Algebraic Constructions for Expanders

Mikhail Volkov

Ural Federal University, Ekaterinburg, Russia

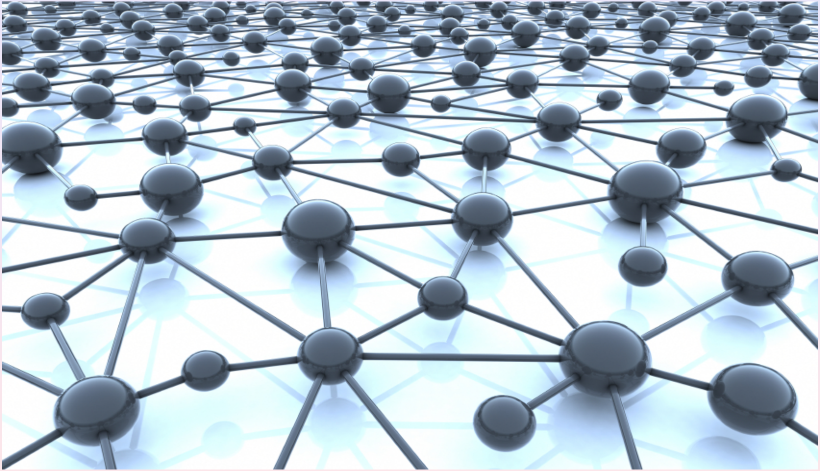
AAA88, June 20th, 2014

A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges).

A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges).

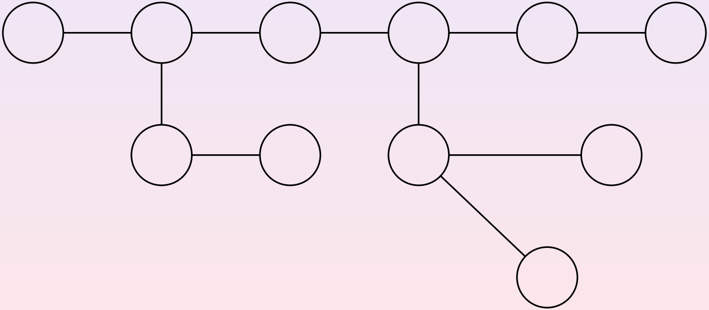


A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges). The graph must be connected.

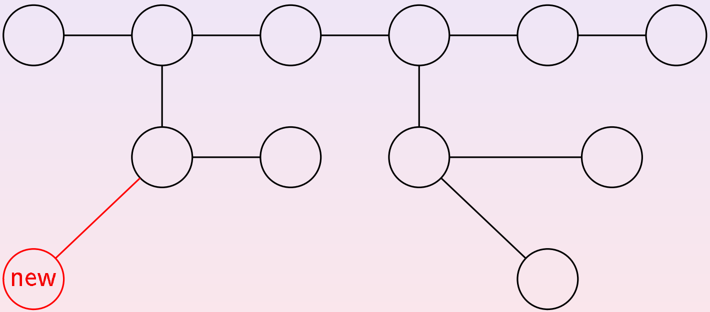
A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges). The graph must be connected. One possible solution: a tree.



A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges). The graph must be connected. One possible solution: a tree.
Pro: low costs of expanding — one extra line for each new node:

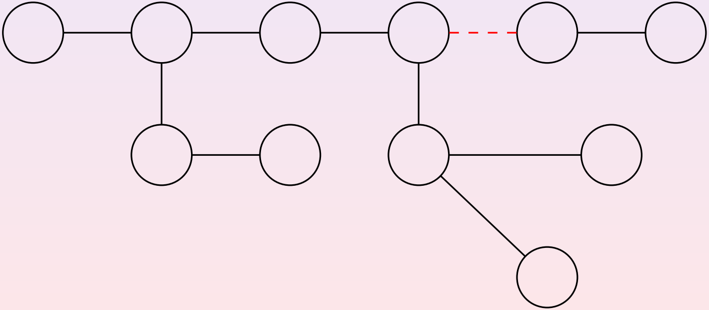


Thus, expanding a network with n nodes to one with twice as many nodes requires only n extra lines.

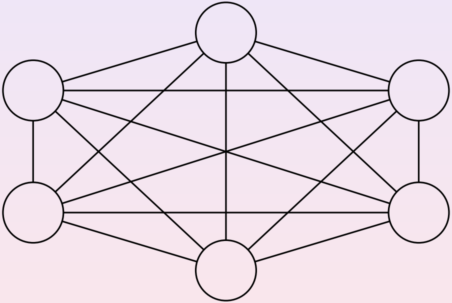
A Motivating Example I

Suppose you want to establish a network consisting of many nodes (vertices) some of which may be connected by lines (edges). The graph must be connected. One possible solution: a tree.

