L., "On time versus space," *J. Assoc. Comput. Mach.*, vol. 24, no. 2, pp. 332–337, 1977.

[53] Itai, A. and Rodeh, M., "Finding a minimum circuit in a graph," *SIAM J. Comput.*, vol. 7, no. 4, pp. 413–423, 1978.

[54] Kabanets, V. and Impagliazzo, R., "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, pp. 1–46, Dec. 2004.

[55] Kalyanasundaram, S., Lipton, R. J., Regan, K. W., and Shokrieh, F., “Improved simulation of nondeterministic turing machines,” in *MFCS 2010: Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, pp. 453–464, 2010.

[56] Kalyanasundaram, S. and Regan, K. W., “Faster simulation of counting classes.” Manuscript, 2011.

[57] Kalyanasundaram, S. and Shapira, A., “A note on even cycles and quasi-random tournaments.” Submitted, 2011.

[58] Kalyanasundaram, S. and Shapira, A., “A Wowzer type lower bound for the strong regularity lemma.” Submitted, 2011.

[59] Kannan, R., “Towards separating nondeterministic time from deterministic time,” in *Foundations of Computer Science, 1981. SFCS ’81. 22nd Annual Symposium on*, pp. 235–243, Oct. 1981.

[60] Kannan, R., “Alternation and the power of nondeterminism,” in *STOC ’83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 344–346, ACM, 1983.

[61] Kohayakawa, Y., Rödl, V., and Schacht, M., “Discrepancy and eigenvalues of cayley graphs,” *Eurocomb 2003*, vol. 145, pp. 242–246, 2003.

[62] Kohayakawa, Y., Nagle, B., and Rödl, V., “Efficient testing of hypergraphs,” in *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP ’02, (London, UK), pp. 1017–1028, Springer-Verlag, 2002.

[63] Kohayakawa, Y., Rödl, V., and Thoma, L., “An optimal algorithm for checking regularity,” *SIAM J. Comput.*, vol. 32, pp. 1210–1235, May 2003. Earlier verison in SODA ’02.

[64] Komlós, J., Shokoufandeh, A., Simonovits, M., and Szemerédi, E., *The regularity lemma and its applications in graph theory*, pp. 84–112. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[65] Krivelevich, M. and Sudakov, B., “Pseudo-random graphs,” in *More Sets, Graphs and Numbers, Bolyai Society Mathematical Studies 15*, pp. 199–262, Springer, 2006.

[66] Kuczyński, J. and Woźniakowski, H., “Estimating the largest eigenvalue by the power and lanczos algorithms with a random start,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1094–1122, 1992.

[67] Lovász, L., “Very large graphs,” in *Current Developments in Mathematics 2008* (Jerison, D., Mazur, B., Mrowka, T., Schmid, W., Stanley, R., and Yau, S. T., eds.), pp. 67–128, Somerville, MA, USA: International Press, 2009.

[68] Lovász, L. and Szegedy, B., "Limits of dense graph sequences," *J. Comb. Theory Ser. B*, vol. 96, pp. 933–957, November 2006.

[69] Lovász, L. and Szegedy, B., "Szemerédis lemma for the analyst," *Geometric and Functional Analysis GAFA*, vol. 17, pp. 252–270, April 2007.

[70] Lovász, L. and Szegedy, B., "Testing properties of graphs and functions," *Israel Journal of Mathematics*, vol. 178, pp. 113–156, 2010.

[71] Luby, M. and Wigderson, A., "Pairwise independence and derandomization," *Foundations and Trends in Theoretical Computer Science*, vol. 1, pp. 237–301, 2006.

[72] Miltersen, P. B., "Derandomizing complexity classes," in *Handbook of Randomized Computing*, Kluwer Academic Publishers, 2001.

[73] Nagle, B., Rödl, V., and Schacht, M., "The counting lemma for regular k-uniform hypergraphs," *Random Structures and Algorithms*, vol. 28, pp. 113–179, 2006.

[74] Nešetřil, J. and Poljak, S., "On the complexity of the subgraph problem," *Comment. Math. Univ. Carolin.*, vol. 26, no. 2, pp. 415–419, 1985.

[75] Niederreiter, H., "Quasi-monte carlo methods and pseudo-random numbers," *Bulletin of the American Mathematical Society*, vol. 84, no. 6, pp. 957–1041, 1978.

[76] O'Leary, D. P., Stewart, G. W., and Vandergraft, J. S., "Quasirandomness, counting and regularity for 3-uniform hypergraphs," *Mathematics of Computation*, vol. 33, pp. 1289–1292, October 1979.

[77] Papadimitriou, C. H., *Computational complexity.* Reading, MA: Addison-Wesley Publishing Company, 1994.

[78] Paul, W. J., Pippenger, N., Szemerédi, E., and Trotter, W. T., "On determinism versus non-determinism and related problems," in *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pp. 429–438, Nov. 1983.

[79] Pippenger, N., "Probabilistic simulations (preliminary version)," in *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 17–26, ACM, 1982.

[80] Pippenger, N. and Fischer, M. J., "Relations among complexity measures," *J. Assoc. Comput. Mach.*, vol. 26, no. 2, pp. 361–381, 1979.

