The Hypergraph Moore Bound

Talk by Anqi Li
Notes by Sanjana Das
April 28, 2023

§1 Introduction

This talk is based on the paper *A simple and sharper proof of the hypergraph Moore bound.*

The question we'll consider is about the tradeoff between girth and density.

§1.1 Moore Bound for Graphs

Question 1.1. What is the maximum possible girth of a graph with n vertices and $nd/2$ edges (i.e., with average degree d)?

When $d = 2$, the answer is n — we can simply take an n-cycle. But when $d > 2$, there is a transition in some sense, and the answer looks quite different.

Theorem 1.2 (Alon, Hoory, Linial 2002)
For $d > 2$, the girth of a n-vertex graph with average degree d is at most $2\log_{d-1} n + 2$.

Remark 1.3. This bound is called the *Moore bound*. A *Moore graph* is a regular graph whose girth is greater than twice its diameter. A Moore graph with degree d, girth $2k+1$, and diameter k can have at most

$$1 + d\sum_{i=0}^{k-1}(d-1)^i$$

vertices (this can be proven by BFS), which approximately corresponds to the above bound.

This theorem is tight up to a constant factor, as seen by the following lower bound:

Theorem 1.4 (Marqulis 1988, Lubotsky–Phillips–Sarnak 1988)
For a fixed prime p, there is a sequence of graphs $G_i = (V_i, E_i)$ such that $|V_i| \to \infty$ and

$$g(G_i) \geq \frac{4}{3}\log_p |V_i| - O(1).$$

(Here $g(G)$ denotes the girth of G, and the average degree of the graphs is determined by p.)

§1.2 Moore Bound for Hypergraphs

We've seen that we have a girth-density tradeoff for usual graphs; we can then ask if the same is true for *hypergraphs*. First, it's not obvious what a *cycle* means in a hypergraph; we'll use the following definition.

Definition 1.5. An *even cover* in a hypergraph is a nonempty collection of hyperedges that hits each vertex an even number of times.

(An even cover may hit some vertices zero times.) Note that in the case of graphs, an even cover is a *union* of cycles, rather than just a cycle. But even covers still track the property we care about correctly — a graph contains no even cover of size ℓ if and only if it contains no cycle of size ℓ (i.e., if and only if it has girth greater than ℓ).

Definition 1.6. The *length* of an even cover is the number of hyperedges it contains.

Conjecture 1.7 (Feige's Conjecture) — There exists a constant c, depending on k but not ℓ, such that for any $\ell \geq 1$, every k-uniform hypergraph with

$$m \geq cn \left(\frac{n}{\ell}\right)^{k/2-1}$$

hyperedges has an even over of length at most $\ell \log_2 n$.

In the case $k = 2$, this is implied by the Moore bound — the second factor has no contribution as its exponent $\frac{k}{2} - 1$ is zero. Then if we set c large enough, then the assumption $m \geq cn$ forces the average degree of our graph to be at least 10, and so by the Moore bound for graphs we obtain that the girth is at most $2\log_{10} n + 2 < \log_2 n \leq \ell \log_2 n$.

In the case $k = 3$ (i.e., for 3-uniform hypergraphs), the bound in the conjecture has been proven for *random* 3-uniform hypergraphs. Meanwhile, it's also known that the bound is tight up to logarithmic factors. This follows from the following result.

Proposition 1.8

There exists a constant c such that we can find a 3-uniform hypergraph on n vertices and dn edges such that there is no subset of $g \leq cn/d^2$ vertices which induce at least $2g/3$ hyperedges.

(The proof is to take the *random* 3-uniform hypergraph with the correct hyperedge probability, and perform a moment calculation.)