Contra: bad connectivity — removing any edge disconnects the whole network:



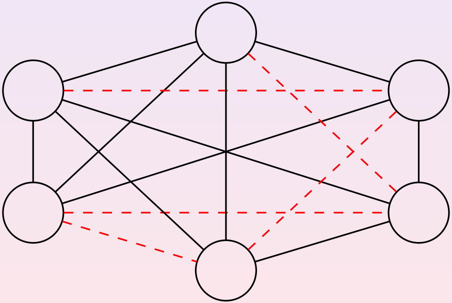
An "opposite" solution: a clique (complete graph).



A Motivating Example II

An "opposite" solution: a clique (complete graph).

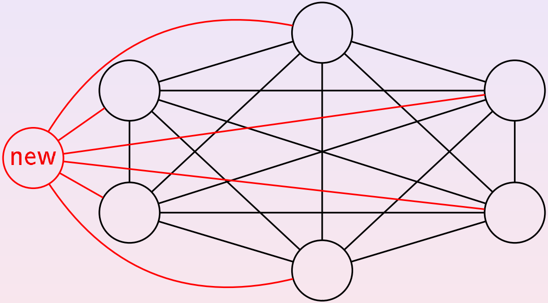
Pro: good connectivity — the network stays connected even after removing several edges:



A Motivating Example II

An “opposite” solution: a clique (complete graph).

Contra: high costs of expanding — n extra lines for each new node:



Thus, expanding a network with n nodes to one with twice as many nodes requires around $\frac{3}{2}n^2$ extra lines.

What Do We Seek?

Summary:

What Do We Seek?

Summary:

Expansion costs
Connectivity

What Do We Seek?

Summary:

	Trees
Expansion costs	Good
Connectivity	Bad

What Do We Seek?

Summary:

	Trees	Cliques
Expansion costs	Good	Bad
Connectivity	Bad	Good

What Do We Seek?

Summary:

	Trees	Cliques	We seek ("expanders")
Expansion costs	Good	Bad	Good
Connectivity	Bad	Good	Good

What Do We Seek?

Summary:

	Trees	Cliques	We seek ("expanders")
Expansion costs	Good	Bad	Good
Connectivity	Bad	Good	Good

We need a family of graphs with good connectivity properties and low expansion costs.

What Do We Seek?

Summary:

	Trees	Cliques	We seek ("expanders")
Expansion costs	Good	Bad	Good
Connectivity	Bad	Good	Good

We need a family of graphs with good connectivity properties and low expansion costs.

Low expansion costs $\Rightarrow$ a constant upper bound for degrees $\Rightarrow$ d-regular graphs (each vertex is incident to exactly d edges).

What Do We Seek?

Summary:

	Trees	Cliques	We seek ("expanders")
Expansion costs	Good	Bad	Good
Connectivity	Bad	Good	Good

We need a family of graphs with good connectivity properties and low expansion costs.

Low expansion costs $\Rightarrow$ a constant upper bound for degrees $\Rightarrow$ d-regular graphs (each vertex is incident to exactly d edges). Then expanding a graph with n nodes to one with twice as many nodes requires only dn extra edges.

What Do We Seek?

Summary:

	Trees	Cliques	We seek ("expanders")
Expansion costs	Good	Bad	Good
Connectivity	Bad	Good	Good

We need a family of graphs with good connectivity properties and low expansion costs.

Low expansion costs $\Rightarrow$ a constant upper bound for degrees $\Rightarrow$ d-regular graphs (each vertex is incident to exactly d edges). Then expanding a graph with n nodes to one with twice as many nodes requires only dn extra edges.

How do we formalize good connectivity?

Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S, T) = \{(u, v) \mid u \in S,\ v \in T,\ (u, v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S,T) = \{(u,v) \mid u \in S,\ v \in T,\ (u,v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.

Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S, T) = \{(u, v) \mid u \in S,\ v \in T,\ (u, v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.

Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S, T) = \{(u, v) \mid u \in S,\ v \in T,\ (u, v) \in E\}$$

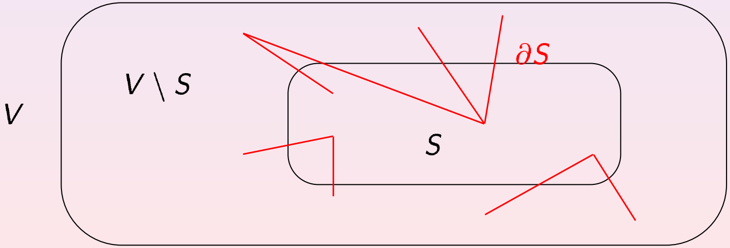
The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.

Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S, T) = \{(u, v) \mid u \in S,\ v \in T,\ (u, v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.



Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S, T) = \{(u, v) \mid u \in S,\ v \in T,\ (u, v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.
The expansion ratio of G is

$$h(G) = \min_{|S| \leq \frac{n}{2}} \frac{|\partial S|}{|S|}$$

Towards A Formal Definition

Let $G = (V, E)$ be an undirected and d-regular graph; $|V| = n$; loops and multiple edges are allowed. For $S, T \subset V$, let $E(S, T)$ be the set of edges from S to T:

$$E(S,T) = \{(u,v) \mid u \in S,\ v \in T,\ (u,v) \in E\}$$

The boundary of $S \subset V$ is the set $\partial S = E(S, V \setminus S)$; the set of edges emanating from S to its complement.
The expansion ratio of G is

$$h(G) = \min_{|S| \le \frac{n}{2}} \frac{|\partial S|}{|S|}$$

Good connectivity $\Rightarrow$ every "small" set of vertices has a relatively big boundary $\Rightarrow$ the expansion ratio is not too small.

