YUFEI ZHAO: We've been spending the past few lectures discussing Szemeredi's Regularity Lemma. And one of the first applications that we discussed of the Regularity Lemma is the triangle removal Lemma. So today, I want to revisit this topic and show you a strengthening of the Removal Lemma for which new regularity techniques are needed.

But first, recall the graph removal Lemma. In the graph removal Lemma, we have that for every graph H and epsilon bigger than zero, there exists some delta such that if an N vertex graph has fewer than delta and to the number of vertices of H, many copies of H, then it can be made H-free by removing fewer than epsilon N squared edges.

Even in the case when H is a triangle, when this is called a triangle removal Lemma, even in that case, basically the regularity method is more or less the only way that we currently know how to prove this theorem. So we saw this a few lectures ago.

What I would like to discuss today is a variant of this result where instead of considering copies of H, we're now considering induced copies of H. OK? So this is the induced graph removal Lemma where the only difference is that the hypothesis is now going to be changed to induced copies of H. And the conclusion is that you can make the graph induced H-free.

So let me remind you, the difference between the induced graph subgraph and the usual subgraph. So we say that H is an induced copy of G, induced subgraph of G. If one can obtain H from G by deleting vertices of G. You're not allowed to delete edges, but only allowed to delete vertices. So in other words, the four cycle is not an induced subgraph because, well, if you select four vertices, you don't generate this four cycle. You get extra edges. So it is a subgraph, but not an induced subgraph.

So it is a theorem, the induced graph removal Lemma. So it's a theorem, and let's discuss how we may prove that theorem. Question.

OK, question is, why is it stronger than the graph removal lemma? So it's not stronger, but we'll see the relationship between the two. So I claim that it is more difficult to do this theorem. Any more questions? So let's pretend for a second that whatever's in here is not quite true.

So here's an example. For example, if your H is three isolated vertices. So what is that saying? We're looking at copies of H which are three isolated vertices. So really you are looking at

triangles in g complement. So this is exactly the triangle removal lemma in the complement of g, but you can't get rid of these guys by removing edges. So we need to make the modification where instead of removing these edges, we need to both remove and add by adding or deleting.

So maybe at the same time. So you're allowed to add some edges, delete some edges. But in total, you change no more than epsilon n squared edges. So those are sometimes also known as the edit distance. You're allowed to change edges. So you can add edges and delete edges. Any questions about the statement?

All right, so let's think about how would you prove this result following the proof that we did for the triangle removal lemma. So let's pretend that we go through this proof and think about what could go wrong. So remember in the application of the removal lemma, so the recipe has three steps. The first step we do a partition.

So we partition applying Szemeredi's regularity lemma to this partition. And the second step is do a cleaning, and the two key things that happen in the cleaning is we remove low density pairs of parts and irregular pairs. And the third step we claim that once we do the cleaning, once we remove those edges, the resulting graphs should be H3. Because if we're not H3, then by considering the vertex parts where H lie and applying the counting lemma, you can generate many more copies of H. So these were the three main steps in the proof of the triangle removal lemma.

So let's see what happens when we try to apply this strategy to the induced version. I mean, the partition you still do the regularity partition. Nothing really changes there. So let's see in the cleaning step what happens. For low density pairs-- well, so now we need to think about not just low density pairs, but also high density pairs. Because in the induced, we think about edges and non-edges at the same time. So you might think of a strategy which is like the edge density is less than n. So less than epsilon, then you remove all those edges. And if the edge density is bigger than 1 plus epsilon, then you add all of those edges in.

So this is the natural generalization of our strategy for triangle removal lemma for the induced setting. So so far, everything's still OK. But now what would you do for the irregular pairs? That's problematic.

Previously for triangle removal lemma, we just said if a pair is irregular, get rid of that pair and

it will never show up in the counting stage. But that strategy no longer works. Because for example, if your graph H being counted is this here, you do the regularity partition, and one of your pairs is irregular. So you, let's say, get rid of all those edges in between. Then maybe you have some embedding of H where you are going to use the removed edges.

And now you don't have a counting lemma. You cannot say, I found this copy of H in my changed graph. And by the counting lemma I could get many copies of H because you have no control over this irregular pair anymore. So the fact that you have to add and remove makes it unclear what to do here, and this is a big obstacle in the application of the regularity lemma to the induced removal lemma application. Any questions about this obstacle?

So make sure you understand why this is an issue. Otherwise you won't really appreciate what will happen next. So somehow we need to find some kind of regularity partition to get no irregular pairs. So the question is, is there a way to partition so that there are no irregular pairs?