To see why this provides a lower bound, note that an even cover which hits g vertices must contain at least $2g/3$ hyperedges — the cover must hit each of these g vertices at least twice, and each hyperedge only hits 3 vertices. So the condition that no $g \leq cn/d^2$ vertices induce at least $2g/3$ hyperedges means that there is no even cover consisting of at most cn/d^2 vertices, and therefore no even cover of length at most $2cn/3d^2$. Then taking $d \approx \sqrt{n/\ell}$ provides a hypergraph containing $n(n/\ell)^{1/2}$ hyperedges but no even cover of length ℓ (up to constant factors), showing that the bound in the conjecture is tight up to the $\log n$ term.

The following results are known. First, in the 3-uniform setting, Alon and Feige proved a bound with n^2/ℓ in place of $n^{1.5}/\ell$ (the ideal in the conjecture).

Theorem 1.9 (Alon, Feige 2009)

If a 3-uniform hypergraph has $m \gtrsim n^2/\ell$ hyperedges, then it contains an even cover of size at most $\ell \log_2 n$.

The bound in the conjecture has since been proven (for general k) up to logarithmic factors.

Theorem 1.10 (Guruswami, Kothari, Manohar 2021)
If a k-uniform hypergraph has $m \gtrsim n(n/\ell)^{k/2-1}(\log n)^{4k+1}$ hyperedges, then it contains an even cover of size at most $\ell \log_2 n$.

The result from this paper reduces the number of $\log n$ factors we need to just one.

Theorem 1.11 (Hsieh, Kothari, Mohanty 2022)
If a k-uniform hypergraph has $m \gtrsim n(n/\ell)^{k/2-1} \log n$ hyperedges, then it contains an even cover of size at most $\ell \log_2 n$.

We'll only consider the case where k is even (the k odd case is somewhat irritating). We'll see a template of the proof for *graphs*, and then see how to generalize to hypergraphs.

§2 Proof of Moore Bound for Graphs

We'll prove a slightly weaker version of the Moore bound — instead of getting $\log_{d-1} n$, we'll get $\log_{d/16} n$.

Proposition 2.1
If a graph has n vertices and $nd/2$ edges, then its girth is at most $2 \log_{d/16} n$.

Assume that our graph has no cycles of length at most ℓ (so we want to upper-bound ℓ). We'll use the trace method. But applying the trace method naively doesn't work because the vertices may have very uneven degrees; so we will reweight the adjacency matrix to fix this.

Lemma 2.2
Let G be a graph with average degree d and with no cycles of length at most ℓ, and let A be the adjacency matrix of G. Let D be the diagonal matrix whose diagonal entries are the degrees of each vertex, and let $\Gamma = D + dI$. Then

$$\left\|\Gamma^{-1/2} A \Gamma^{-1/2}\right\|_2 < \frac{2n^{1/\ell}}{\sqrt{d}}.$$

Proof. Let $\widetilde{A} = \Gamma^{-1/2} A \Gamma^{-1/2}$ be the relevant matrix. Then we have

$$\left\|\widetilde{A}\right\|_2^\ell \leq \operatorname{tr}(\widetilde{A}^\ell) = \operatorname{tr}((\Gamma^{-1}A)^\ell),$$

so it suffices to bound this trace. (The trace method is commonly used for random matrices; here $\tilde{A}$ isn't a random matrix, but the fact that it has no short cycles means that the method still works well.)

In order to find this trace, we consider closed walks in our graph G. (These walks will need to be reweighted because of the Γ^{-1} term; we'll deal with that later.) A walk $v_1 \to v_2 \to \cdots \to v_\ell \to v_1$ can't contain any cycles, because our graph doesn't have any cycles of length at most ℓ; this means it must be a backtracking walk (sort of like DFS on a tree), using $\ell/2$ 'new' edges and $\ell/2$ backtracking steps.

