Thresholds for Latin squares and Steiner triple systems

Talk by Huy Pham

Notes by Sanjana Das

December 7, 2022

This is joint work with Vishesh Jain.

§1 Latin squares, Steiner triple systems, and edge colorings

Definition 1.1. A Latin square is a $n \times n$ array of numbers $x_{ij} \in [n]$ such that for each $y \in [n]$, every row and every column has exactly one instance of y.

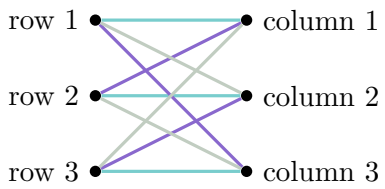
Example 1.2

Here is one example of a 3×3 Latin square:

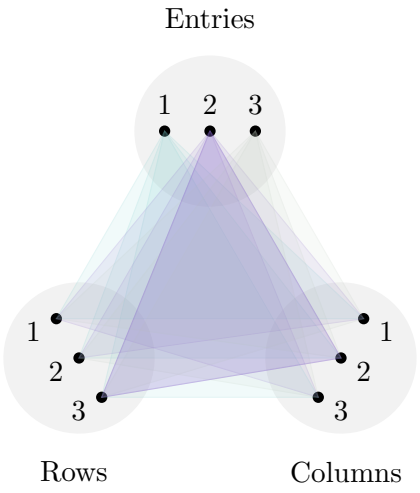
1	3	2
2	1	3
3	2	1

So a Latin square is a sort of 2-dimensional generalization of a permutation — it's a grid where every row and every column is a permutation of $[n]$.

Another way to think about a Latin square is as a (proper) edge-coloring of $K_{n,n}$ using n colors. Here, each row represents a vertex on the left side of the graph, and each column represents a vertex on the right; and the color of the edge connecting two vertices corresponds to the entry in that position. It's easy to check that the Latin square condition corresponds precisely to the condition that this edge-coloring is proper.



One can also think of a Latin square as a 3-partite 3-uniform hypergraph with the property that every pair of vertices in different parts is contained in exactly one hyperedge. For this correspondence, the three parts correspond to rows, columns, and entries, respectively; and we draw a hyperedge (i, j, y) if the entry in position (i, j) is y.



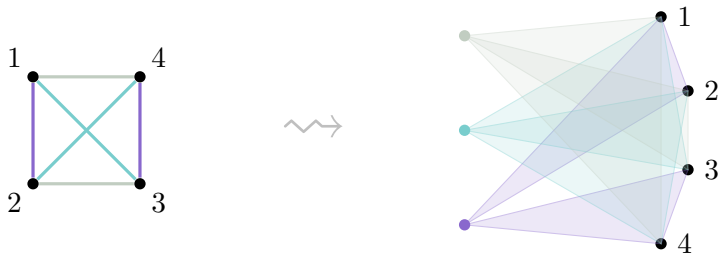
There's also many more variations of these viewpoints — for example, we can think of an edge-coloring of $K_{n,n}$ as a decomposition of $K_{n,n}$ into n perfect matchings, and so on.

The definition of Steiner triple systems is similar to the hypergraph viewpoint on Latin squares, but without tripartiteness.

Definition 1.3. A Steiner triple system is a 3-uniform hypergraph on n vertices where each pair of vertices is contained in exactly one hyperedge.

As another related object, we saw that Latin squares correspond to edge-colorings of $K_{n,n}$. We can also define hypergraphs associated with edge-colorings of *complete* graphs — suppose we have an edge-coloring of K_{2n} using $2n-1$ colors. Then we can draw a 3-uniform hypergraph, one of size $2n-1$ (representing colors) and one of size $2n$ (representing vertices), where edges (c,u,v) correspond to the color c being used on the edge uv. This leads to the following definition.

Definition 1.4. An edge coloring is a hypergraph with vertex parts of sizes $2n-1$ and $2n$, where each hyperedge connects two vertices on the right and one on the left, and every pair of vertices (with either both on the right, or one on the right and one on the left) is contained in exactly one hyperedge.



§2 Thresholds

We're interested in the following question:

Question 2.1. When does a random 3-uniform hypergraph contain one of these structures (a Latin square, Steiner triple system, or edge coloring) with reasonable probability?

We'll work with the 3-uniform Erdős–Rényi random hypergraph model.

Definition 2.2. The 3-uniform Erdős–Rényi random hypergraph, denoted by $\mathcal{G}^{(3)}(n,p)$, is the random n-vertex hypergraph where for each unordered triple of vertices, we place a hyperedge with probability p, independently.

