Expander Graphs and their Applications*
Lecture 1: The Magical Mystery Tour

Notes taken by Ran Gilad-Bachrach

March 2002

Summary:

Since the introduction of *Expander Graphs* during the 1970's they turn to be a significant tool both in theory and practice. They have been used in solving problems in communication and construction of error correcting codes as well as a tool for proving results in number theory and computational complexity. In this course we will explore *Expander Graphs,* both the properties and the use of such graphs will be studied.

The goal of this lecture is to sample the wide range of applications for expander graphs. This should serve as a motivation for the rest of the course.

1 Some Problems

To begin our tour we will look at three questions from three very different domains. Note that in these problems the connection to graph theory, and especially to expander graphs is not clear.

1.1 Hardness results for linear transformation

Maybe the most important open problem in mathematics these days is the famous $P = NP$ (or $P \neq NP$) problem. Although it has been studied for decades now, almost no significant progress has been made. One of the reasons lies in the fact that we have very few problems that are known to be hard. During the 1970'th, Leslie Valiant [5] addressed this problem. He defined the following simple problem:

Problem 1. Let $\mathcal{F}$ be a finite field. Let A be a linear transformation over $\mathcal{F}$, i.e. A is an $n \times n$ matrix. We would like to build a circuit which computes the transformation $x \mapsto Ax$. Each gate of this circuit computes addition or multiplication. How many gates do we need in this network?

Assume for instance that the transformation A represent the Fourier Transform. Cooley and Tukey [1] presented the Fast Fourier Transform (FFT) which computes the transformation using $O(n \log n)$ gates. However there is no matching lower bound so it might be possible to do the computation using only $O(n)$ gates. The implications of a Very Fast Fourier Transform, i.e an $O(n)$ algorithm for computing the transform are hard to over estimate.

By counting the number of circuits and comparing to the number of linear transformations it could be verified that the average size of such circuit is $O\left(n^2/\log n\right)$ gates, however we don't know of any transformation which needs more then $O(n)$ gates.

Valiant [5] tried to present transformations for which the number of gates needed is greater then $O(n)$. He suggested that super regular transformation have this property:

Definition 1 (Super Regular Matrix). A matrix A is *Super Regular* if any rectangular sub-matrix of A has full rank.

The main observation of Valiant was that if we look at the graph layout of a circuit which computes a super regular matrix then this graph is a *Super Concentrator*:

* Lecture Notes for a course given by Nati Linial and Avi Wigderson at the Hebrew University, Israel.

Figure 1: Leslie G. Valiant

Definition 2 (Super Concentrator). A graph G is a *Super Concentrator* if it has n input vertexes denoted by I and n output vertexes denoted by O such that for every k and every $S \subseteq I$ and $T \subseteq O$ of size k (i.e. $|S| = |T| = k$) there exists k paths in G from S to T which are vertexes disjoint.

Valiant conjectured that any *Super Concentrator* graph must have $\gg n$ edges and hence any circuit which computes a super regular matrix must have $\gg n$ gates. However, Valiant himself disproved the conjecture and presented super concentrators with $O(n)$ edges, and as you might have guessed this is where expanders come into the picture.

For the moment we will skip to a totally different problem.

1.2 Error Correcting Codes

One of the most fundamental problems in communication is noise. Assume that Alice has a message of k bits which she would like to deliver to Bob over some communication channel. The problem is that the channel might interfere in the way and thus the message that Bob receives might be different then the one that Alice sent.

During the 1940's Clude Elwood Shannon has developed the theory of communication which is called Information Theory. In his innovative paper "A Mathematical Theory of Communication" [4] the problem of communication over noisy channel is one of the problems he addressed. Let us first define the problem[1]:

Problem 2 (communication over noisy channel). Alice and Bob can communicate over a noisy channel that might change a proportion p of the bits sent through it. How can Alice send Bob a message of k bits?