We'll encode such walks using ℓ-bit strings $b = \{0,1\}^\ell$ containing a 0 each time we pick a new edge and 1 each time we backtrack — so given a string b, to construct a corresponding walk, we first choose a starting

vertex, and on the ith step if $b_i = 0$ then we choose a new edge and if $b_i = 1$ then we backtrack. Using this, we can expand

$$\operatorname{tr}(\Gamma^{-1}A)^\ell = \sum_b \sum_{v_1\in[n]} \sum_{v_2\in N_{b_1}(v_1)} \cdots \sum_{v_{\ell+1}\in N_{b_\ell}(v_\ell)} (\Gamma^{-1})_{v_1v_1}(\Gamma^{-1})_{v_2v_2}\cdots(\Gamma^{-1})_{v_\ell v_\ell}\cdot \mathbf{1}_{v_{\ell+1}=v_1},$$

where $N_{b_i}(v_i)$ denotes the set of possible vertices we can go to from v_i given the bit b_i (if $b_i = 0$ then it's the set of new neighbors of v_i, and if $b_i = 1$ then it's the vertex from which we came to v_i that we need to backtrack to). (Note that Γ^{-1} is a diagonal matrix, so the terms $(\Gamma^{-1})_{v_iv_i}$ are the entries of $\Gamma^{-1}A$ in position (v_i, v_{i+1}), which are what we need to multiply to obtain the trace.)

Now for each i, the ith step contributes a factor of at most $|N_{b_i}(v_i)|\,(\Gamma^{-1})_{v_iv_i}$ to the sum (since we get $|N_{b_i}(v_i)|$ choices for the next vertex, and we multiply by a factor of $(\Gamma^{-1})_{v_iv_i}$). We defined Γ as the diagonal matrix with diagonal entries of $d + \deg(v_i)$. If $b_i = 0$ then we have $|N_{b_i}(v_i)| \le \deg(v_i)$, so this contribution is at most

$$\frac{\deg(v_i)}{d+\deg(v_i)} < 1.$$

Meanwhile if $b_i = 1$ then we have $|N_{b_i}(v_i)| = 1$ (as we must backtrack), so this contribution is at most

$$\frac{1}{d+\deg(v_i)} < \frac{1}{d}.$$

(The point of the multiplication of Γ^{-1} was to mask out the high-degree vertices in the way seen here — if a vertex v_i contributes a lot of choices for v_{i+1}, this is neutralized by the division by $(\Gamma^{-1})_{v_iv_i}$.)

For any fixed b, since $b_i = 0$ for at least $\ell/2$ of our steps, we have at least $\ell/2$ factors of less than $1/d$, and the remaining factors are less than 1; this means

$$\operatorname{tr}((\Gamma^{-1}A)^\ell) < 2^\ell n\left(\frac{1}{d}\right)^{\ell/2}$$

(the 2^ℓ comes from choosing b, and the n from choosing v_1). This gives precisely the bound we want. □

Proof of Proposition 2.1. The above lemma means we can sort of use the matrix Γ as a proxy for A — the above statement implies that

$$A \prec \frac{2n^{1/\ell}}{\sqrt{d}}\Gamma.$$

We'll use this to make use of the fact that our graph has $nd/2$ edges — $\mathbf{1}^\intercal A\mathbf{1}$ counts twice the number of edges in a graph, so

$$nd = \mathbf{1}^\intercal A\mathbf{1} < \frac{2n^{1/\ell}}{\sqrt{d}}\mathbf{1}^\intercal\Gamma\mathbf{1} = \frac{2n^{1/\ell}}{\sqrt{d}}\left(\sum_v(\deg(v)+d)\right) = \frac{2n^{1/\ell}(2nd)}{\sqrt{d}}.$$

This rearranges to

$$n^{1/\ell} > \frac{\sqrt{d}}{4} \implies \ell < 2\log_{d/16} n,$$

as desired. □

In order to generalize this method to hypergraphs, we need to find a matrix $A_{\mathcal{H}}$ associated to our hypergraph $\mathcal{H}$ such that $\mathbf{1}^\intercal A_{\mathcal{H}}\mathbf{1}$ again counts the number of hyperedges in the graph. Then we want to use the same trace moment method to show that some suitable reweighting of $A_{\mathcal{H}}$ has small L^2–norm — i.e., we want to show that for some matrix W, the quantity

$$\left\|W^{-k/2}A_{\mathcal{H}}W^{-k/2}\right\|_2$$

is small (here W serves the same function as Γ — reweighting to deal with the fact that some vertices might have high degree).