Definition 2.3. Given a family $\mathcal{H}$ of 3-uniform hypergraphs, the threshold for $\mathcal{H}$, denoted by $p_c(\mathcal{H})$, is the value of p for which

$$\mathbb{P}[\mathcal{G}^{(3)}(n,p) \supseteq H \text{ for some } H \in \mathcal{H}] = \frac{1}{2}.$$

We'll use $\mathcal{H}_{\mathsf{LS}}$ to denote the collection of 3-uniform hypergraphs corresponding to Latin squares, $\mathcal{H}_{\mathsf{STS}}$ to denote the collection of hypergraphs corresponding to Steiner triple systems, and $\mathcal{H}_{\mathsf{EC}}$ to denote the collection of hypergraphs corresponding to edge colorings. (The number of vertices n is the same throughout — so we're looking for *spanning* substructures.)

Question 2.4. What is the order of magnitude of the threshold for $\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$?

§2.1 Expectation thresholds

Several recent works suggest a way to determine the threshold of a given family of hypergraphs, by looking at a related quantity called the *expectation threshold*.

Definition 2.5. The expectation threshold of a collection $\mathcal{H}$ of n-vertex hypergraphs, denoted by $p_e(\mathcal{H})$, is the largest value of p such that there exists a collection $\mathcal{H}'$ of n-vertex hypergraphs such that every $H \in \mathcal{H}$ contains at least one $H' \in \mathcal{H}'$, and such that

$$\sum_{H' \in \mathcal{H}'} p^{|E(H')|} \le \frac{1}{2}.$$

Proposition 2.6

We always have $p_e(\mathcal{H}) \le p_c(\mathcal{H})$.

Proof. The idea is to look at the expected number of $H' \in \mathcal{H}'$ that $\mathcal{G}^{(3)}(n,p)$ contains (where $\mathcal{H}'$ is as in the definition of the expectation threshold). For $p < p_e(\mathcal{H})$, this number is precisely

$$\sum_{H' \in \mathcal{H}'} \mathbb{P}[\mathcal{G}^{(3)}(n,p) \supseteq H'] = \sum_{H' \in \mathcal{H}'} p^{|E(H')|} < \frac{1}{2}.$$

Meanwhile, for $p \ge p_c(\mathcal{H})$, this number is at least $\frac{1}{2}$ — this is because $\mathcal{G}^{(3)}(n,p)$ contains some $H \in \mathcal{H}$ with probability at least $\frac{1}{2}$, and every $H \in \mathcal{H}$ contains some $H' \in \mathcal{H}'$. □

In many nice, symmetric situations, the expectation threshold turns out to be relatively easy to estimate, so we would like to know whether it's actually a *good* approximation for the actual threshold. This is true by the Kahn–Kalai conjecture, which was proved recently in joint work with Jinyoung Park.

Theorem 2.7 (Park–Pham)

For some universal constant C, we always have $p_c(\mathcal{H}) \le C \log n \cdot p_e(\mathcal{H})$.

It's not immediately clear how easy the expectation threshold is to estimate, though. It's easy to find a *lower* bound on the expectation threshold, because we can explicitly construct a family $\mathcal{H}'$ with the desired containment property and with small weight. But looking at the definition of the expectation threshold, it's unclear how we can get an *upper* bound.

But it turns out that there *is* often still an easy way to do this. We first take a fractional relaxation of the expectation threshold, and then use linear programming duality to certify an upper bound. We'll skip how this works, but it ends up giving the following.

Definition 2.8. A probability measure λ on $\mathcal{H}$ is called p-spread if for all $S \subseteq E(K_n^{(3)})$ (i.e., subsets of the edges of the complete 3-uniform hypergraph), we have

$$\lambda(\{H \in \mathcal{H} \mid H \supseteq S\}) \leq 2p^{|S|}.$$

Proposition 2.9

If there exists a p-spread measure on $\mathcal{H}$, then $p_e(\mathcal{H}) \leq p$.

Remark 2.10. Even before Theorem 2.7, there were earlier works (due to Alweiss–Lorett–Wu–Zhang, Frankston–Kahn–Narayanan–Park, and Mossel–Niles-Weed–Sun–Zadik) showing that to prove an upper bound on the threshold, it suffices to construct a p-spread measure — specifically, if there exists a p-spread measure on $\mathcal{H}$, then $p_c(\mathcal{H}) \leq Cp \log n$.