Shannon presented an answer to the above given question. He suggested building a dictionary (or code) $C \subseteq \{0,1\}^n$ such that $|C| = 2^k$. Every k-bits message is encoded by a code word in C and transmitted. Bob receives n bits and finds the closest code word in C in terms of hamming distance and determines the k-bits associated with it. If the minimal distance between two words in C is greater then $2pn$ it is guaranteed that the k-bits that Bob will find is exactly the bits Alice encoded.

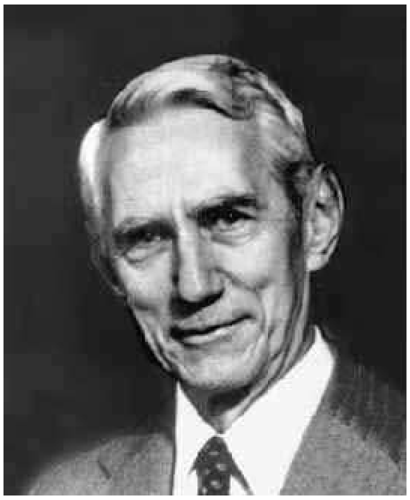
Therefore the problem of communicating over noisy channel is reduced to the problem of finding a good dictionary (code). A good dictionary is one that is both big (i.e. $|C|$ is big) and at the same time the length of the words in C is small. This is seen from the next definition:

Definition 3 (the rate and distance of a dictionary). Let $C \subseteq \{0,1\}^n$ be a dictionary. The **rate** of the dictionary is defined as

$$R = \frac{\log |C|}{n}$$

[1] The problem as described here is a simplification of the original problem presented by Shannon.

Figure 2: Clude Shannon.



while the **distance** of the code is

$$\delta = \frac{\min_{c_1 \neq c_2 \in C} d_H(c_1, c_2)}{n}$$

where d_H is the hamming distance.

As we saw before the distance of a dictionary governs it's ability to overcome noisy channels while the rate counts the efficiency of the code. At this point we can refine the problem just state:

Problem 3 (refined communication problem). Is it possible to design a series of dictionaries $\{C_k\}_{k=1}^{\infty}$ such that $|C_k| = 2^k$, the distance of each dictionary is greater than $\delta_0 > 0$ and the rate of each code is greater than $R_0 > 0$.

We will see that a solution to this problem can be found using expander graphs. However, we will now present yet another problem.

1.3 De-randomizing Algorithms

Rabin [3] presented in 1980 an algorithm for checking primality. Given an integer x of k bits and a set of k random bits r the algorithm computes a function $f(x, r)$ such that if x is primal $f(x, r) = 1$, on the other hand if x is not primal $f(x, r) = 1$ with probability smaller then $1/4$. Applying this algorithm over and over again can reduce the error to be arbitrary small. However this process involves the use of more and more random bits.

The primality test is a special case of a Random Polynomial algorithm (RP). Let $L \subseteq \{0, 1\}^k$ be a language. An algorithm which decides on $x \in \{0, 1\}^k$ weather it is in L or not is Random Polynomial, if it runs in polynomial time and using poly(k) random bits and gives an answer which is always one if $x \in L$ and has probability smaller than $1/4$ to give 1 if $x \notin L$.

Problem 4 (Saving Random Bits). Assume that that $L \subseteq \{0, 1\}^k$ has a random polynomial algorithm. How many random bits are needed in order to give an answer with probability of mistake smaller then $1/d$?

2 Magical Graphs

In the previous section we presented three problems which seems to be unrelated. We will now present a new object called Magical Graph which will enable us to find a solution for all these problems.

