Hypergraph Ramsey

Talk by Xiaoyu He
Notes by Sanjana Das
September 22, 2023

§1 Introduction

The fundamental question we'll explore today is the growth rate of *hypergraph Ramsey numbers*.

Definition 1.1. A *k-uniform hypergraph* (or a *k-graph*) is an ordered pair $\mathcal{H} = (V, E)$ where $E \subseteq \binom{V}{k}$ — in other words, a k-graph consists of vertices and edges such that each edge is a k-tuple of vertices.

Definition 1.2. Given k-graphs $\mathcal{H}_1, \ldots, \mathcal{H}_r$, their *Ramsey number* $R(\mathcal{H}_1, \ldots, \mathcal{H}_r)$ is defined as the smallest n such that any r-coloring of the edges of a complete k-graph on n vertices contains a copy of $\mathcal{H}_i$ in color i for some i.

In other words, we have a giant complete graph (with n vertices), and no matter how we color its edges, we should be able to find one monochromatic object. (Of course we can't expect to find *more* than one, because the k-graph could be colored with just one color.)

Theorem 1.3 (Ramsey 1930)
For any hypergraphs $\mathcal{H}_1, \ldots, \mathcal{H}_r$, the Ramsey number $R(\mathcal{H}_1, \ldots, \mathcal{H}_r)$ is finite.

We can think of Ramsey's theorem as 'higher-order pigeonhole' — the pigeonhole principle states that if we have enough things and we color our things with finitely many colors, then we can find lots of things of the same color. Ramsey's theorem is a generalization where instead of coloring the things themselves, we color *k-wise relations* between the things (and we want to find lots of things such that all k-wise relations between them have the same color) — the pigeonhole principle is the case $k = 1$.

Example 1.4
Here are some examples of applications of Ramsey's theorem:

- Any long sequence of real numbers has a monotone subsequence of length 1000. (This comes from applying Ramsey's theorem with uniformity 2.)
- Any large matrix over $\mathbb{R}$ has a 1000×1000 submatrix where all 10×10 ranks are the same. (This might need a bipartite version.)
- Given any large graph, we can find 1000 linear-sized sets such that all pairs of these sets are ε-regular and have density within ε of each other.

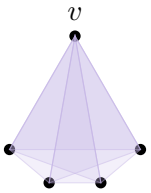
So far we've only talked about Ramsey's theorem *qualitatively*, but we're really interested in how these Ramsey numbers grow. We'll define a few natural hypergraphs to focus on.

Definition 1.5. We use $K_t^{(k)}$ to denote the *complete k-graph* on t vertices, which has all $\binom{t}{k}$ edges.

Definition 1.6. We use $K_{t,\dots,t}^{(k)}$ to denote the *complete k-partite k-graph* with t vertices in each part — in other words, the t-blowup of a single k-edge.

Definition 1.7. We use $S_t^{(3)}$ to denote the *star* with $t+1$ vertices and $\binom{t}{2}$ edges — the 3-graph with vertices $\{v, w_1, \dots, w_t\}$ and edges vw_iw_j.

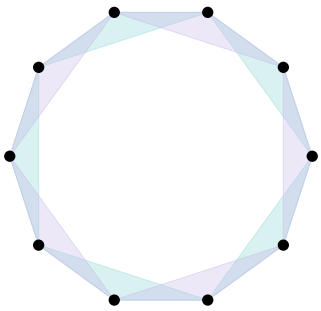
We can think of $S_t^{(3)}$ as a cone over the complete 2-graph K_t.



For most of this talk, we'll focus on uniformity 3; but several things can be generalized to higher uniformities.

There's several generalizations of cycles to hypergraphs; the one we'll use is *tight cycles*.

Definition 1.8. The *tight cycle* $C_n^{(k)}$ is defined as the k-graph we get by writing n points on a circle and taking our edges to be all consecutive k-tuples.



Definition 1.9. A *linear hypergraph* is a hypergraph $\mathcal{H}$ such that $|e_1 \cap e_2| \leq 1$ for all edges $e_1, e_2 \in E(\mathcal{H})$.

We'll mostly focus on the Ramsey numbers of complete hypergraphs of uniformity 3, but we'll also look at some of these other hypergraphs and see how they relate.

§1.1 Some techniques

First, here's a brief overview of some of the techniques involved in bounding Ramsey numbers.

For lower bounds, techniques involve random colorings, induced colorings (where we color an underlying *graph* randomly, and then color our hypergraph in a way induced by the graph coloring), and *stepping up* (which lets you lift lower uniformity hypergraphs to higher uniformity).

For upper bounds, for *graphs* there's the Erdős–Szekeres greedy embedding algorithm. For hypergraphs, there's a few different ways to try doing a greedy embedding — for example, in uniformity 3, you can embed by vertex (embedding one vertex at a time), by pair (embedding one element of the 2-shadow at a time), or by triples (adding a single edge at a time). It turns out that embedding one vertex at a time is the worst, but the other two methods both give good bounds in different regimes.

§1.2 Known results

Definition 1.10. We use $r_k(s,t)$ to denote the k-uniform Ramsey number for a s-clique and a t-clique.

We'll mostly focus on $r_3(t,t)$.

Theorem 1.11
We have $2^{\Omega(t^2)} \le r_3(t,t) \le 2^{2^{O(t)}}$.