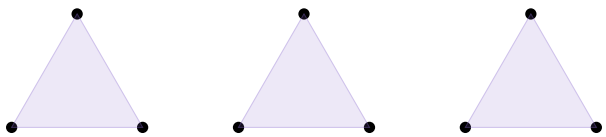
§2.2 Finding a p-spread measure

These results on thresholds hold for very general properties, but we'll be interested in applying them to our specific families of hypergraphs ($\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$). In this case, the main challenge will be finding a p-spread measure on our family of hypergraphs.

Before we get into the details, in many simple cases establishing such a spread measure can be easy.

Example 2.11

Consider $\mathcal{H}_{\mathsf{PM}}$, the family of 3-uniform perfect matchings on n vertices (a perfect matching is a collection of hyperedges such that every vertex is contained in exactly one hyperedge).



Here we can take λ to be the uniform measure on $\mathcal{H}_{\mathsf{PM}}$. It's not hard to verify that λ is $O(1/n^2)$-spread, and by the Kahn–Kalai conjecture, this implies that the threshold is $O(\log n/n^2)$. This turns out to be the correct order of the threshold — we can prove a matching lower bound by simple considerations (for smaller values of p, the hypergraph will typically have isolated vertices).

The main reason we can determine the spread here is that the family $\mathcal{H}_{\mathsf{PM}}$ is very nice and symmetric, and we can enumerate things very explicitly — the number of 3-uniform perfect matchings on n vertices is

$$\frac{n!}{3^{n/3}(n/3)!},$$

and we can similarly enumerate the number of perfect matchings containing a given set of edges S.

But with the families we're interested in — $\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$ — we can't enumerate things explicitly, so things become much harder. In particular, having a spread measure on $\mathcal{H}$ immediately implies a lower bound on the *number* of hypergraphs in $\mathcal{H}$ (taking $S = E(H)$ gives $\lambda(H) \le 2p^{|e(H)|}$ for all $H \in \mathcal{H}$; and in our settings $e(H)$ is fixed). And it's already hard to prove lower bounds on the sizes of the families we're interested in. A lower bound on the number of Latin squares can be obtained using bounds on the permanent (like Bregman's proof on the Minc conjecture, or a result resolving the van der Waerden conjecture). For Steiner triple systems, a lower bound was only obtained very recently by Peter Keevash, in his breakthrough on the existence of designs.

In general, if we have very good control on counts in certain induced families, then we can get spread properties by taking λ to be uniform. But for the families we're interested in, even proving lower bounds for counts in the *entire* 3-uniform complete hypergraph is difficult, and these bounds don't let us deal with more general induced hypergraphs. So even with the strongest results we know, it's not clear how to show spread by taking λ to be the uniform distribution.

So in our proofs, we won't take λ to be the uniform distribution; instead, we'll take a certain algorithmically constructed measure on our family of hypergraphs.

§3 History and results

First, what are the values for the thresholds that we *expect*?

> **Conjecture 3.1 —** The threshold for the family of Latin squares is $\Theta(\log n/n)$; the same is true for Steiner triple systems and edge colorings.

(This is due to several authors.)

It's easy to prove that $\log n/n$ is a *lower* bound (by considering when it becomes the case that every pair of vertices is in at least one hyperedge), so we want to prove it's an upper bound.

> **Theorem 3.2** (Sah–Sawhney–Simkin)
> For each of $\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$, we have $p_c(\mathcal{H}) = O(\exp((\log n)^{3/4})/n)$.

This gets the threshold up to a subexponential factor. It was proved by establishing a spread measure on these families of hypergraphs, using the absorption method and a boosting procedure.

> **Theorem 3.3** (Kang–Kelly–Kühn–Methuku–Osthus)
> For each of $\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$, we have $p_c(\mathcal{H}) = O((\log n)^2/n)$.

This was also proved by establishing a spread measure — the authors showed that there is an $O(\log n/n)$-spread measure on these families of hypergraphs.