Figure 3: Michael Rabin

Definition 4 (Magical Graph). Let G be a two sided graph with n vertexes on each side. Let L be the vertexes on the left side and R the vertexes on the right. Assume that any vertex in L has d neighbors in R. We say that G is (d, n) **magical graph** if it has the following two properties

1. For any $S \subseteq L$ such that $|S| \leq \frac{n}{3d} \Longrightarrow |\Gamma(S)| \geq |S| \frac{d}{4}$
2. For any $S \subseteq L$ such that $\frac{n}{3d} < |S| \leq \frac{n}{2} \Longrightarrow |\Gamma(S)| \geq |S| + \frac{n}{3d}$

where $\Gamma(S)$ is the set of neighbors of S in G.

We will now turn to explore some properties of magical graphs.

Lemma 5. *For each d* ≥ 8 *and sufficiently large* n *there exists a* (d, n) *magical graph.*

Proof. Construct a random graph as follows: for each vertex $v \in L$ choose randomly d vertexes in R and connect them with v. We claim that with high probability the graph generated by this process is a magical graph.

Let $S \subseteq L$ be such that $s = |S| \leq \frac{n}{3d}$. Let $T \subseteq R$ be such that $t = |T| < |S| \frac{d}{4}$.

Let $X_{S,T}$ be an indicator random variable for the event that all the edges from S go to T. It is clear that if $\sum X_{S,T} = 0$ then the first property in the definition of magical graphs hold.

The probability of the event $X_{S,T}$ is $\left(\frac{t}{n}\right)^{sd}$ and therefore using a union bound

$$\begin{aligned}
\Pr\left[\sum_{S,T} X_{S,T} > 0\right] &\leq \sum_{S,T} \Pr\left[X_{S,T} = 1\right] \\
&= \sum_{S,T} \left(\frac{t}{n}\right)^{sd} \\
&\leq n^2 \binom{n}{\frac{n}{3d}} \binom{n}{\frac{n}{12}} \left(\frac{1}{12}\right)^{n/3} \\
&\cong n^2 2^{n\mathcal{H}\left(\frac{1}{3d}\right) + n\mathcal{H}\left(\frac{1}{12}\right) + \frac{n}{3}\log\frac{1}{12}} \\
&\leq n^2 2^{-n/8}
\end{aligned}$$

When n is sufficiently large $n^2 2^{-n/8}$ is smaller then 0.25 and therefore the probability that requirement 1 in the definition of magical graph will holds is greater than 0.75.

We use the same technique to bound the probability that requirement 2 in the definition of magic graph hold. For every $S \subset L$ such that $\frac{n}{3d} < |S| \leq \frac{n}{2}$ and $T \subset R$ such that $|T| < |S| + \frac{n}{3d}$ let $Y_{S,T}$ be an indicator random variable for the event that all the edges from S go to T. As in the previous case, if $\sum Y_{S,T} = 0$ then the second property in the definition of magical graphs hold.

The probability of the event $Y_{S,T}$ is $\left(\frac{t}{n}\right)^{sd}$ and therefore using a union bound

$$\begin{aligned}
\Pr\left[\sum_{S,T} Y_{S,T} > 0\right] &\leq \sum_{S,T} \Pr\left[Y_{S,T} = 1\right] \\
&= \sum_{S,T} \left(\frac{t}{n}\right)^{sd} \\
&\leq n^2 \begin{pmatrix} n \\ \frac{n}{2} \end{pmatrix} \begin{pmatrix} n \\ \frac{n}{2} \end{pmatrix} \left(\frac{\frac{n}{2} + \frac{n}{3d}}{n}\right)^{\frac{n}{2}d} \\
&\cong n^2 2^{2n\mathcal{H}\left(\frac{1}{2}\right) + \frac{nd}{2}\log\left(\frac{1}{2} + \frac{1}{3d}\right)} \\
&\leq n^2 2^{n\left(2 - \frac{d}{3}\right)}
\end{aligned}$$

For n sufficiently large $n^2 2^{n\left(2 - \frac{d}{3}\right)} \leq 0.25$ and hence the second property of magical graph holds with probability 0.75 at least.