The upper and lower bounds in this theorem have a gap in tower heights, which is somewhat unsettling. But the good news is that if we figured out this gap, we'd be able to lift both bounds to all higher uniformities as well, using the following theorem.

Theorem 1.12
For all $k \ge 4$, we have

$$2^{r_{k-1}(\varepsilon t, \varepsilon t)} \le r_k(t,t) \le 2^{r_{k-1}(t,t)^k}.$$

(The bounds here may suppress constant factors in the exponents.)

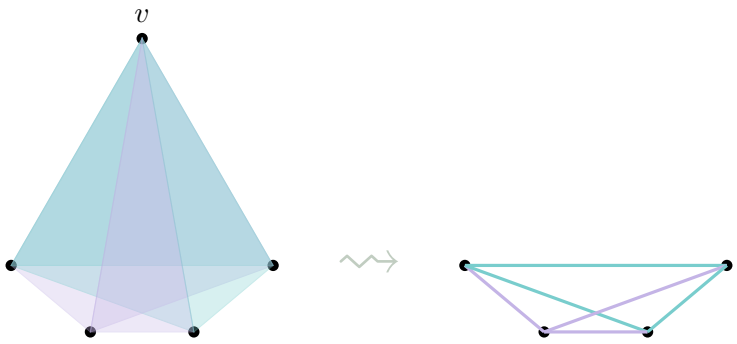
This means that starting from uniformity $k = 4$, when we go from uniformity $k-1$ to k, both our upper and lower bounds increase in tower height by 1. So if we could settle the tower height for $k = 3$, then we could settle it for *all* uniformities.

§2 Upper bounds

We'll now talk in more detail about proving upper bounds (focusing on uniformity 3).

§2.1 Embedding vertices

The first technique we can try is greedily embedding *vertices*. Just like in the Erdős–Szekeres proof for $k = 2$, we start by picking a vertex v. In Erdős–Szekeres we looked at a monochromatic neighborhood of v. Here we'll look at the *link* of v — the ordinary *graph* on the remaining vertices where we color each edge uw with the color of vuw in our original 3-graph. (This is an analog of the neighborhood for higher uniformities.)



Then by graph Ramsey, if the original 3-graph had N vertices, then in this link there exists a monochromatic clique of size at least $\frac{1}{2}\log N$. We'll then consider only the 3-graph formed by these vertices (and attempt

to find either a (3-uniform) $(t-1)$-clique of the same color — to which we can add v to get a t-clique — or a t-clique of the opposite color).

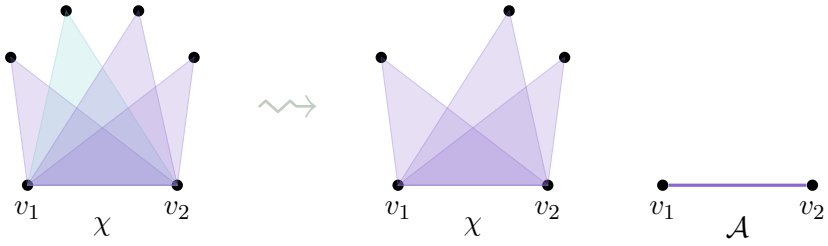
But the problem is that we lose a log at each step — we started with N vertices and ended up with $\log N$ after embedding just one vertex, and each additional vertex we embed introduces another log. This means we get a bound of roughly

$$r_3(t,t) \le 4^{4^{4^{\cdots}}},$$

with a tower height of t. (We can replace 4 with 3.999 now, but this is still a pretty bad bound.)

§2.2 Embedding pairs

Instead of embedding vertices as above, we'll embed *pairs* of vertices, using an auxiliary structure $\mathcal{A}$ (which will be a graph). We start by pulling out two vertices v_1 and v_2 of our hypergraph, and we look at all edges v_1v_2u containing this pair. At least half of these edges will have the same color; let's say this majority color is *red*. Then we'll color the edge v_1v_2 in $\mathcal{A}$ red, and we'll throw away all the vertices u for which the edge v_1v_2u was blue. So by losing a factor of 2 in our candidate set of vertices, we've drawn one edge in $\mathcal{A}$.



Then we pull out a new vertex v_3, and draw the edges v_1v_3 and v_2v_3 in $\mathcal{A}$. For the edge v_1v_3, we check whether the edges v_1v_3u in our hypergraph coloring χ are more commonly red or blue (over all remaining vertices u — i.e., vertices other than v_2); we color the edge in $\mathcal{A}$ with this majority color, and throw out all vertices u with the less common color. And then we do the same for the edge v_2v_3, and so on. (We're essentially doing an analog of Erdős–Szekeres where instead of embedding single vertices, we're embedding a 2-shadow.) Every time we draw an edge in $\mathcal{A}$, we're losing a factor of 2.

Eventually, if we started out with enough vertices, then in the end our auxiliary structure $\mathcal{A}$ is a complete graph colored with two colors, with the property that for all vertices $u < v < w$ (where we move vertices into $\mathcal{A}$ in order), the color of uvw in χ is the same as the color of uv in $\mathcal{A}$. (This is because when we move over v and draw the edge uv in $\mathcal{A}$, we throw away all vertices w still in our candidate set for which uvw has the wrong color, so we can't add such a vertex to $\mathcal{A}$ later.)