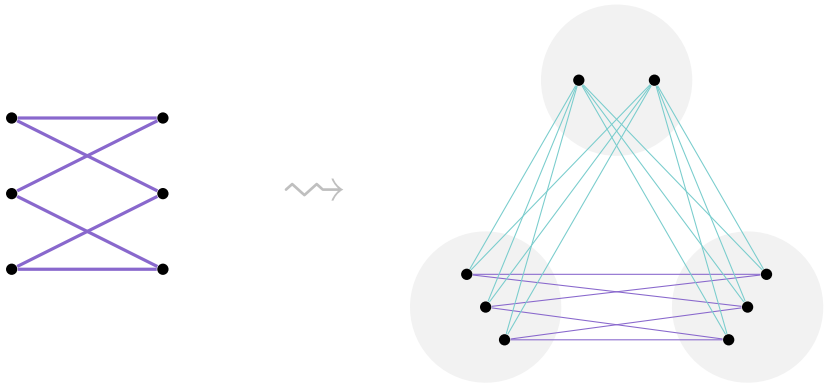
> **Theorem 3.4** (Jain–Pham)
> There is an $O(1/n)$-spread measure on each of $\mathcal{H}_{\mathsf{LS}}$, $\mathcal{H}_{\mathsf{STS}}$, and $\mathcal{H}_{\mathsf{EC}}$; so their thresholds are $O(\log n/n)$.

Since the matching lower bound is easy to obtain, this obtains the correct order of the threshold for these families.

In the remaining time, we'll sketch the ideas of the proof.

§4 Some setup

First, the KKKMO paper reduced the problem of constructing spread measures for the three families to a *single* problem: Suppose we have a regular bipartite graph G with n vertices on each side and common degree $d_G \geq (1-\delta)n$ (where δ is a small absolute constant). Then we can construct a 3-partite graph H_G by taking G and adding a third part of size d_G, and drawing all edges between the two original parts and this third part.



Then we consider the family $\mathcal{H}$ of 3-uniform hypergraphs H with the property that each edge of this 3-partite graph H_G is contained in exactly one hyperedge of H. And our goal is to construct a spread measure on $\mathcal{H}$.

Latin squares directly correspond to the case where H_G is the complete 3-partite graph where each part has size n; then this condition requires each pair of vertices in different parts to be contained in exactly one hyperedge. In this more general setting, we have two parts between which H_G is regular and *almost* complete, and a third part such that H_G is regular and complete between it and each of the other two. And it turns out that if we can get a spread measure in this slightly more general case, then we can get a spread measure for the other two types of objects (Steiner triple systems and edge colorings) as well.

An equivalent and slightly more intuitive way to think about members of $\mathcal{H}$ is as decompositions of G (the original bipartite graph) into d_G perfect matchings $R_1, R_2, \ldots, R_{d_G}$. (To see the equivalence between these, we think of vertices in the third part as indices of the perfect matchings — so a hyperedge (x, y, i) corresponds to placing the edge xy in the ith perfect matching R_i.)

It's an easy result that a d_G-regular graph always has at least *one* way to decompose its edges into d_G perfect matchings. We're interested in not just finding one way to do this, but a *distribution* over all of them that satisfies the spread property — i.e., that for all $S_1, \ldots, S_{d_G} \subseteq E(G)$, we have

$$\mathbb{P}[S_i \subseteq R_i \text{ for all } i] \leq 2p^{\sum_i |S_i|}. \tag{4.1}$$

And our goal is to get this with $p = O(1/n)$.

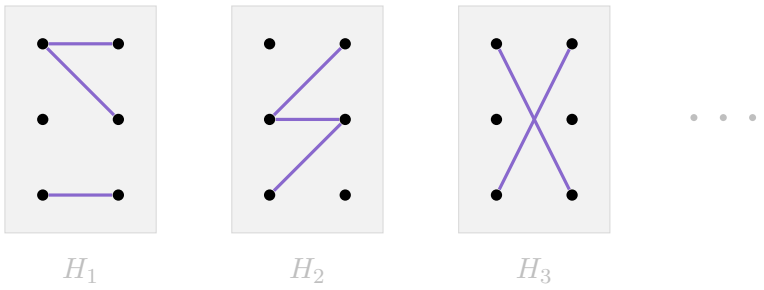
For the rest of the talk, we'll assume we're working with Latin squares (so $G = K_{n,n}$); this simplifies the setting a bit but still illustrates all the ideas in the argument.

§5 Proof ideas for Theorem 3.3

This is the setting where the work of KKKMO takes place. Their idea is that instead of directly decomposing G into perfect matchings, we first decompose it into regular graphs; we want to make their degrees as small as possible and to do this in a sufficiently random way that we get good spread. Specifically, we decompose G into d-regular graphs with $R_1, \ldots, R_t$ where $d = \Theta(\log n)$ and $t = n/d = \Theta(n/\log n)$ (we'll see the reason for

these parameters later), in such a way that (4.1) holds with $p = \Theta(\log n/n)$. This gives a $O(\log n/n)$-spread measure (we can decompose each R_i into d perfect matchings arbitrarily), which gets Theorem 3.3.