Finally if we chose a sufficiently large n we get that the two requirements of magical graphs hold with probability greater than 0.5. Therefore not only that there exist an (n, d) magic graph but there are many of those.

□

Before introducing the solutions to the above mentioned problems, we will present a small variation on magical graphs. We will delete $\frac{n}{4d^2}$ vertexes from R, i.e. the right side of the graph such that the main properties of the graph will remain:

Lemma 6. *Let G be a magical graph then there exists $B \subset R$ such that $|B| \geq \frac{n}{4d^2}$ and for each vertex $v \in L$ there is at most one neighbor in B.*

Proof. We will present an algorithm which constructs the set B. We begin by holding the two sets of vertexes $L_0 = L$ and $R_0 = R$ and we reset B to be the empty set.

At each iteration we choose $v \in R_i$ with degree at most $2d$. We add v to B and then construct L_{i+1} such that $L_{i+1} = L_i \setminus \Gamma(v)$ and $R_{i+1} = R_i \setminus \Gamma(\Gamma(v))$, i.e. we delete all the neighbors of v from L_i and delete all the second-degree neighbors of v from R_i. We keep the process running as long as there is a vertex $v \in R_i$ with degree at most $2d$.

From the way we constructed the set B it is clear that any $u \in L$ has at most one neighbor in B. We would like to count how many iterations can we do with the above algorithm, this will give us a lower bound on the size of B.

At each step we have that $|L_i| \geq n - 2di$ since the vertex v which we add to B has degree at most $2d$. Also we have that $|R_i| \geq n - d(n - |L_i|)$ since each vertex in L has degree d. Since the number of vertexes in the graph induced on $L_i \cup R_i$ is at most $d|L_i|$, the average degree of the vertexes in R_i is at most

$$\frac{d|L_i|}{|R_i|} \leq \frac{d|L_i|}{n - d(n - |L_i|)}$$

and hence as long as $|L_i| \geq \left(1 - \frac{1}{2d-1}\right)n$ the average degree of the vertexes in R_i is at most $2d$ and therefore there exist a vertex with degree at most $2d$.

Since $|L_i| \geq n - 2di$ we have that as long as $i \leq \frac{n}{4d^2}$ that $|L_i| \geq \left(1 - \frac{1}{2d-1}\right)n$ as required. Therefore we can build a set B of size at $\frac{n}{4d^2}$.

□

We will call **modified magical graph** a magical graph that a set of size $\frac{n}{4d^2}$ of vertexes were deleted from the right side as described in lemma 6.

Since magical graphs exists as we saw in lemma 5 and can be modified as we saw in lemma 6 we now turn to use this construction to solve the problems presented in the first section of this lecture.

2.1 A Super Concentrator with $O(n)$ edges

As we saw in section 1.1 Valiant's conjecture was that a super concentrator must have many edges. This sounds reasonable from the definition of super concentrators (see definition 2). However we will see that using magical graphs it is possible to build such graphs with only $O(n)$ edges.

Let G be a modified magical graph such that there are n vertexes on the left side of G but only $n - \frac{n}{4d^2}$ vertexes on the right side as we saw in lemma 6.

By the construction of G we have that for each $S \subset L$ such that $|S| \leq \frac{n}{2}$ has $|\Gamma(S)| \geq |S|$. Hence by Hall's theorem [2, Theorem 2.1.2] for any $|S| \leq \frac{n}{2}$ in L there is a perfect matching from S to $\Gamma(S)$. We will use these facts to build a super concentrator.

The construction can be presented recursively. Let n_0 be the minimal size of modified magical graph. If we are required to build a super concentrator with less then n_0 vertexes we just return the full two sided graph. The full two sided graph is a concentrator with n_0^2 edges (we will use notation $S(n)$ for the number of edges).