A Formal Definition

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) \geq \varepsilon$ for all i.

Families of expander graphs have found extensive and surprising applications in designing algorithms and error correcting codes; they have also been used in proofs of many important results in computational complexity theory, in cryptography, and also in several areas of pure mathematics and statistical physics.
An excellent source for basic material, a wide range of applications as well as research up to 2005 is the monograph by Shlomo Hoory, Nati Linial, and Avi Wigderson: "Expander graphs and their applications", Bull. Amer. Math. Soc., 43(4):439–561, 2006.
"... expansion is a fundamental mathematical concept, well deserving to be thoroughly investigated on its own."

AAA88, June 20th, 2014

A Formal Definition

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) \geq \varepsilon$ for all i.

Families of expander graphs have found extensive and surprising applications in designing algorithms and error correcting codes; they have also been used in proofs of many important results in computational complexity theory, in cryptography, and also in several areas of pure mathematics and statistical physics.

An excellent source for basic material, a wide range of applications as well as research up to 2005 is the monograph by Shlomo Hoory, Nati Linial, and Avi Wigderson: "Expander graphs and their applications", Bull. Amer. Math. Soc., 43(4):439–561, 2006.

"... expansion is a fundamental mathematical concept, well deserving to be thoroughly investigated on its own."

A Formal Definition

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) \geq \varepsilon$ for all i.

Families of expander graphs have found extensive and surprising applications in designing algorithms and error correcting codes; they have also been used in proofs of many important results in computational complexity theory, in cryptography, and also in several areas of pure mathematics and statistical physics.
An excellent source for basic material, a wide range of applications as well as research up to 2005 is the monograph by Shlomo Hoory, Nati Linial, and Avi Wigderson: "Expander graphs and their applications", Bull. Amer. Math. Soc., 43(4):439–561, 2006.

"...expansion is a fundamental mathematical concept, well deserving to be thoroughly investigated on its own."

A Formal Definition

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) \geq \varepsilon$ for all i.

Families of expander graphs have found extensive and surprising applications in designing algorithms and error correcting codes; they have also been used in proofs of many important results in computational complexity theory, in cryptography, and also in several areas of pure mathematics and statistical physics.
An excellent source for basic material, a wide range of applications as well as research up to 2005 is the monograph by Shlomo Hoory, Nati Linial, and Avi Wigderson: "Expander graphs and their applications", Bull. Amer. Math. Soc., 43(4):439–561, 2006.
"... expansion is a fundamental mathematical concept, well deserving to be thoroughly investigated on its own."

An Application: Error Correcting Codes

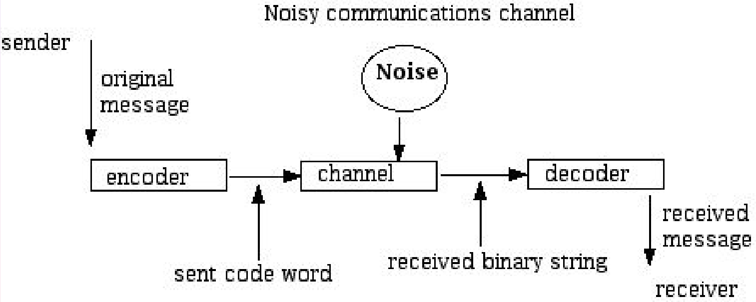
This talk is mainly about the role of algebra in studying expanders rather than expanders themselves and their applications. So here I mention only one application: designing error correcting codes.

An Application: Error Correcting Codes

This talk is mainly about the role of algebra in studying expanders rather than expanders themselves and their applications. So here I mention only one application: designing error correcting codes.

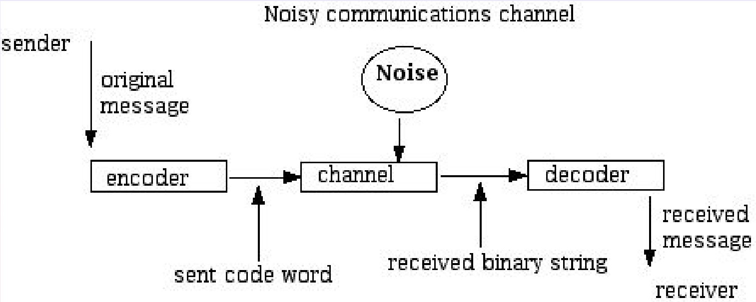
An Application: Error Correcting Codes

This talk is mainly about the role of algebra in studying expanders rather than expanders themselves and their applications. So here I mention only one application: designing error correcting codes.



An Application: Error Correcting Codes

This talk is mainly about the role of algebra in studying expanders rather than expanders themselves and their applications. So here I mention only one application: designing error correcting codes.