To get this decomposition, we start with G and randomly partition it into t graphs $H_1, \dots, H_t$, where for each edge in G, we assign it a uniformly random index between 1 and t (independently). If $d = \Theta(\log n)$, then each H_i is *close* to regular (with high probability).



The idea is that the partition into these graphs H_i is as random as we could wish, because every edge was assigned completely at random; so it's easy to see that the H_i satisfy the spread condition that we want. However, the problem is that they're not regular; they're only *close* to regular. So our goal is to fix these almost-regular graphs into actually regular graphs.

The approach of KKKMO is to do this dyadically. We first look at the last $t/2$ graphs H_i. These graphs are all very close to regular, so we can isolate a very small part of the edges of each one and throw it away, such that the remaining graph becomes regular.

Now we need to deal with these leftover edges; so we take the next $t/4$ graphs H_i (from the right), and we assign our leftover edges randomly to one of these $t/4$ graphs.

So in this next dyadic block, we have $t/4$ random graphs H_i from the original sampling, and each of them is absorbing some piece of the leftovers from the first dyadic block. So we take each of these H_i together with the leftovers assigned to it, and find a regular graph that contains most of the edges of H_i as well as the assigned leftovers. This gives us more leftover edges, which we randomly assign to the next dyadic block (the next $t/8$ graphs H_i), and so on.

It seems magical that this just works out; but the feature where we fix the graphs in a block from almost regular to regular makes things a bit nicer as we continue, which is how we can push this argument all the way through to end up with a collection of regular graphs.

Then the graphs R_i we obtain are almost the same as the random graphs H_i from the original partition, so the spread property is still retained. Finally, this gives a decomposition of G into d-regular graphs; if we take an arbitrary partition of each into perfect matchings, we get a decomposition into perfect matchings with the same spread parameter.

§6 Proof ideas for Theorem 3.4

The only thing that forces us to stop at $\log n$ in the KKKMO argument (i.e., to take $d = \Theta(\log n)$ instead of d constant) is that for this to work, we really do need all the random graphs H_i to be almost regular, so that this kind of fixing procedure can go through. If we want a smaller average degree, then we won't get such good control over concentration, so we'll have to do things a bit differently.

We'll have an outer iteration over an index r. For each r, we define $d_r \approx n/16^r$. Then in this outer iteration, we find a distribution $\mathbb{P}_r$ over decompositions $\mathcal{P}_r$ of the original graph G into regular graphs $R_1, \ldots, R_t$, each of degree $d_{R_i} = (1 + o(1))d_r$. (We stop the iteration once the degree hits a large constant.)

Inside, we run an inner iteration where we use $\mathbb{P}_{r-1}$ to obtain $\mathbb{P}_r$ — this means we take a decomposition $\mathcal{P}_{r-1}$ of the original graph G into regular subgraphs of degree roughly d_{r-1}, and we want to decompose each of its pieces into regular subgraphs of degree roughly d_r. From now on, we'll use G to denote the piece of $\mathcal{P}_{r-1}$ that we're trying to partition (instead of the original graph).

We'll first partition G into random graphs H_i, which we again think of as coming in dyadic blocks. On top of this, inside each H_i, we sample a small fraction of its edges, which we call H_i^+.

In the first dyadic interval (the last $t/2$ graphs H_i), we try to find a regular subgraph of each H_i such that the leftover is contained in H_i^+. Then we take all these leftover edges and random assign them to the graphs H_j in the previous dyadic interval (the previous $t/4$ graphs). For each of those graphs, we try to do the same thing — we take the original graph H_j together with the leftover edges it got assigned, and find a regular subgraph such that the leftovers are contained in H_j^+. And we again iterate (assigning these leftovers randomly to the next dyadic block, and so on).

The main caveat is that if we choose the original partition into graphs H_i uniformly at random, then we won't have good concentration of degrees — we won't be able to say that these graphs are close to regular. So in fact, we won't use a completely random assignment of edges to graphs H_i.

The idea is that if we look at the degree of a vertex v in some H_i (under a uniform random assignment of edges), it will have *somewhat* good concentration — for example, we have

$$\mathbb{P}\left[|d_{H_i}(v) - \mathbb{E}[d_{H_i}(v)]| \geq \sqrt{\mathbb{E}[d_{H_i}(v)] \log d_G}\right] \leq d_G^{-9}.$$

The only issue is that when d_G is small, we can't do a union bound (over all vertices).