Assume we would like to build a super concentrator with $n > n_0$ vertexes. Let G be a modified magical graph with n vertexes on the left side and $n - \frac{n}{4d^2}$ vertexes on the right. Let C be a super concentrator with $n - \frac{n}{4d^2}$ input and output edges. Such a concentrator exists according to our induction assumption.

Using G and C we will construct a new concentrator with n inputs and outputs. The inputs of the new concentrator will be the left side of G. We connect the right hand side of G to the inputs of C. We will place another copy of G on the outputs of C, this is illustrated in figure 4(a). Finally we add direct edges between the two copies of G, each vertex on the left side of G we placed in the input is connected to the matching vertex on the left side of G we placed in the output as illustrated in figure 4(b).

We would like to show that the graph constructed is indeed concentrator and count the number of edges in this graph. Let S be a set of vertexes from the input of the new graph and T be vertexes on the output such that $|S| = |T| = k$. If $k \leq n/2$ then due to the properties of the modified magical graph G we know that $|\Gamma(S)| \geq |S|$ and $|\Gamma(T)| \geq |T|$. Using Halls marriage theorem it is possible to construct a perfect matching between S and $\Gamma(S)$ and on the other side between T and $\Gamma(T)$. Since C is a super concentrator, the matches of S in $\Gamma(S)$ and of T in $\Gamma(T)$ can be connected by k disjoint paths and hence S and T can be connected by disjoint paths.

If the two sets S and T are big, i.e. $|S| = |T| = k > n/2$ then there must exists at least $k - n/2$ vertexes in S that are matched to vertexes in T by direct edges. These edges form paths and hence we can exclude these vertexes from S and their matches from T. After doing so we are left with groups of size $n/2$ at most which we already know how to treat.

After we proved that the graph we constructed is a super concentrator we turn to count the number of edges. Let $S(n)$ be the number of edges in the graph with n inputs. From the construction we know that $S(n) = |C| + 2|G|$. We also know that $|G| = nd$ and $|C| = S\left(n\left(1 - \frac{1}{4d^2}\right)\right)$. Hence we obtain a recursive formula for $S(n)$:

1. for $n > n_0$ we have that $S(n) \leq 2nd + S\left(n\left(1 - \frac{1}{4d^2}\right)\right)$.
2. for $n \leq n_0$ we have that $S(n) \leq n^2$.

Solving this recursive formula we get

$$S(n) \leq cn$$

such that $c = n_0 + 8d^3$.

Therefore, using magical graphs it is possible to construct super concentrators with $O(n)$ edges.