Alice and Bob communicate over a noisy channel. A fraction p of the bits sent may be altered. What is the smallest number of bits that Alice can send, assuming she wants to communicate an arbitrary k-bit message, so that Bob should be able to unambiguously recover the original message?

Error Correcting Codes I

The basic idea (due to Claude Shannon) is to create a code $C \subset \{0,1\}^n$ of size $|C| = 2^k$ such that the Hamming distance between any two strings in C is greater than $2pn$. (The Hamming distance $d_H(u,v)$ is the number of coordinates i such that $u_i \neq v_i$.) Alice and Bob agree about the encoding: a bijection $\varphi : \{0,1\}^k \to C$. If Alice needs to send a message $x \in \{0,1\}^k$, she transmits $\varphi(x) \in C$. Bob receives $y \in \{0,1\}^n$ which is a corrupted version of $\varphi(x)$. Since $d_H(y, \varphi(x)) \leq pn$, Bob can recover $\varphi(x)$ as the string $z \in C$ that is closest to y and then find $x = \varphi^{-1}(z)$.

Error Correcting Codes I

The basic idea (due to Claude Shannon) is to create a code $C \subset \{0,1\}^n$ of size $|C| = 2^k$ such that the Hamming distance between any two strings in C is greater than $2pn$. (The Hamming distance $d_H(u,v)$ is the number of coordinates i such that $u_i \neq v_i$.) Alice and Bob agree about the encoding: a bijection $\varphi : \{0,1\}^k \to C$. If Alice needs to send a message $x \in \{0,1\}^k$, she transmits $\varphi(x) \in C$. Bob receives $y \in \{0,1\}^n$ which is a corrupted version of $\varphi(x)$. Since $d_H(y, \varphi(x)) \leq pn$, Bob can recover $\varphi(x)$ as the string $z \in C$ that is closest to y and then find $x = \varphi^{-1}(z)$.

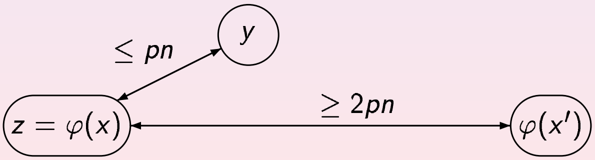
Error Correcting Codes I

The basic idea (due to Claude Shannon) is to create a code $C \subset \{0,1\}^n$ of size $|C| = 2^k$ such that the Hamming distance between any two strings in C is greater than $2pn$. (The Hamming distance $d_H(u,v)$ is the number of coordinates i such that $u_i \neq v_i$.) Alice and Bob agree about the encoding: a bijection $\varphi : \{0,1\}^k \to C$. If Alice needs to send a message $x \in \{0,1\}^k$, she transmits $\varphi(x) \in C$. Bob receives $y \in \{0,1\}^n$ which is a corrupted version of $\varphi(x)$. Since $d_H(y,\varphi(x)) \leq pn$, Bob can recover $\varphi(x)$ as the string $z \in C$ that is closest to y and then find $x = \varphi^{-1}(z)$.

Error Correcting Codes I

The basic idea (due to Claude Shannon) is to create a code $C \subset \{0,1\}^n$ of size $|C| = 2^k$ such that the Hamming distance between any two strings in C is greater than $2pn$. (The Hamming distance $d_H(u,v)$ is the number of coordinates i such that $u_i \neq v_i$.) Alice and Bob agree about the encoding: a bijection $\varphi : \{0,1\}^k \to C$. If Alice needs to send a message $x \in \{0,1\}^k$, she transmits $\varphi(x) \in C$. Bob receives $y \in \{0,1\}^n$ which is a corrupted version of $\varphi(x)$. Since $d_H(y, \varphi(x)) \leq pn$, Bob can recover $\varphi(x)$ as the string $z \in C$ that is closest to y and then find $x = \varphi^{-1}(z)$.

The basic idea (due to Claude Shannon) is to create a code $C \subset \{0,1\}^n$ of size $|C| = 2^k$ such that the Hamming distance between any two strings in C is greater than $2pn$. (The Hamming distance $d_H(u,v)$ is the number of coordinates i such that $u_i \neq v_i$.) Alice and Bob agree about the encoding: a bijection $\varphi : \{0,1\}^k \to C$. If Alice needs to send a message $x \in \{0,1\}^k$, she transmits $\varphi(x) \in C$. Bob receives $y \in \{0,1\}^n$ which is a corrupted version of $\varphi(x)$. Since $d_H(y, \varphi(x)) \leq pn$, Bob can recover $\varphi(x)$ as the string $z \in C$ that is closest to y and then find $x = \varphi^{-1}(z)$.



Error Correcting Codes II

For a code $C \subset \{0,1\}^n$, its rate is

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

and its (normalized) distance is

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

The distance of a code controls its power to overcome noise while its rate measures its efficiency in channel utilization.

Error Correcting Codes II

For a code $C \subset \{0,1\}^n$, its rate is

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

and its (normalized) distance is

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

The distance of a code controls its power to overcome noise while its rate measures its efficiency in channel utilization.