But this suggests that we can use the Lovász local lemma — the Lovász local lemma tells us that there *exists* an assignment of edges to the graphs H_i such that the degree of each vertex in each H_i is close to its expected value. This is because the degree of a vertex only depends on where we assign the edges incident to it. So we have independent random variables for each edge (corresponding to which graph we assign that edge to), and a dependency graph where each bad event has degree roughly d_G (since an event corresponding to v only depends on the events corresponding to neighbors of v in G). And the upper bound on the probabilities of bad events is certainly good enough that we can apply the Lovász local lemma.

So instead of taking a uniformly random assignment (of edges to graphs H_i), we'll actually sample from the local lemma distribution — we'll take a uniformly random *satisfying* assignment (i.e., one that avoids all the bad events).

In the end, we still need to control the spread, and it seems we're losing a lot by switching from a uniform assignment to a uniform satisfying one (the probability a random assignment is satisfying could be exponentially small, so it seems we could lose an exponential factor in the relevant probability for spread). To get this control on the spread, we use the following variant on the local lemma, which states that for events B that don't have too many dependencies, the probabilities of B under a uniform assignment and uniform satisfying assignment are not too different.

Lemma 6.1

Suppose we have bad events E_i and real numbers $x_i \in (0,1)$ such that $\mathbb{P}[E_i] \le x_i \prod_{j\sim i}(1-x_j)$ for all i. Then for every event B, we have

$$\mathbb{P}_{\mathsf{sat}}[B] \le \frac{\mathbb{P}[B]}{\prod_{j\sim B}(1-x_j)}.$$

(Here $j \sim i$ and $j \sim B$ denote dependencies (i.e., adjacency in the dependency graph), and $\mathbb{P}$ and $\mathbb{P}_{\mathsf{sat}}$ denote the probability distributions over a uniform assignment and uniform satisfying assignment, respectively.)

This has a very cute proof, so we'll finish the talk by giving this proof.

Proof. We can prove this by closely following the proof of the local lemma. In that proof, we inductively show that $\mathbb{P}[E_i \mid \bigcap_{j\in S} \overline{E_j}] \le x_i$ for all i and S, which implies that

$$\mathbb{P}\Big[\bigcap_{j\in S_1} \overline{E_j} \;\Big|\; \bigcap_{j\in S_2} \overline{E_j}\Big] \ge \prod_{j\in S_1}(1-x_j) \tag{6.1}$$

by iterating Bayes' rule (for any S_1 and S_2).

And we can use this to prove the statement we want — we'll show that for all S, we have

$$\mathbb{P}\Big[B \;\Big|\; \bigcap_{j\in S} \overline{E_j}\Big] \le \frac{\mathbb{P}[B]}{\prod_{j\in S, j\sim B}(1-x_j)}.$$

(The desired statement corresponds to taking S to be the entire index set — taking a uniform satisfying assignment is equivalent to conditioning on the intersection of all $\overline{E_j}$.)

For this, similarly to the proof of the original local lemma, we can split the conditioning into events that are adjacent and non-adjacent to B, and write

$$\mathbb{P}\Big[B \;\Big|\; \bigcap_{j\in S} \overline{E_j}\Big] = \frac{\mathbb{P}[B \cap \bigcap_{j\in S, j\sim B} \overline{E_j} \mid \bigcap_{j\in S, j\not\sim B} \overline{E_j}]}{\mathbb{P}[\bigcap_{j\in S, j\sim B} \overline{E_j} \mid \bigcap_{j\in S, j\not\sim B} \overline{E_j}]}.$$

For the numerator, we can first drop the intersection with $\bigcap_{j\in S, j\sim B} \overline{E_j}$ (this only decreases the probability); then the conditioning has no effect (because we're only conditioning on events $j \not\sim B$, which are independent

of B); so the numerator is at most $\mathbb{P}[B]$. Meanwhile, the denominator is at least $\prod_{j\in S, j\sim B}(1-x_j)$ by (6.1); putting these together gives the bound we want. □

Then we need some technical things to show that we can do the dyadic fixing procedure; this requires some control over the distribution of edges in the pieces. For certain large sets, this control can be obtained inductively; the main problem is in dealing with the distribution of edges between small sets (we need small sets to have quite significant expansion). For this, the inductive approach doesn't work; we instead have to prove this by bootstrapping on the spread of the outer iteration (i.e., we use that $\mathbb{P}_{r-1}$ has good spread).