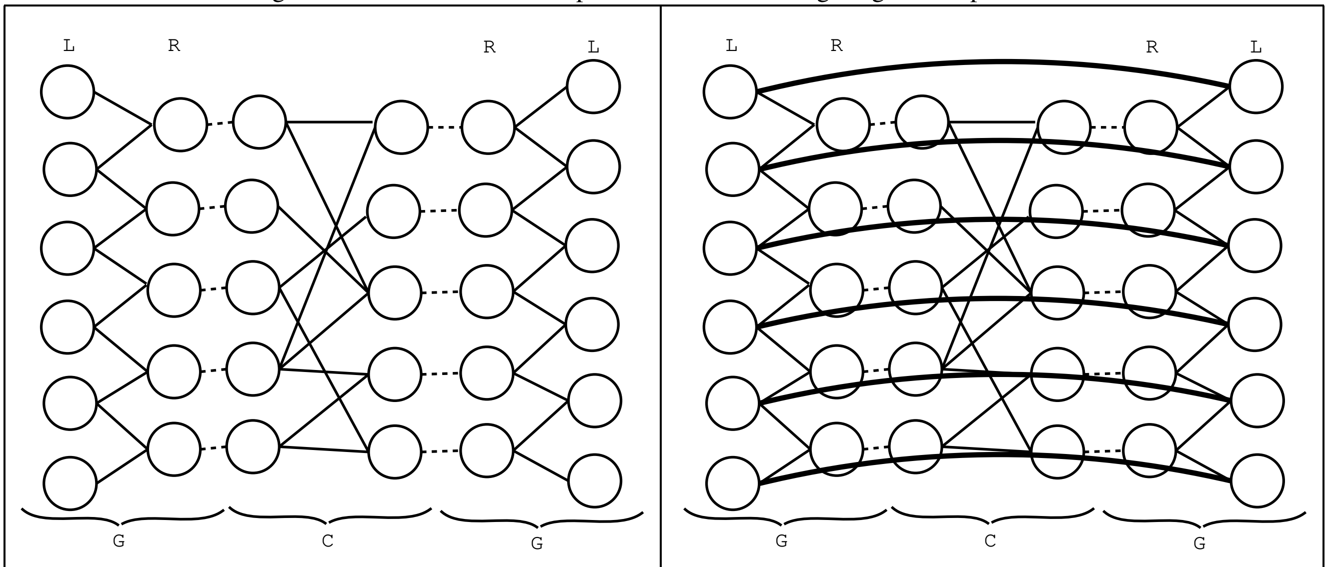
2.2 Error Correcting Codes

We now turn to present a solution to Shannon's problem of correcting errors over communication channels. Again let G be a modified magical graph, i.e. a magical graph such that it's right side consist of less then $n(1 - \frac{1}{d^2})$ vertexes.

Let $S \subset L$ be such that $|S| \leq \frac{n}{3d}$ then there exists $v \in S$ and $u \in R$ such that v is the only neighbor of u in the group S, i.e. $u \in \Gamma(S) \setminus \Gamma(S \setminus \{v\})$. This follows since $|\Gamma(S)| \geq (\frac{3d}{4} - 1)|S| > \frac{d}{2}|S|$. To prove the existence of u and v, consider the set E of edges between S and $\Gamma(S)$. Then $d|S| \geq |E| \geq 2|\Gamma(S)|$ and so $|\Gamma(S)| \leq \frac{d}{2}|S|$ which is a contradiction.

We use this construction to build a code $C \subset \{0,1\}^n$ of size 2^k such that the hamming distance between any two distinct code words is at least $\frac{n}{3d}$ so the code has distance $\frac{1}{3d}$ and the rate is $\frac{k}{n} = \frac{1}{3d}$.

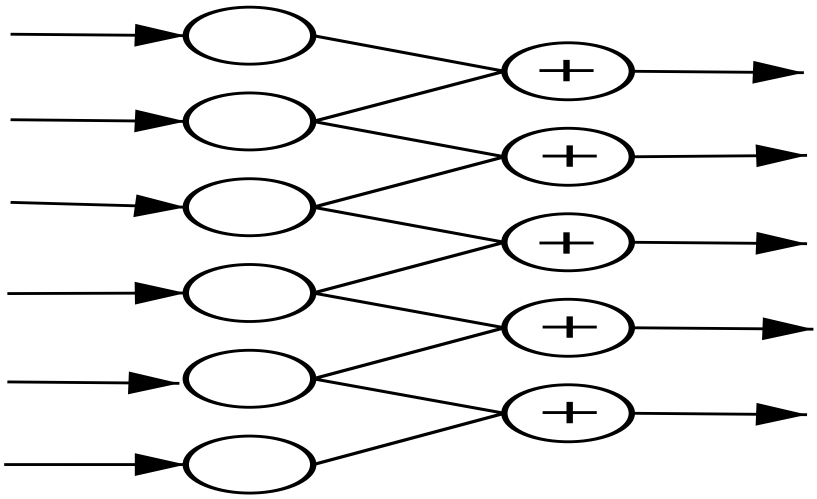
Figure 4: A construction of Super Concentrators using Magical Graphs



(a) first stage

(b) second stage

Figure 5: A construction of an error correcting code



Let G be a modified magical graph. We will view the graph G as a function from $\{0,1\}^n$ to $\{0,1\}^{n(1-\frac{1}{d^2})}$ by assigning a parity function to each vertex in the right side, i.e. $G(x)_u = \oplus_{v\in\Gamma(u)} x_v$.

$$C = \{x \in \{0,1\}^n \; | G(x) = 0\}$$

I.e. a word x is in the code C if the parity assigned to each vertex on the right side is zero. Figure 5 demonstrates the code.