There are two difficulties in performing this generalization. First, we need to figure out what the matrix $A_{\mathcal{H}}$ should be — we want $A_{\mathcal{H}}$ to play a similar role to the adjacency matrix for a graph (in that it counts edges), but it's not clear what the correct matrix to use is. We also need to figure out how to use the condition that there's no covers (we used the lack of cycles to say that our walk must have been backtracking, and it's unclear how this generalizes to even covers).

§3 The Tensor PCX Problem

We'll now take a detour and discuss a different problem; these two problems look very different, but we'll later see a connection between them, and this connection can be used to motivate the construction of $A_{\mathcal{H}}$ we will eventually take.

In this problem, we will work with $(\mathbb{R}^n)^{\otimes m}$, the space of $n \times n \times \cdots \times n$ tensors (with m n's in the product) with real entries. (Think of n as much bigger than m.) We first consider a Gaussian 'noise tensor' $\widetilde{G} \in (\mathbb{R}^n)^{\otimes m}$, where each of the n^m entries is drawn (independently) from the normal distribution $\mathcal{N}(0,1)$. We then let $G \in (\mathbb{R}^n)^{\otimes m}$ be a symmetrized version of $\widetilde{G}$.

For some fixed λ, we then take a uniform random vector $X \in \{\pm 1\}^n$, and output $Y_\lambda = \lambda X^{\otimes m} + G$ — we can think of X as a planted signal, and G as noise that we use to corrupt this signal. (If $\lambda = 0$, then we just have noise.) We let $\mathbb{P}_\lambda$ denote the law for Y_λ.

> **Question 3.1.** Given an output $Y \in (\mathbb{R}^n)^{\otimes m}$, can we detect whether it came from $\mathbb{P}_0$ or $\mathbb{P}_\lambda$ (for a given value of λ)?

More precisely, we want to construct a function $f : (\mathbb{R}^n)^{\otimes m} \to \{0,1\}$ (where f takes Y as its input, and outputs 0 to claim that Y came from $\mathbb{P}_0$ and 1 to claim that Y came from $\mathbb{P}_\lambda$) such that f has *strong detection* for $\mathbb{P}_0$ and $\mathbb{P}_\lambda$, meaning that

$$\lim_{n\to\infty} \mathbb{P}_\lambda(f(Y) = 1) = \lim_{n\to\infty} \mathbb{P}_0(f(Y) = 0) = 1.$$

(To be more precise, f is a sequence of functions — we have one function for each value of n.)

> **Theorem 3.2**
> If $\lambda \gg \ell^{-(m-1)/4} n^{-m/4} \sqrt{\log n}$, then strong detection is possible in time $n^{O(\ell)}$ (where $\ell \le n$).

This theorem gives us a tradeoff between computation time and the threshold needed to detect the signal — with slower computation time, we can certify larger ℓ (meaning that we can detect a weaker signal). If we want a polynomial time algorithm, then the threshold for which λ we can perform detection is around $n^{-m/4}$; meanwhile if we just want *any* algorithm, taking $\ell = n$, the threshold becomes around $n^{-m/2}$. The general statement sort of interpolates between these two cases (we can do better and better detection by relaxing our time constraints).

§3.1 Method for Strong Detection

To see how this works, suppose that $Y = \lambda X^{\otimes m} + G$ (where X and G are chosen as above). We will consider $\mathbb{P}(X \mid Y)$ — we have

$$\mathbb{P}(X \mid Y) \propto \exp\left(-\frac{1}{2}\sum (Y_{i_1 \cdots i_m} - \lambda X_{i_1} \cdots X_{i_m})^2\right)$$

where the sum is over all $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ (note that Y is a $n \times n \times \cdots \times n$ tensor, so its entries are indexed by m-tuples of elements of $\{1, \ldots, n\}$, and therefore $Y_{i_1 \cdots i_m}$ is one of its entries; meanwhile X is a n-entry vector, so $X_{i_1} \cdots X_{i_m}$ is the product of m of its entries).