Error Correcting Codes II

For a code $C \subset \{0,1\}^n$, its rate is

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

and its (normalized) distance is

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

The distance of a code controls its power to overcome noise while its rate measures its efficiency in channel utilization.

Error Correcting Codes II

For a code $C \subset \{0,1\}^n$, its rate is

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

and its (normalized) distance is

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

The distance of a code controls its power to overcome noise while its rate measures its efficiency in channel utilization.

Problem

Is it possible to explicitly design arbitrarily large codes $\{C_k\}$ of size $|C_k| = 2^k$, with $r(C_k) \geq r_0$ and $\delta_(C_k) \geq \delta_0$ for some absolute constants $r_0, \delta_0 > 0$?

Magic Bipartite Graphs

We utilize a bipartite version of expanders.

A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:

(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;

(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.
A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:
(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;
(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.

A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:

(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;

(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.
A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:
(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;
(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.

A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:

(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;

(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.
A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:
(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;
(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Bipartite Graphs

We utilize a bipartite version of expanders.
A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:
(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;
(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

We utilize a bipartite version of expanders.
A bipartite graph $G = (L \cup R, E)$ is said to be (n, d)-magic if $|L| = n$, $|R| = 3n/4$, every left vertex has degree d, and the following two properties hold:
(1) for every $S \subset L$ with $|S| \leq \frac{n}{10d}$, the set $\Gamma(S)$ of all neighbors of S in R is of size at least $\frac{5d}{8}|S|$;
(2) for every $S \subset L$ with $\frac{n}{10d} < |S| \leq \frac{n}{2}$, the set $\Gamma(S)$ is of size at least $|S|$.

Observe that for every nonempty $S \subset L$ with $s = |S| \leq \frac{n}{10d}$, there is a vertex in R with exactly one neighbor in S. Indeed, there are exactly ds edges between S and $\Gamma(S)$. Since $|\Gamma(S)| \geq \frac{5ds}{8}$, the average number of neighbors in S that a vertex in $\Gamma(S)$ has is at most $ds : \frac{5ds}{8} = \frac{8}{5} < 2$. Since each vertex in $\Gamma(S)$ has at least one neighbor in S, some vertices must have exactly one neighbor in S.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0,1\}^n \mid Ax = 0\}$.

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0,1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$. Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log |C|}{n} \geq 1/4$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.
Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log |C|}{n} \geq 1/4$.
Now let $x \neq 0$ be an n-bit vector and $S = \{j \in L \mid x_j \neq 0\}$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.
Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log |C|}{n} \geq 1/4$.
Now let $x \neq 0$ be an n-bit vector and $S = \{j \in L \mid x_j \neq 0\}$. If $|S| \leq \frac{n}{10d}$, there is some $i \in R$ with exactly one neighbor in S.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0, 1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.
Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log |C|}{n} \geq 1/4$.
Now let $x \neq 0$ be an n-bit vector and $S = \{j \in L \mid x_j \neq 0\}$. If $|S| \leq \frac{n}{10d}$, there is some $i \in R$ with exactly one neighbor in S.
Then the i-th coordinate in Ax is 1 whence $Ax \neq 0$ and $x \notin C$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0,1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.
Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log|C|}{n} \geq 1/4$.
Now let $x \neq 0$ be an n-bit vector and $S = \{j \in L \mid x_j \neq 0\}$. If $|S| \leq \frac{n}{10d}$, there is some $i \in R$ with exactly one neighbor in S.
Then the i-th coordinate in Ax is 1 whence $Ax \neq 0$ and $x \notin C$.
Thus, $d_H(c, 0) > \frac{n}{10d}$ for every $c \in C \setminus \{0\}$ whence $d_H(c_1, c_2) = d_H(c_1 - c_2, 0) > \frac{n}{10d}$ for all distinct $c_1, c_2 \in C$.

Magic Graphs Provide Powerful Codes

Let $G = (L \cup R, E)$ be an (n, d)-magic graph. Define the $R \times L$ matrix $A = (a_{ij})$ by setting $a_{ij} = 1$ if $i \in R$ is adjacent to $j \in L$ and $a_{ij} = 0$ otherwise. Let $C = \{x \in \{0,1\}^n \mid Ax = 0\}$.

Theorem (Sipser and Spielman, "Expander codes", 1996)

The code C has rate at least $1/4$ and distance at least $1/10d$.

Proof. A has rank at most $|R| = 3n/4$ whence $\dim C \geq n/4$.
Thus, $|C| \geq 2^{n/4}$ and $r(C) = \frac{\log |C|}{n} \geq 1/4$.
Now let $x \neq 0$ be an n-bit vector and $S = \{j \in L \mid x_j \neq 0\}$. If $|S| \leq \frac{n}{10d}$, there is some $i \in R$ with exactly one neighbor in S.
Then the i-th coordinate in Ax is 1 whence $Ax \neq 0$ and $x \notin C$.
Thus, $d_H(c, 0) > \frac{n}{10d}$ for every $c \in C \setminus \{0\}$ whence
$d_H(c_1, c_2) = d_H(c_1 - c_2, 0) > \frac{n}{10d}$ for all distinct $c_1, c_2 \in C$.
One gets $\delta(C) = \frac{\min_{c_1 \neq c_2 \in C} d_H(c_1, c_2)}{n} > 1/10d$.