This means a monochromatic *graph* clique in $\mathcal{A}$ gives a monochromatic *hypergraph* clique in χ — if we can find $t-1$ vertices $v_1, \ldots, v_{t-1}$ such that every edge v_iv_j between them in $\mathcal{A}$ is red, then we can choose any vertex w that comes after them, and $v_1, \ldots, v_{t-1}, w$ form a red t-clique in our 3-graph.

So it's enough to ensure that $\mathcal{A}$ has $r_2(t,t)$ vertices (then we can find a monochromatic t-clique in $\mathcal{A}$ and win). And since we pay a factor of 2 for every *edge* we build in $\mathcal{A}$, we end up getting a bound of

$$r_3(t,t) \le 2^{\binom{r_2(t,t)}{2}}.$$

This is the same method used to prove the general upper bound of $r_k(t,t) \le 2^{r_{k-1}(t,t)^k}$ (the exponent may be $k-1$ in place of k) — we do the same proof using the $(k-1)$-shadow (so $\mathcal{A}$ is a $(k-1)$-uniform hypergraph).

This proof gives us $r_3(t,t) \le 2^{16^t}$; in fact, the constant 16 has been improved.

Theorem 2.1 (Conlon–Fox–Sudakov)
We have $r_3(t,t) \leq 2^{4^t}$.

(We may be dropping lower-order terms in this statement.)

The idea for the improvement is that we don't actually have to build *every* edge in our auxiliary graph $\mathcal{A}$ — instead we can choose only certain edges to build, making our choices adaptively based on what colors we've seen so far. This turns into the *vertex online Ramsey game* — a game where we build edges one at a time and an adversary colors each red or blue, and we want to find a monochromatic clique. Here we have essentially the same situation, except that order matters — when we introduce a new vertex v we can *only* build edges in $\mathcal{A}$ out of that vertex, and once we leave this vertex and add a new one, we can never come back and fill in the edges at v that we left out. So we have less adaptiveness here than in the vertex online Ramsey game, but as far as we know, the behavior is the same — so in some sense, we've reduced upper bounds for 3-uniform hypergraphs to bounds on a certain kind of graph Ramsey number.

Here's a formulation of how we run this algorithm (i.e., choose which edges to look at).

Algorithm 2.2 — Every time we introduce a new vertex into $\mathcal{A}$, we label it with a string, in the following way. We start off by labelling v_1 with $\emptyset$. When we introduce the ith vertex v_i:

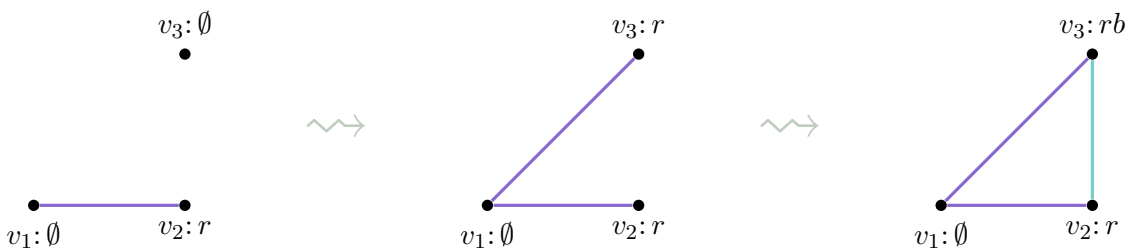
- First, label v_i with $\emptyset$.
- If the label of v_i collides with the label of one of the previous vertices (i.e., a vertex v_j with $1 \leq j \leq i-1$), then we build the edge v_iv_j and ask the adversary its color; we append this color to the current label of v_i.
- We repeat the above step until the label of v_i doesn't collide with any of the previous labels.

(For our purposes, asking the 'adversary' for the color of v_iv_j corresponds to considering the majority color of v_iv_ju over the remaining candidate vertices u, as above.)

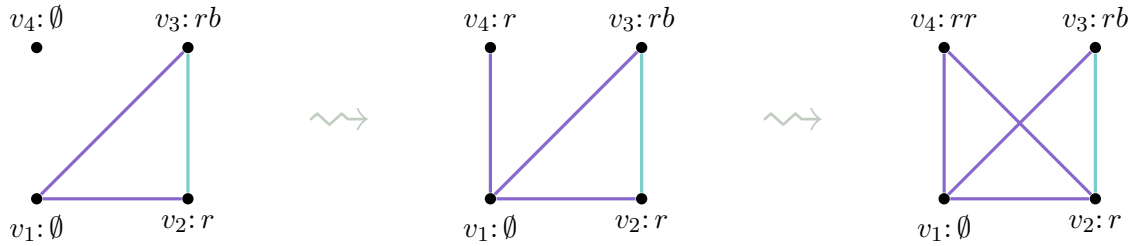
So we start by introducing v_1 with label $\emptyset$.

Then when we introduce v_2, we initialize its label to $\emptyset$ as well. However, this creates a collision, so we build the edge v_1v_2; if this edge is red, then we update the label of v_2 to r.

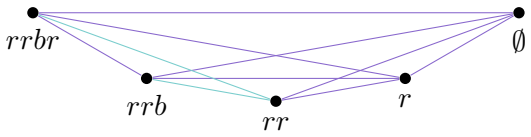
Then when we introduce v_3, we first label it $\emptyset$ as well. This again creates a collision with v_1, so we build the edge v_1v_3 and check its color; if this color is red, then we change the label of v_3 to r. Now this collides with v_2, so we build the edge v_2v_3; if its color is blue, then v_3 now gets the label rb.