[81] Rabin, M. O., "Probabilistic algorithm for testing primality," *Journal of Number Theory*, vol. 12, no. 1, pp. 128–138, 1980.

[82] Rödl, V. and Schacht, M., "Generalizations of the removal lemma," *Combinatorica*, vol. 29, pp. 467–501, July 2009.

[83] Rödl, V. and Schacht, M., "Regularity lemmas for graphs," in *Fete of Combinatorics and Computer Science* (Tóth, G. F., Katona, G. O. H., Lovász, L., Pálfy, P. P., Recski, A., Stipsicz, A., Szász, D., Miklós, D., Katona, G. O. H., Schrijver, A., Szonyi, T., and Sági, G., eds.), vol. 20 of *Bolyai Society Mathematical Studies*, pp. 287–325, Springer Berlin Heidelberg, 2010.

[84] Rödl, V. and Skokan, J., "Regularity lemma for k-uniform hypergraphs," *Random Structures and Algorithms*, vol. 25, pp. 1–42, 2004.

[85] Roth, K. F., "On certain sets of integers (II)," *Journal of The London Mathematical Society-second Series*, vol. s1-29, pp. 20–26, 1954.

[86] Ruzsa, I. Z. and Szemerédi, E., "Triple systems with no six points carrying three triangles," *Colloq. Math. Soc. János Bolyai*, vol. 18, pp. 939–945, 1978.

[87] Santhanam, R., "Relationships among time and space complexity classes." http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.5170, 2001.

[88] Savage, J. E., "Computational work and time on finite machines," *J. Assoc. Comput. Mach.*, vol. 19, pp. 660–674, 1972.

[89] Schroeppel, R. and Shamir, A., "A $T \cdot S^2 = O(2^n)$ time/space tradeoff for certain NP-complete problems," in *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979)*, pp. 328–336, New York: IEEE, 1979.

[90] Schwartz, J. T., "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, pp. 701–717, October 1980.

[91] Simonovits, M. and Sós, V. T., "Szemerédi's partition and quasirandomness," *Random Structures & Algorithms*, vol. 2, no. 1, pp. 1–10, 1991.

[92] Szemerédi, E., "On sets of integers containing no k elements in arithmetic progressions," *Polska Akademia Nauk. Instytut Matematyczny. Acta Arithmetica*, vol. 27, pp. 199–245, 1975.

[93] Szemerédi, E., "Regular partitions of graphs," in *Problémes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, (Paris), pp. 399–401, Éditions du Centre National de la Recherche Scientifique (CNRS), 1978.

[94] Tao, T., "A variant of the hypergraph removal lemma," *J. Comb. Theory Ser. A*, vol. 113, pp. 1257–1280, October 2006.

[95] Tao, T., "Structure and randomness in combinatorics," 2007.

[96] Tarjan, R. E. and Trojanowski, A. E., "Finding a maximum independent set," *SIAM J. Comput.*, vol. 6, no. 3, pp. 537–546, 1977.

[97] Thomason, A., "Pseudo-random graphs," in *Proceedings of Random Graphs* (Karoński, M., ed.), vol. 33 of *Annals of Discrete Mathematics*, pp. 307 – 331, 1985.

[98] Thomason, A., "Random graphs, strongly regular graphs and pseudo-random graphs," in *Surveys in Combinatorics* (Whitehead, C., ed.), vol. 123 of *LMS Lecture Note Series*, pp. 173–195, 1987.

[99] Toda, S., "On the computational power of PP and $\oplus P$," in *FOCS '89: Proceedings of the 30th Annual symposium on Foundations of Computer Science*, pp. 514–519, IEEE, 1989.

[100] Trevisan, L., "Pseudorandomness in computer science and in additive combinatorics," in *An Irregular Mind* (Tóth, G. F., Katona, G. O. H., Lovász, L., Pálfy, P. P., Recski, A., Stipsicz, A., Szász, D., Miklós, D., Bárány, I., Solymosi, J., and Sági, G., eds.), vol. 21 of *Bolyai Society Mathematical Studies*, pp. 619–650, Springer Berlin Heidelberg, 2010.

[101] Trevisan, L., "Lecture notes for cs359g: Graph partitioning and expanders," 2011. Available online at http://cs.stanford.edu/people/trevisan/cs359g/index.html.

[102] Turing, A. M., "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937.

[103] van Melkebeek, D. and Santhanam, R., "Holographic proofs and derandomization," *SIAM J. Comput.*, vol. 35, no. 1, pp. 59–90 (electronic), 2005. Earlier version in *Proceedings of the 18th Annual IEEE Conference on Computational Complexity*, 2003.

[104] Williams, R., 2009. Private Communication.

[105] Williams, R., "Improving exhaustive search implies superpolynomial lower bounds," in *STOC '10: Proceedings of the fortysecond annual ACM symposium on Theory of computing*, 2010.

[106] Zippel, R., "Probabilistic algorithms for sparse polynomials," in *Symbolic and Algebraic Computation* (Ng, E., ed.), vol. 72 of *Lecture Notes in Computer Science*, pp. 216–226, Springer Berlin / Heidelberg, 1979.