C is a linear sub space of $\{0,1\}^n$ defined by $n(1-\frac{1}{4d^2})$ linear equations and hence $|C| \geq 2^{n/4d^2}$. Since C is a linear code (i.e. a linear sub-space) the minimal distance between two code words in C is the minimal weight of a non-zero code word in C. Let $x \in \{0,1\}^n$. We can look at x as an indicating function of vertexes in L. Let S be the set of vertexes to which x assigns the value 1. If $|S| < \frac{n}{3d}$ then there exists a unique neighbor, i.e there exists $u \in R$ and $v \in S$ such that v is the only neighbor of u in S. Hence the parity function associated with u will assign the value 1 to x and hence $x \notin C$. Therefore the minimal distance between two code words in C is at least $n/3d$.

Therefore we presented a way to construct "good" dictionaries (codes) using magical graphs.

2.3 De-randomizing Random Algorithms

The last problem we presented was that of random algorithms. Let LANG be a language such that there exists an algorithm such that when it receives x of size k and a string r of k random bits, it calculates a function $f(x,r)$ such that if $x \in$ LANG then $f(x,r) = 1$ but if $x \notin$ LANG then $f(x,r) = 1$ with probability at most $1/12$ over the choice of r.

Let G be a magical graph over $n = 2^k$ vertexes. By the definition of such graphs we know that for every $S \subseteq L$ such that $|S| \geq \frac{n}{3d}$ we have that $|\Gamma(S)| \geq \frac{n}{12}$. Now assume that $x \notin$ LANG. Let B be the "bad" set i.e. $B = \left\{r \in \{0,1\}^k \mid f(x,r) = 1\right\}$. We know that $|B| \leq \frac{n}{12}$.

For each string of k random bits we assign a vertex in L. The graph G then direct us to d string of k bits which are associated to vertexes in R, we will call these strings $r_1, \ldots, r_d$. Our algorithm will apply the function $f(x,\cdot)$ with $r_1, \ldots, r_d$. If all the values we receive were 1 then we will predict that $x \in$ LANG. If $f(x, r_j) = 0$ for some j then we know for sure that $x \notin$ LANG.

Our algorithm will fail only if $x \notin$ LANG and $r_1, \ldots, r_d \in B$, i.e. $\Gamma(r) \subseteq B$. Let $S \subset L$ be the set of vertexes for which all their neighbors are in B, so $S = \{v \in L \mid \Gamma(v) \subseteq B\}$. Since $|B| \leq \frac{n}{12}$ we have that $|S| \leq \frac{n}{3d}$ and therefore our algorithm will fail with probability $\frac{1}{3d}$ at most, using only k random bits.

3 Conclusions

In the first section of this lecture we presented three problems of different nature. All these problem had no direct connection to graph theory. However we saw that by constructing magical graphs we could find a solution to these problems. During our discussion we explored some of the features of magical graphs.

What other magic can these graphs do? what properties do they have. Can we construct these graphs efficiently? All these questions are the topic of this course. We will explore both the theory and applications of magical graphs.

And one last word, the magical graphs we used in this lecture are a special case of the exciting family of ***Expanders***.

References

[1] J. W. Cooley and Tukey J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 90(19):297–301, 1965.

[2] R. Diestel. *Graph Theory*. Springer-Verlag, 1997.

[3] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(128-138), 1980.

[4] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, July and October 1948.

[5] L. G. Valiant. Graph-theoretic properties in computational complexity. *JCSS*, 13(3):278–285, 1976.