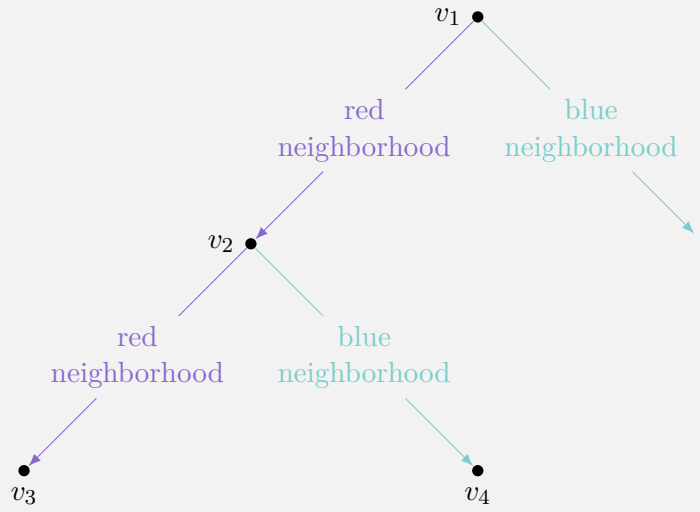
Now when we add v_4, we first label it with $\emptyset$. We then build the edge v_1v_4; if this is red, then we change the label to r and build the edge v_2v_4; if this is red as well, then we change the label to rr.



This process ensures that every vertex gets a different string, so once we've built 4^t vertices, some vertex will have to have a string of length $2t$, and such a string will have at least t reds or at least t blues. We can use this to obtain our monochromatic t-clique — for example, if we have a vertex v with label $rrbr$, then its last neighbor in our graph has label rrb, the neighbor before it has label rr, the one before that has label r, and the first has label $\emptyset$. Furthermore, we know the colors of the edges between all of these vertices (since each would have collided with all of the previous ones at some point), and we can read off our clique from the locations of the r's (here, it'll be the vertices labelled $rrbr$, rrb, r, and $\emptyset$).



Remark 2.3. You can think of this process as essentially Erdős–Szekeres done in a weird reverse order. Another way to think of it is that we have a sort of type tree going on — every time we add a new vertex, we ask which half of Erdős–Szekeres we would have been in on the first step, the second, and so on (where we embed vertices in numerical order) in order to eventually get to the current vertex; and we only build the edges in the path up the tree at each vertex.



§2.2.1 The off-diagonal case

We'll now consider the off-diagonal case — specifically, we'll look at $r_3(4,t)$.

Theorem 2.4

We have $2^{\Omega(t \log t)} \leq r_3(4,t) \leq 2^{O(t^2 \log t)}$.

Remarkably, this interval overlaps with our interval for $r_3(t,t)$ — so we don't even know that the growth rate in the diagonal case is bigger than the growth rate in the off-diagonal case.

In order to prove an upper bound, we can again try to run the Erdős–Rado online argument (where we build edges in an auxiliary graph $\mathcal{A}$). We want our graph $\mathcal{A}$ to have either a red triangle or a blue K_{t-1} — if $\mathcal{A}$ has a red triangle then our hypergraph has a red 4-clique, and if $\mathcal{A}$ has a blue K_{t-1} then our hypergraph has a blue t-clique. So getting a bound on $r_3(4,t)$ corrresponds to putting in the right off-diagonal *graph* bound, where we again pay a factor of 2 for each edge — this gives

$$r_3(4,t) \leq 2^{\binom{r_2(3,t)}{2}} \leq 2^{t^{4-o(1)}}.$$

Remark 2.5. This is the bound we get from the Erdős–Rado argument *without* playing the game (i.e., making the choices of which edges to build adaptively). People haven't really thought about what would happen if you optimized the game in this argument, because there's another improvement we'll see soon (and so people optimize that improvement and the game simultaneously). But just optimizing the game here might improve t^4 to t^3.

However, this is quite inefficient, and Conlon–Fox–Sudakov found an improvement. The idea is that we don't want to treat red and blue edges equally — we want to *weight* so that we pay a much heavier cost for building red edges than blue edges. (This makes sense because we win if we build a red triangle or a blue K_{t-1}, so we would expect to build much fewer red edges than blue edges.) This means we color the edge v_iv_j in $\mathcal{A}$ red if at least p of the edges v_iv_ju in our hypergraph are red (over all candidate vertices u), and blue otherwise (for some weight p — in the diagonal case we used $p = \frac{1}{2}$).

So there's three ideas we've seen here — greedy embedding by pairs, making our choices adaptively, and weighting by p in the off-diagonal case (it turns out the optimal value of p is around $\frac{1}{t}$). Roughly speaking, this is the most common way to get upper bounds for off-diagonal Ramsey numbers; but there's another way that works for very sparse hypergraphs, which we'll see next.