We can rewrite this as

$$\mathbb{P}(X \mid Y) \propto \exp\left(\lambda \sum_{|E|=m} Y_E X^E\right),$$

where the sum is over all subsets $E \subseteq \{1, \ldots, n\}$ of size m, and if $E = \{i_1, \ldots, i_m\}$, then we use Y_E to denote $Y_{i_1 \cdots i_m}$ and X^E to denote $X_{i_1} \cdots X_{i_m}$. (This is because in each of the terms in the previous expression, if we expand out the square, the term $Y^2_{i_1 \cdots i_m}$ is fixed (for fixed Y as X varies), and the term $(\lambda X_{i_1} \cdots X_{i_m})^2$ is also fixed because the entries of X are all ± 1 so square to 1, so up to proportionality the only term that matters is the cross term, which gives the above expression.)

We can rewrite this by taking E to be a symmetric difference $S \,\Delta\, T$ — this gives

$$\mathbb{P}(X \mid Y) \propto \exp\left(\frac{\lambda}{2^n} \sum_{|S \Delta T|=m} Y_{S \Delta T} X^S X^T\right),$$

where the sum is over all pairs of subsets $S, T \subseteq \{1, \ldots, n\}$ whose symmetric difference has size m. (The factor of 2^n is because every set E can be written as $S \,\Delta\, T$ in 2^n ways; note that if $E = S \,\Delta\, T$ then $X^E = X^S X^T$, since the entries of X are all ± 1.)

Now let M be the $2^n \times 2^n$ matrix indexed by pairs of subsets $S, T \subseteq \{1, \ldots, n\}$, whose entries are given by

$$M_{S,T} = \begin{cases} Y_{S \Delta T} & \text{if } |S \,\Delta\, T| = m \\ 0 & \text{otherwise.} \end{cases}$$

For any $x \in \{\pm 1\}^n$, consider the 2^m-dimensional vector u^x (whose entries are indexed by subsets $S \subseteq \{1, \ldots, n\}$, the same index set we're using for M) defined as

$$(u^x)_S = \prod_{i \in S} x_i.$$

Then for any $X \in \{\pm 1\}^n$ we have

$$\frac{1}{2^n}(u^X)^\intercal M u^X = \frac{1}{2^n} \sum_{S,T} X^S \cdot M_{S,T} \cdot X^T = \frac{1}{2^n} \sum_{|S \Delta T|=m} Y_{S \Delta T} X^S X^T = \sum_{|E|=m} Y_E X^E.$$

This is exactly the expression we had above when calculating $\mathbb{P}(X \mid Y)$. This means that heuristically speaking, the top eigenvector of M should correspond roughly to the maximum likelihood estimate (i.e., the vector $x \in \{\pm 1\}^n$ for which $\mathbb{P}(X = x \mid Y)$ is largest) — if u^* is the top eigenvector of M and x the maximum likelihood estimate (so u^* is a 2^n-dimensional vector, and x is n-dimensional), then we should have

$$(u^*)_S \approx \prod_{i \in I} x_i$$

for all S. (We won't prove this, but it does make some heuristic sense because u^* is the vector maximizing $u^\intercal M u$, and for u of the specific form $u_S = \prod_{i \in S} x_i$ for some $x \in \{\pm 1\}^n$, maximizing $u^\intercal M u$ is the same as maximizing $\mathbb{P}(X = x \mid Y)$ by the above calculations.)