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

Low Density Parity Check Codes

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs.
It is not hard to prove by a probabilistic argument the following:

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

Theorem (Pinsker, “On the complexity of a concentrator”, 1973)

There is a constant n_0 such that for every $d \geq 32$ and $n \geq n_0$, there exists an (n, d)-magic graph.

Low Density Parity Check Codes

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

Theorem (Pinsker, "On the complexity of a concentrator", 1973)

There is a constant n_0 such that for every $d \geq 32$ and $n \geq n_0$, there exists an (n, d)-magic graph.

This however is not sufficient for practical applications—one needs explicit and efficient constructions for magic graphs.

Low Density Parity Check Codes

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

Theorem (Pinsker, "On the complexity of a concentrator", 1973)

There is a constant n_0 such that for every $d \geq 32$ and $n \geq n_0$, there exists an (n, d)-magic graph.

This however is not sufficient for practical applications—one needs explicit and efficient constructions for magic graphs. These have been found only recently, see M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders", Proc. 34th STOC, 659–668, 2002.

Low Density Parity Check Codes

The above construction is a special case of so-called Low Density Parity Check codes suggested by Gallager in the early 1960s. Its feasibility heavily depends on the existence of (n, d)-magic graphs. It is not hard to prove by a probabilistic argument the following:

Theorem (Pinsker, "On the complexity of a concentrator", 1973)

There is a constant n_0 such that for every $d \geq 32$ and $n \geq n_0$, there exists an (n, d)-magic graph.

This however is not sufficient for practical applications—one needs explicit and efficient constructions for magic graphs. These have been found only recently, see M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders", Proc. 34th STOC, 659–668, 2002.

The corresponding LDPC codes give simultaneously the best coding parameters as well as best algorithmic performance.

For general expanders, the situation is quite similar.

Constructing Expanders: The Problem

For general expanders, the situation is quite similar.
Recall the formal definition:

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size n_i increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) = \min_{|S|\leq\frac{n_i}{2}} \frac{|\partial S|}{|S|} \geq \varepsilon$ for all i.

Constructing Expanders: The Problem

For general expanders, the situation is quite similar.
Recall the formal definition:

Definition

A sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of size n_i increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) = \min_{|S|\leq\frac{n_i}{2}} \frac{|\partial S|}{|S|} \geq \varepsilon$ for all i.

The existence of expanders is easy to prove: in fact, it can be shown that if one chooses for each i a random d-regular graph G_i with i vertices, then the sequence $\{G_i\}_{i\in\mathbb{N}}$ will be an expander family almost surely.

Constructing Expanders: The Problem

For general expanders, the situation is quite similar.
Recall the formal definition:

Definition

A sequence of d-regular graphs $\{G_i\}_{i \in \mathbb{N}}$ of size n_i increasing with i is a Family of Expander Graphs if there exists $\varepsilon > 0$ such that $h(G_i) = \min_{|S| \leq \frac{n_i}{2}} \frac{|\partial S|}{|S|} \geq \varepsilon$ for all i.

The existence of expanders is easy to prove: in fact, it can be shown that if one chooses for each i a random d-regular graph G_i with i vertices, then the sequence $\{G_i\}_{i \in \mathbb{N}}$ will be an expander family almost surely.
However, it is not what we really need: the applications require explicit and efficient constructions for expander families.

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

Constructing Expanders: Margulis's Family

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y+1), y)$, $(x, y \pm x)$, and $(x, y \pm (x+1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

Constructing Expanders: Margulis's Family

The first explicitly constructed family of expander graphs is due to Grigory Margulis, "Explicit Construction of Concentrators", Prob. Per. Infor. 9(4):325–332, 1975 (in Russian). This is a family of 8-regular graphs M_n where n runs over the set of positive integers. The vertex set of M_n is $\mathbb{Z}_n \times \mathbb{Z}_n$; the neighbors of the vertex (x, y) are $(x \pm y, y)$, $(x \pm (y + 1), y)$, $(x, y \pm x)$, and $(x, y \pm (x + 1))$, where the arithmetics is modulo m. This is an example of a very explicit expander family.

Margulis had not provided any specific bound on $h(M_n)$; later it was shown that $h(M_n) \geq \frac{8-5\sqrt{2}}{2}$ (Gabber and Galil, 1981). Both Margulis's and Gabber–Galil's results are rather hard to prove and use some heavy mathematical machinery.