§2.3 Bounds for sparse hypergraphs

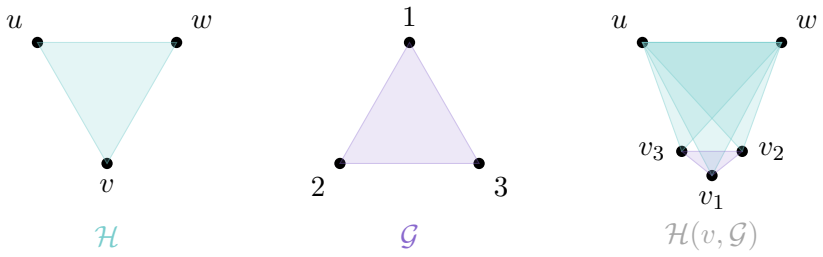
We'll now consider off-diagonal Ramsey numbers $r(\mathcal{H}, K_t^{(3)})$, where $\mathcal{H}$ is a fixed 3-graph and we're interested in the growth rate as $t \to \infty$. When $\mathcal{H}$ is 'sparse' or 'blowup-like' in some sense, then there's a third kind of greedy embedding argument we can use — where we add a single edge at a time, using supersaturation. For example, one specific $\mathcal{H}$ we'll look at is $K_{s,s,s}^{(3)}$ — this is the simplest possible blowup (it's just the blowup of a single edge).

Definition 2.6. Given 3-graphs $\mathcal{H}$ and $\mathcal{G}$, we define $\mathcal{H}(v, \mathcal{G})$ to be the blowup of $\mathcal{H}$ where we replace the vertex $v \in V(\mathcal{H})$ with a copy of $\mathcal{G}$.

In other words, we take a single vertex v in $\mathcal{H}$, clone it (so that its clones have the same adjacency relations with the rest of $\mathcal{H}$), and put a copy of $\mathcal{G}$ in these clones.

Example 2.7

If $\mathcal{H}$ and $\mathcal{G}$ are both a single edge, then $\mathcal{H}(v, \mathcal{G})$ is the 5-vertex graph with vertices u, w, v_1, v_2, v_3 and edges uwv_1, uwv_2, uwv_3, and $v_1v_2v_3$.



$\mathcal{H}$ $\mathcal{G}$ $\mathcal{H}(v, \mathcal{G})$

For this operation, we can prove bounds such as the following.

Theorem 2.8

For any 3-graphs $\mathcal{H}$ and $\mathcal{G}$, we have

$$r(\mathcal{H}(v,\mathcal{G}), K_t^{(3)}) \le r(\mathcal{H}, K_t^{(3)})^{v(\mathcal{H})} \cdot r(\mathcal{G}, K_t^{(3)}).$$

Corollary 2.9

If $\mathcal{H}$ is any iterated blowup of a single edge, then we have $r(\mathcal{H}, K_t^{(3)}) \le t^{O_{\mathcal{H}}(1)}$.

We'll consider *iterated blowups*, the 3-graphs we can form by using this operation repeatedly (so we're allowed to blow up vertices, and we're allowed to put other 3-graphs (which we can build in this way) in those blowups of our vertex).

Example 2.10

One example of an iterated blowup is the graph formed by taking a complete tripartite graph, then putting a complete tripartite graph in each of its parts, then putting a complete tripartite graph in each of these smaller parts, and so on.

Proof of theorem. Let $n = r(\mathcal{H}, K_t^{(3)})$ and $m = r(\mathcal{H}, K_t^{(3)})^{v(\mathcal{H})} \cdot r(\mathcal{G}, K_t^{(3)})$, and suppose that χ is a red-blue coloring of our giant graph $K_m^{(3)}$ with no red $\mathcal{H}(v,\mathcal{G})$ or blue $K_t^{(3)}$. Then every size-n subset of vertices has a red copy of $\mathcal{H}$. The idea is that we'll count the number of red copies of $\mathcal{H}$ in our big coloring, and since we can imagine building a copy of $\mathcal{H}$ by first choosing $\mathcal{H} \setminus v$ and then adding in v, we'll get a lower bound on the average number of extensions of a red $\mathcal{H} \setminus v$ into $\mathcal{H}$ in our coloring; we'll then take a red $\mathcal{H} \setminus v$ with lots of extensions and use it to find a red $\mathcal{H}(v,\mathcal{G})$.

First, we can bound the number of red copies of $\mathcal{H}$ in our coloring by

$$\#\text{copies of } \mathcal{H} \ge \frac{\binom{m}{n}}{\binom{m-v(\mathcal{H})}{n-v(\mathcal{H})}} \ge \left(\frac{m}{n}\right)^{v(\mathcal{H})}$$

(since each of the $\binom{m}{n}$ n-vertex subsets has a copy of $\mathcal{H}$, and each copy of $\mathcal{H}$ is overcounted $\binom{m-v(\mathcal{H})}{n-v(\mathcal{H})}$ times — the number of ways to extend it to a n-vertex subset). Meanwhile, the number of copies of $\mathcal{H} \setminus v$ is at most $m^{v(\mathcal{H})-1}$, and we have

$$\left(\frac{m}{n}\right)^{v(\mathcal{H})} \ge m^{v(\mathcal{H})-1} \cdot r(\mathcal{G}, K_t^{(3)})$$

by the way we defined m (this is true for all large enough m).

This means there exists a red $\mathcal{H} \setminus v$ which can be extended to a red $\mathcal{H}$ in at least $r(\mathcal{G}, K_t^{(3)})$ ways — this means there are at least this many potential vertices v we could use. Then we can find a red copy of $\mathcal{G}$

among these potential vertices v (since they don't have a blue $K_t^{(3)}$), and adding them to our red $\mathcal{H} \setminus v$ gives a red $\mathcal{H}(v, \mathcal{G})$. □

So when $\mathcal{H}$ is an iterated blowup of an edge, $r(\mathcal{H}, K_t^{(3)})$ is polynomial in t. A similar statement holds when $\mathcal{H}$ is *linear*, though with the clique replaced with a complete tri-partite 3-graph.