This means we could use the matrix M to perform our strong detection, by calculating its top eigenvector. However, this is very inefficient, because M is huge (it is a $2^n \times 2^n$ matrix). So to actually obtain an efficient

algorithm, we truncate M — we take a parameter ℓ, and truncate M so that it only consists of the subsets of size ℓ. Explicitly, we define $\widetilde{M}$ to be the matrix indexed by pairs of size-ℓ subsets of $\{1, \dots, n\}$ defined as

$$\widetilde{M}_{S,T} = \begin{cases} Y_{S \Delta T} & \text{if } |S \,\Delta\, T| = m \\ 0 & \text{otherwise.} \end{cases}$$

It turns out that the top eigenvector of $\widetilde{M}$ approximates u^* (the top eigenvector of M) well, and therefore it also approximates our signal x well. (We won't prove this.) This can be used to obtain an $n^{O(\ell)}$ algorithm for strong detection (as the matrix has dimensions $\binom{n}{\ell} = n^{O(\ell)}$).

§3.2 Connection to the Hypergraph Moore Bound

It turns out that a construction similar to $\widetilde{M}$ can be used to obtain the matrix $A_{\mathcal{H}}$ we were looking for. Before we see the construction, we'll discuss the intuition behind why the above problem is similar to our original one (the tradeoff between small even covers and density).

Given a hypergraph, we can construct a certain boolean satisfiability problem — assign a variable x_r to each vertex $r \in \mathcal{H}$, and a Boolean b_C to each edge $C \in \mathcal{H}$. Then for every edge $C \in \mathcal{H}$, we create the equation $\bigoplus_{r \in C} x_r = b_C$ (so we want to find assignments of the x_i which satisfy this equation for all edges $C \in \mathcal{H}$).

To interpret small even covers in this framework, if the hyperedges $C_1, \dots, C_t$ form an even cover, then we have

$$\bigoplus_{j=1}^{t} b_{C_j} = \bigoplus_{j=1}^{t} \bigoplus_{r \in C_j} x_r = 0,$$

since every vertex appears an even number of times on the right-hand side. This means any system where $\bigoplus_{j=1}^{t} b_{C_j} = 1$ is not solvable — so an even cover is a sort of certificate saying that a certain system isn't feasible.

Suppose that instead of trying to fully solve the system, we're trying to satisfy as many clauses as possible. Then we can consider the function

$$\psi(x) = \sum_{C \in \mathcal{H}} b_C \oplus \bigoplus_{r \in C} x_r,$$

which records 1 for every clause not satisfied by the assignment x. We can rewrite this multiplicatively — letting $b' = (-1)^b$ for $b \in \{0, 1\}$, we obtain the function

$$\psi'(x) = \sum_{C \in \mathcal{H}} b'_C \prod_{r \in C} x'_r$$

(this records 1 for every clause satisfied, and -1 for every clause not satisfied).

We're trying to find x satisfying as many clauses as possible, which means we're trying to maximize $\psi'(x)$, and the existence of even covers gives an upper bound on how big $\psi'(x)$ can be. So even covers and solving binary systems are dual notions in some sense, which is why our original problem is related to this one.

But $\varphi'(x)$ is an expression of the same form as the one we saw when trying to maximize $\mathbb{P}(X = x \mid Y)$ in the tensor problem — so this can be done in the same way, where we write down the corresponding matrix and compute its top eigenvector. The value k here corresponds to m in the tensor problem, and the value ℓ here will correspond to the parameter ℓ in the tensor problem as well.

§4 Proof of the Moore Bound for Hypergraphs

Now we'll return to the original problem and prove the Moore bound for hypergraphs. (This proof doesn't depend on the tensor problem and can stand on its own, but the solution to the tensor problem is one way to motivate the construction of $A_{\mathcal{H}}$ we'll see here.)

We'll assume that k is even.

Definition 4.1. Given a k-uniform hypergraph $\mathcal{H}$ (with k even), its *Kikuchi graph* K_ℓ is the graph on vertex set $\binom{[n]}{\ell}$ with edges given by $S \sim T$ if $S \oplus T \in \mathcal{H}$.

(We use $\oplus$ to denote the symmetric difference of S and T — this has the same meaning as the notation $S \,\Delta\, T$ we used earlier.)