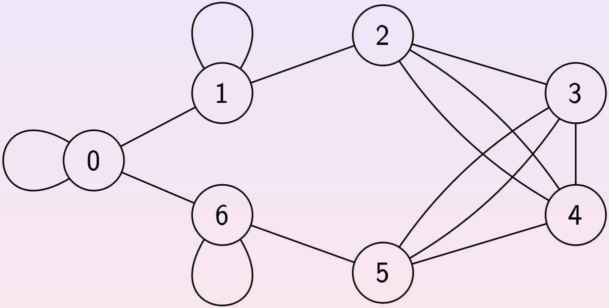
Another family consists of 3-regular graphs I_p indexed by primes.
The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetic is modulo p and $0^{-1} := 0$.
The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

Another family consists of 3-regular graphs I_p indexed by primes. The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetics is modulo p and $0^{-1} := 0$. The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

Another family consists of 3-regular graphs I_p indexed by primes. The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetics is modulo p and $0^{-1} := 0$. The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

Constructing Expanders: Another Family

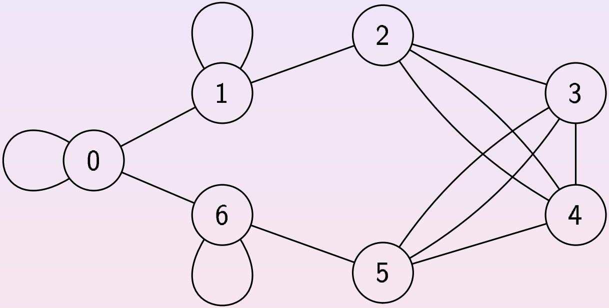
Another family consists of 3-regular graphs I_p indexed by primes. The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetics is modulo p and $0^{-1} := 0$.



The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

Constructing Expanders: Another Family

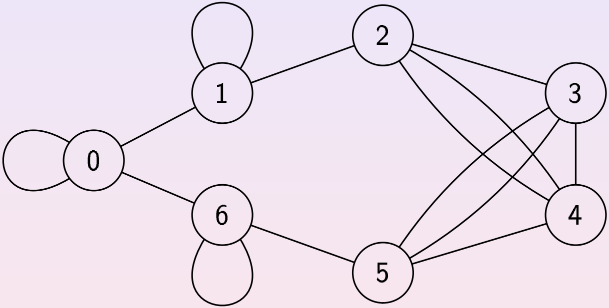
Another family consists of 3-regular graphs I_p indexed by primes. The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetics is modulo p and $0^{-1} := 0$.



The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

Constructing Expanders: Another Family

Another family consists of 3-regular graphs I_p indexed by primes. The vertex set of I_p is $\mathbb{Z}_p$; the neighbors of the vertex x are $x \pm 1$ and x^{-1}, where the arithmetics is modulo p and $0^{-1} := 0$.



The proof relies on a deep result in number theory (Selberg's 3/16 theorem). Observe that the family I_p is in a sense less explicit than Margulis's family since no deterministic polynomial algorithm for generating large primes is known.

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H,S)$ has H as the vertex set and a pair (g,h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H, S)$ has H as the vertex set and a pair (g, h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.

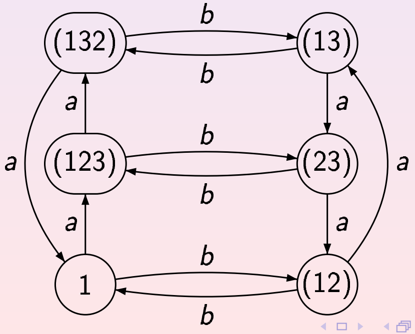
Algebra Comes Into The Play: Cayley Graphs

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H, S)$ has H as the vertex set and a pair (g, h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.

Algebra Comes Into The Play: Cayley Graphs

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H,S)$ has H as the vertex set and a pair (g,h) is an edge in the graph if and only if $gs=h$ for some $s \in S$.

Here is the Cayley graph of the symmetric group $\mathbb{S}_3$ with respect to its generating set $\{a=(123), b=(12)\}$:



It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H, S)$ has H as the vertex set and a pair (g, h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.
This definition yields a directed graph.

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H, S)$ has H as the vertex set and a pair (g, h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.
This definition yields a directed graph. However, we will assume that the set S is symmetric, i.e., $s \in S$ implies $s^{-1} \in S$. In this case $\mathrm{Cay}(H, S)$ is undirected and $|S|$-regular.

Algebra Comes Into The Play: Cayley Graphs

It turns out that the Cayley graphs of finite groups form a powerful source for explicit constructions of expanders. Let H be a finite group and let S be a generating set for H. The Cayley graph $\mathrm{Cay}(H, S)$ has H as the vertex set and a pair (g, h) is an edge in the graph if and only if $gs = h$ for some $s \in S$.
This definition yields a directed graph. However, we will assume that the set S is symmetric, i.e., $s \in S$ implies $s^{-1} \in S$. In this case $\mathrm{Cay}(H, S)$ is undirected and $|S|$-regular.

Problem

For which finite groups H and their generating set S do the Cayley graphs $\mathrm{Cay}(H, S)$ form a family of expander graphs?

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log|H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \ldots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \ldots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \ldots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log|H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log|H|$ elements.

Observation

Every finite group H has a generating set of size $\log|H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \ldots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \ldots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \ldots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log|H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log|H|$ elements.

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log|H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \ldots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \ldots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \ldots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log|H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log|H|$ elements.