Theorem 2.11
If $\mathcal{H}$ is linear, then $r(\mathcal{H}, K_{t,t,t}^{(3)}) \le t^{O_{\mathcal{H}}(1)}$.

Remark 2.12. It matters that we have complete tri-partite 3-graphs here instead of cliques — this isn't true for cliques (there exist quasipolynomial lower bounds for the Ramsey numbers of some linear hypergraphs $\mathcal{H}$ against $K_t^{(3)}$).

Proof. We induct on the number of edges of $\mathcal{H}$. Suppose that this statement is true for $\mathcal{H} \setminus e$ (where e is some edge of $\mathcal{H}$), i.e., that there exists a constant c such that

$$r(\mathcal{H} \setminus e, K_{t,t,t}^{(3)}) \le t^c$$

for all t. Now let $e = uvw$, so that all other edges in $\mathcal{H}$ contain at most one of u, v, and w (because $\mathcal{H}$ is linear). The main idea is to again use supersaturation — given a copy of $\mathcal{H} \setminus \{u, v, w\}$, then the choices of which vertices we use for u, v, and w to extend it to a copy of $\mathcal{H} \setminus e$ are essentially independent (they don't interact with each other). So using supersaturation, we can show that if m is sufficiently large, then we can find a copy of $\mathcal{H} \setminus \{u, v, w\}$ which extends to at least $m^{3-\varepsilon}$ copies of $\mathcal{H} \setminus e$. Since the choices of u, v, and w are independent, this means the number of choices for *each* has to be at least $m^{1-\varepsilon}$ (and all combinations of these choices give us a copy of $\mathcal{H} \setminus e$).

So if we have a single red edge among the candidates for u, v, and w, then we're done — this gives us a red copy of $\mathcal{H}$ (choosing those candidates). Otherwise, all edges between them are blue, so we get a complete tripartite 3-graph in blue. □

§2.4 Another upper bound

We've seen several methods that give upper bounds; the most naive way of generalizing Erdős–Szekeres is the worst, but the other methods give good bounds in different cases. Finally, here's one more upper bound that doesn't fit into the above methods.

Theorem 2.13
We have $r(K_{t,t,t}^{(3)}, K_{t,t,t}^{(3)}) = 2^{\Theta(t^2)}$.

The proof of the upper bound is Kővári–Sós–Turán for hypergraphs — we pick the more dense color, and if n is large enough we can find a complete tripartite graph in this color; up to the constant in the exponent, this matches the Ramsey lower bound.

Remark 2.14. In general, when we take Ramsey problems for cliques and replace the cliques by tripartite graphs, then we're going halfway to Turán problems — we can start doing things like looking only at the denser color, which we can't do for cliques.

Remark 2.15. This proof works for higher uniformities too, and should give a bound of $2^{t^{k-1}}$.

§3 Lower bounds

Next, we'll talk about the three techniques for lower bounds.

§3.1 Random colorings

First, we'll get a lower bound on $r_3(t,t)$ using a purely random coloring — suppose that we have n vertices, and we color each edge red or blue with probability $\frac{1}{2}$. Then we have

$$\mathbb{E}[\#\text{red cliques}] = \binom{n}{t} \cdot 2^{-\binom{t}{3}} \approx n^t \cdot 2^{-t^3}$$

(ignoring constant factors in the exponent). So if $n \approx 2^{t^2}$ then this number is less than $\frac{1}{2}$, which means we win (there exists a coloring with no red or blue t-clique).

This method also gives the lower bound for $r(K^{(3)}_{t,t,t}, K^{(3)}_{t,t,t})$ in the earlier theorem.

So far, this is similar to the graph case. But what happens if we want to do the same for $r_3(4,t)$? In order for our random coloring to avoid having a red 4-clique, we need to choose our red probability p to be quite small; and it turns out that we can only get polynomial lower bounds in this way (using purely random models). This means in order to get good bounds, we need completely different kinds of ideas.

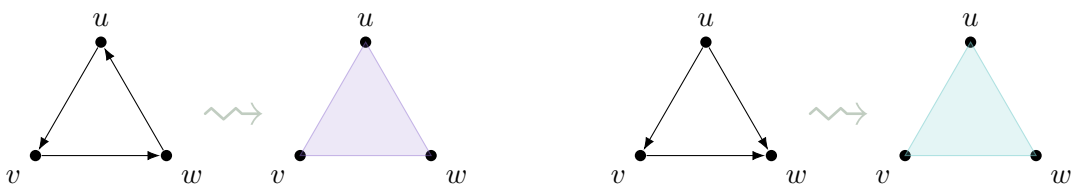
Remark 3.1. For the diagonal case, the bound of $r_3(t,t) > 2^{\Omega(t^2)}$ from this argument is the best bound we have — we don't even have a better constant in the exponent. (We have some induced colorings which give the same bound but with a worse constant in the exponent, and we don't have anything better.)

§3.2 Induced colorings

There's two magical ideas in this area, due to Erdős and Hajnal — *induced colorings* and *stepping up*. We'll talk about induced colorings first.