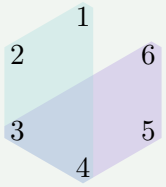
We can label the edges in the Kikuchi graph with the corresponding hyperedges in $\mathcal{H}$ — if $S \sim T$ with $S \oplus T = C$ (where C is a hyperedge in $\mathcal{H}$), then we write $S \overset{C}{\longleftrightarrow} T$.

Example 4.2

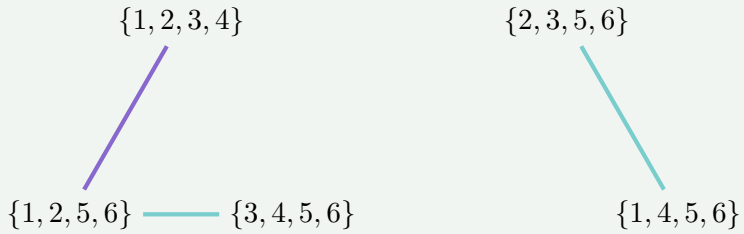
If $\mathcal{H}$ is a graph (i.e., $k = 2$) and $\ell = 1$, then the Kikuchi graph is simply the original graph.

Example 4.3

As an example with $k = 4$, let $\mathcal{H}$ have vertices $\{1, \dots, 6\}$ and hyperedges $\{1, 2, 3, 4\}$ and $\{3, 4, 5, 6\}$.



Then if $\ell = 4$, the vertices of the Kikuchi graph are 4-element subsets of $\{1, \dots, 6\}$, and we'll have the following edges (among others).



Similarly to the graph case, a lack of even covers in $\mathcal{H}$ corresponds to a lack of 'nontrivial' closed walks in the Kikuchi graph — in the graph case, a lack of small cycles meant that any short closed walk had to be backtracking. Here we have the following similar statement.

Lemma 4.4

Suppose that $\mathcal{H}$ has no even cover of size at most ℓ, and let K_r be the Kikuchi graph corresponding to $\mathcal{H}$ (for any r). Then for any closed walk

$$S_1 \overset{C_1}{\longleftrightarrow} S_2 \overset{C_2}{\longleftrightarrow} \cdots \overset{C_{\ell-1}}{\longleftrightarrow} S_\ell \overset{C_\ell}{\longleftrightarrow} S_1$$

in K_r, every hyperedge $C \in \mathcal{H}$ must appear (as an edge label) an even number of times.

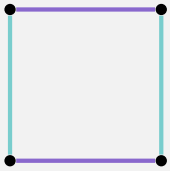
Proof. By the definition of our edges, we must have $S_1 \oplus S_2 = C_1$, $S_2 \oplus S_3 = C_2$, and so on. Taking the XOR of all these equations in the closed walk gives

$$C_1 \oplus C_2 \oplus \cdots \oplus C_\ell = (S_1 \oplus S_2) \oplus (S_2 \oplus S_3) \oplus \cdots \oplus (S_\ell \oplus S_1) = 0.$$

But since there is no even cover of size at most ℓ, this means each hyperedge on the left-hand side must cancel out (otherwise, once we cancel out all pairs of repeated terms, the remaining hyperedges would form an even cover), and therefore must appear an even number of times. □

Now we define $A_{\mathcal{H}}$ to be the adjacency matrix of the Kikuchi graph K_ℓ (the above argument is true for any r, but here we take r to be the same value ℓ). We then repeat the same trace moment argument that we used in the proof for graphs (here using the Kikuchi graph and $A_{\mathcal{H}}$). The above lemma again implies that any closed walk in the Kikuchi graph must be 'backtracking' (though the meaning of backtracking is a bit different than before); then everything works out similarly, and we obtain the correct bound.

Remark 4.5. Note that the Kikuchi graph *can* have small cycles, but those cycles must be trivial — for example, it's possible there may be a 4-cycle of the following form.



This will not be a problem for the argument, but it is why we need a slightly different notion of backtracking (here a backtracking step means we take an edge corresponding to a previously used *hyperedge*, rather than directly taking a previously used edge).