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log|H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \ldots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \ldots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \ldots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log|H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log|H|$ elements.

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log|H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \dots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \dots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \dots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log|H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log|H|$ elements.

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log |H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \ldots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \ldots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \ldots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log |H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log |H|$ elements.

All Groups Have Small Generating Sets

Observation

Every finite group H has a generating set of size $\log |H|$.

Proof: It is a simple greedy algorithm: having picked i elements $g_1, g_2, \dots, g_i$ from H into the generating set, we list out the elements of the subgroup H_i generated by the set $\{g_1, g_2, \dots, g_i\}$. If $H_i \neq H$, we pick any $g_{i+1} \in H \setminus H_i$ as the next element in the generating set. The subgroup H_{i+1} generated by $\{g_1, \dots, g_i, g_{i+1}\}$ contains H_i properly whence $|H_{i+1}| \geq 2|H_i|$ by Lagrange's theorem. The process will stop after at most $\log |H|$ steps.

This bound is tight: the group $H = \mathbb{Z}_2^n$ has 2^n elements and dimension n as the vector space over $\mathbb{Z}_2$. Hence every generating set of H contains at least $n = \log |H|$ elements.

Alon–Roichman Theorem

Noga Alon and Yuval Roichman ("Random Cayley Graphs and Expanders", Random Struct. Algorithms, 5(2):271–285, 1994) have proved that every finite group has a choice of logarithmically many generators which yield an expanding Cayley graph.

Alon–Roichman Theorem

Noga Alon and Yuval Roichman ("Random Cayley Graphs and Expanders", Random Struct. Algorithms, 5(2):271–285, 1994) have proved that every finite group has a choice of logarithmically many generators which yield an expanding Cayley graph.

Theorem (Alon and Roichman)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. Let H be a group of order n, and let S be a random set of $c(\varepsilon) \log n$ elements of H, then the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$ almost surely. (The probability that G is such an expander tends to 1 as $n \to \infty$.)

Alon–Roichman Theorem

Noga Alon and Yuval Roichman ("Random Cayley Graphs and Expanders", Random Struct. Algorithms, 5(2):271–285, 1994) have proved that every finite group has a choice of logarithmically many generators which yield an expanding Cayley graph.

Theorem (Alon and Roichman)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. Let H be a group of order n, and let S be a random set of $c(\varepsilon)\log n$ elements of H, then the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$ almost surely. (The probability that G is such an expander tends to 1 as $n \to \infty$.)

Recently, it has been shown by Arvind, Mukhopadhyay, and Nimbhorkar ("Erdös-Rényi sequences and deterministic construction of expanding Cayley graphs", LATIN 2012: 37–48) that the Alon–Roichman theorem admits an efficient derandomization.

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon)\log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Alon–Roichman Theorem: Derandomization

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon)\log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Alon–Roichman Theorem: Derandomization

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon) \log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Alon–Roichman Theorem: Derandomization

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon) \log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Alon–Roichman Theorem: Derandomization

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon)\log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Alon–Roichman Theorem: Derandomization

Theorem (Derandomized version of the Alon–Roichman theorem)

For every ε such that $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that the following holds. There exists a polynomial in n algorithm that, given a group H of order n, produces a generating set S of H of size $c(\varepsilon) \log n$ elements of H such that the Cayley graph $G = \mathrm{Cay}(H, S)$ is an expander with $h(G) \geq \varepsilon$.

Thus, the algebraic approach can be used to produce explicit families of expanders.

What else should be done? If H is given in a more efficient way, we want the algorithm to work in time polynomial in the size of the description of H rather than the size of H itself. A more efficient way—as a group of permutations or matrices; if, say, H is specified as a subgroup of $\mathbb{S}_m$, the algorithm is to be polynomial in m, etc.

Conclusion

Modern computer science asks many questions that:

- are of major theoretical and practical importance;
- can be stated in term of very "classical" algebra;
- are somewhat different from the "standard" questions studied by algebraists and thus often require new algebraic tools.

This will keep busy many generations of algebraists!

Conclusion

Modern computer science asks many questions that:

- are of major theoretical and practical importance;
- can be stated in term of very "classical" algebra;
- are somewhat different from the "standard" questions studied by algebraists and thus often require new algebraic tools.

This will keep busy many generations of algebraists!

Conclusion

Modern computer science asks many questions that:

- are of major theoretical and practical importance;
- can be stated in term of very "classical" algebra;
- are somewhat different from the "standard" questions studied by algebraists and thus often require new algebraic tools.

This will keep busy many generations of algebraists!

Conclusion

Modern computer science asks many questions that:

- are of major theoretical and practical importance;
- can be stated in term of very “classical” algebra;
- are somewhat different from the “standard” questions studied by algebraists and thus often require new algebraic tools.

This will keep busy many generations of algebraists!

Conclusion

Modern computer science asks many questions that:

- are of major theoretical and practical importance;
- can be stated in term of very "classical" algebra;
- are somewhat different from the "standard" questions studied by algebraists and thus often require new algebraic tools.

This will keep busy many generations of algebraists!