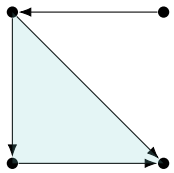
Theorem 3.2 (Erdős–Hajnal)
We have $r_3(4,t) \geq 2^{\Omega(t)}$.

Proof. First, we start with a random *tournament* T (on our vertex set with $n = 2^{\Omega(t)}$ vertices) — we take a complete graph and orient its edges uniformly at random. We then obtain an induced coloring from this random tournament as follows: for each triple of vertices u, v, and w, we color the edge uvw red if u, v, and w form a cyclic triangle in the tournament, and blue if they form a transitive triangle.



(In general, by an *induced coloring* we mean that we take a 2-uniform structure, and lift it to a 3-uniform structure by looking at its induced subgraphs.)

The magic is that we can't have a red 4-clique in our induced coloring — if we have 4 vertices in our tournament then each has in-degree or out-degree at least 2, and a vertex and its two out-neighbors (or two in-neighbors) will not form a cyclic triangle.



On the other hand, if we have a blue t-clique in our graph, then all triples among these t vertices are transitive, so the *entire* subtournament on these t vertices must be transitive. This is essentially as unlikely as a monochromatic clique in a randomly 2-colored graph — the probability of a tournament on t given vertices being transitive is $t! \cdot 2^{-\binom{t}{2}}$, and the $t!$ is a lower-order term, so the first moment calculation works out just as in the graph case (i.e., the lower bound for $r_2(t,t)$ via a random construction). Explicitly, if $n = 2^{ct}$ (for some c), then with positive probability the hypergraph coloring has no blue t-clique. □

Remark 3.3. There are a bunch of generalizations of this argument, which we will discuss in the afternoon seminar.

§3.3 Stepping up

We'll now look at the final kind of lower-bound colorings, namely *stepping up*. The most important example of stepping up is the recursive bound for $r_k(t,t)$ for uniformities $k \geq 4$.

Theorem 3.4

For all $k \geq 4$, we have $r_k(t,t) \geq 2^{r_{k-1}(\varepsilon t, \varepsilon t)}$ (for some $\varepsilon > 0$).

So once we reach uniformity 3, everything afterwards grows in tower height. For $k = 3$, this doesn't quite work, but it almost does — we can use stepping-up to get a similar statement, but it requires more colors.

Theorem 3.5

We have $r_3(t,t,t,t) \geq 2^{2^{\Omega(t)}}$.

For two colors, the best lower bound we know is single-exponential in t^2; but we can get double-exponential bounds with four colors. This means we know the exact tower heights for four colors or more for *all* uniformities (we can get a matching upper bound by generalizing our earlier proof of the 2-color upper bound), but we don't for 2 or 3 colors.

The idea is as follows — suppose we have a $(k-1)$-uniform good Ramsey coloring χ of $[n]$. We want to lift it to a k-uniform Ramsey coloring φ of $\{0,1\}^n$ (which is exponentially large in n).

To do this, we first need a natural way of going from k-tuples of binary strings (these are the things we're trying to color in φ) to $(k-1)$-tuples of indices in those strings (these are the things that are colored in χ).

Definition 3.6. Given two (distinct) binary strings $a, b \in \{0,1\}^n$, we define $\delta(a,b)$ to be the leftmost bit at which a and b differ.

In other words, given two strings, we consider the leftmost bit at which one is 0 and the other is 1.

Fact 3.7 — If $v_1 < \cdots < v_k$ are length-n binary strings, then there are at most $k-1$ distinct values among $\delta(v_i, v_j)$ over all $i \neq j$.

Example 3.8

For the strings 0000, 0001, 0100, 0110, and 0111, we have:

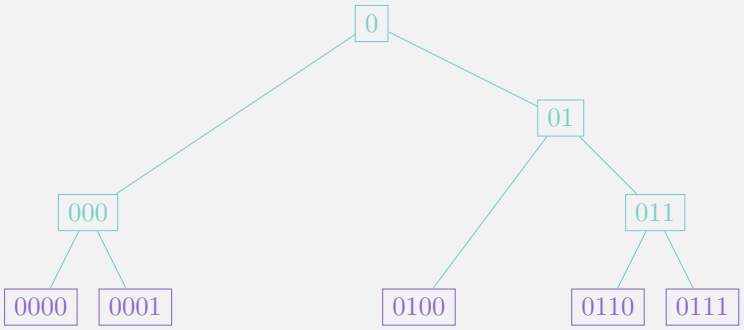
- $\delta(0000, 0001) = 4$;
- $\delta(0000, 0100) = \delta(0000, 0110) = \delta(0000, 0111) = 2$; the same holds if we replace 0000 with 0001;
- $\delta(0100, 0110) = \delta(0100, 0111) = 3$;
- $\delta(0110, 0111) = 4$.

In fact, we can write down explicitly what the $k-1$ possible values are — they're precisely

$$\{\delta(v_1, v_2), \delta(v_2, v_3), \ldots, \delta(v_{k-1}, v_k)\}$$

(it can be shown that all other δ-values are equal to one of these). This is a cute fact that lets us take k-tuples of strings and pass to $(k-1)$-tuples of indices.

Remark 3.9. One way of interpreting this fact is as the statement that a binary tree with k leaves has $k-1$ internal vertices (i.e., non-leaves) — here we think of our k strings as leaves of the tree, and the levels at which the common ancestors (i.e., the internal vertices) are as the values of δ. (There are $k-1$ internal vertices, so they occupy at most $k-1$ levels.)



The idea of stepping up is basically (with one important modification) that we'll obtain φ by taking our k-tuple of vertices $\{v_1, \ldots, v_k\}$, applying δ to get a $(k-1)$-tuple, and then applying χ to this $(k-1)$-tuple to get a color (i.e., red or blue). In other words, given any k-tuple of vertices, we get a $(k-1)$-tuple using δ, and then read off our color from the $(k-1)$-uniform Ramsey coloring.

But this doesn't actually work — let's see why not and how to fix it. For concreteness, let's take $k = 4$, and suppose that χ is a Ramsey coloring of $[n]$ for $r_3(t, t)$. Now to find $\chi(v_1v_2v_3v_4)$ (for $v_i \in \{0,1\}^n$, sorted so that $v_1 < v_2 < v_3 < v_4$), we want to compute $\delta_1 = \delta(v_1, v_2)$, $\delta_2 = \delta(v_2, v_3)$, and $\delta_3 = \delta(v_3, v_4)$, and then look at $\chi(\delta_1\delta_2\delta_3)$. But this doesn't actually work — for one thing, δ_1, δ_2, and δ_3 might not all be distinct (in which case this color is not even defined). So we'll sometimes use χ to color, and sometimes just use the pattern formed by the δ_i — we define

$$\varphi(v_1v_2v_3v_4) = \begin{cases} \chi(\delta_1\delta_2\delta_3) & \text{if } \delta_1,\ \delta_2,\ \delta_3 \text{ is monotone} \\ \text{red} & \text{if } \delta_1 < \delta_2 > \delta_3 \\ \text{blue} & \text{if } \delta_1 > \delta_2 < \delta_3. \end{cases}$$

(Note that if $v_1 < v_2 < v_3$ then $\delta(v_1, v_2) \neq \delta(v_2, v_3)$ — this means that *adjacent* δ_i's are always distinct. Non-adjacent δ_i's don't have to be distinct, but in the monotone case they are.)

In other words, if our δ_i's are monotone (either increasing or decreasing) then we color based on χ; otherwise, we color based on which non-monotone pattern they form.

Claim 3.10 — This coloring φ has no monochromatic $K_{2t}^{(4)}$.

Proof. Assume for contradiction that it has a red $K_{2t}^{(4)}$; this means we have $2t$ vertices $v_1, \ldots, v_{2t}$ such that for every four of them — for example, $v_1 < v_2 < v_3 < v_4$ — the edge consisting of these four vertices is red. There's two possible reasons this edge could be red — either $(\delta_1, \delta_2, \delta_3)$ is monotone and red in χ, or it's increasing and then decreasing. In particular, it can't be decreasing and then increasing.

Now define $\delta_i = \delta(v_i, v_{i+1})$. Then when we scan through $v_1, \ldots, v_{2t}$ in order, looking at 4 consecutive vertices at a time, we find that $\delta_1, \ldots, \delta_{2t-1}$ must be monotone increasing up to some point and then monotone decreasing past that point (since it doesn't have any decreasing-increasing adjacent pattern). So either at least half is monotone increasing, or at least half is monotone decreasing.

Consider the larger half, which has at least t vertices. Then in φ, all the edges between these vertices must have been colored according to χ; this means these t vertices form a monochromatic t-clique in χ, which is a contradiction. □

In general, what stepping up (for any uniformity and number of colors) looks like is that we look at the patterns of the δ_i's; for the monotone patterns we color using χ, and we do some combinatorics to find a way to divide the non-monotone patterns into red and blue such that a monochromatic clique ensures a long monotone stretch. This works for all uniformities $k \geq 4$, with more complicated rules for how to divvy up the non-monotone patterns.

Why does this break when we're trying to step up from uniformity 2 to 3, and why do we need to double the number of colors? Suppose we tried to run this scheme to step up to uniformity 3 — this means we start with a 2-uniform Ramsey coloring χ of $[n]$ (where $n = 2^{t/2}$), and we define

$$\varphi(v_1v_2v_3) = \chi(\delta_1\delta_2)$$

for all edges $v_1v_2v_3$ (since there are no non-monotone patterns of length 2). Then we can't guarantee anything about monotone stretches in the δ_i's, so this doesn't work.

So we can't step up to uniformity 3 with just two colors. But with *four* colors we can — we define

$$\varphi(v_1v_2v_3) = (\chi(\delta_1\delta_2), \mathbf{1}_{\delta_1<\delta_2})$$

(here our colors are $(\text{red}, 0)$, $(\text{red}, 1)$, $(\text{blue}, 0)$, and $(\text{blue}, 1)$). Now a monochromatic clique in φ does need to have monotone δ_i's. This means any monochromatic clique in the 3-uniform coloring descends to one in the 2-uniform coloring, and we win.

So this gives the bound $r_3(t, t, t, t) \geq 2^{2^{\Omega(t)}}$, but we don't know any bound like this for 2 colors. For 3 colors, we have an 'in-between' bound — we have a bound of

$$r_3(t, t, t) \geq 2^{2^{(\log t)^c}}$$

for some c (using some kind of lopsided stepping-up). But we still don't know for 2 or 3 colors exactly what the growth rate should be.