For those of you who have started your homework problem on time, you realize that the answer is no. So one of the homework problems is for you to show that for the specific graph known as the half graph. So there was an example in homework that for the half graph-- so you'll see in the homework what this graph is-- you cannot partition it so that you get rid of all irregular pairs. Irregular pairs are necessary in the statement of regularity lemma.

So what I want to show you today is a way to do what's called a strong regularity lemma in which you obtain a somewhat different consequence that will allow you to get rid of irregular pairs in the more restricted setting. So this is the issue, the irregular pairs.

Before telling you what this regularity lemma is, I want to give you a small generalization of the induced graph removal lemma, or just a different way to think about the statement. And you can think of it as a colorful version instead of induced where you have edges and no edges. You can also have colored edges. So colorful removal lemma, although this name is not standard. So colorful-- so when we talk about graphs, it's colorful graph removal lemma.

So for every k, r, and epsilon, there exists delta such that if curly H is a set of r edge of the complete graph on little k vertices. So edge coloring just means using r colors to color the edges. So there are no restrictions about what are allowed, what are not allowed. So just a set of possible r colorings.

Then if the complete graph-- say it slightly differently. So then every r edge coloring of the complete graph on n vertices with fewer than delta fraction of its k vertex subsets, say k vertex subgraphs, belonging to the script H. So every such graph can be made curly H free by recoloring, so using the same r colors, a fewer than epsilon fraction. So less than epsilon fraction of the edges of this kn.

So in particular, the version that we just stated, the induced version, so the induced graph removal lemma, is the same as having two colors and H having exactly one red-blue coloring of k of the complete graph on the same number of vertices as H. So you color red the edges and blue the non-edges, for instance.

And you're saying, I want to color the big complete graph with red and blue in such a way that there are very few copies of that pattern. So then I can recolor the red and blue in a small number of places to get rid of all such patterns. So having a colored pattern somewhere in your graph in this complete graph coloring is the same as having an induced subgraph. Yeah?

AUDIENCE: So after done-- like the statement after done is a really long sentence. Can I--

YUFEI ZHAO: Yeah, OK. So every r edge coloring of kn with a small number of patterns can be made h-free by recoloring a small fraction of the edges. So like in a triangle removal lemma, every graph with a small number of triangles can be made triangle-free by removing a small number of edges. Any other questions?

So this is a restatement of the induced removal lemma with a bit more generality. It's OK if you like this one more or less, but let's talk about the induced version from now on. But the same proofs that I will talk about also applies to this version where you have somewhat more colors.

So the variant of the regularity lemma that we'll need is known as a strong regularity lemma. To state the strong regularity lemma, let me recall a notion that came up in the proof of Szemeredi's regularity lemma. And this was the notion of an energy. So recall that if you have a partition, denoted P. So if this is a partition of the vertex set of a graph, G, and here n is the number of vertices, we defined this notion of energy to be this quantity denoted q, which is basically a squared mean of the densities between vertex parts appropriately normalized if the vertexes do not all have the same size.

In the proof of Szemeredi's regularity lemma, there was an important energy increment step which says that if you have some partition p that is not epsilon regular, then there exists a

refinement, Q. And this refinement has the property that Q has a small number of pieces, or not too large as a function of P So it's bounded at least in terms of P. But also if P is not epsilon regular, then the energy of Q is significantly larger than the energy of P. So remember, this was an important step in the proof of regularity lemma.

So to state the strong regularity lemma, we need that notion of energy. And the statement of the strong regularity lemma, if you've never seen this kind of thing before, will seem a bit intimidating at first because it involves a whole sequence of parameters. But we'll get used to it.

So instead of one epsilon parameter, now you have a sequence of positive epsilons. And part of the strength of this regularity lemma is that depending on the application you have in mind, you can make the sequence go to zero pretty quickly. Thereby increasing the strength of the regularity lemma.

So there exists some m bound, which depends only on your epsilons such that every graph has not just one, but now we're going to get a pair of vertex partitions P and Q with the following properties. So first, P refines-- so Q refines P. So it's a pair of partitions, one refining the other.

The number of parts of Q is bounded just like in the usual regularity lemma. The partition P epsilon 0 regular. And here is the new part that's the most important one.

Q is very epsilon regular. So it's not just epsilon 0 regular, it's epsilon sub the number of parts of P regular. So you should think of this as extremely regular because you get to choose what the sequence of epsilon is. And finally, the energy difference between P and Q is not too big.

This is the statement of the strong regularity lemma. It produces for you not just one partition, but a pair of partitions. And in this pair of partitions, you have one partition, P, which is similar to the one that we obtained from Szemeredi's regularity lemma is some epsilon 0 regular, but we also get a refinement Q. And this Q is extremely regular. So you can think that is P, then Q is an extremely regular refinement of P. Any questions about the statement of the strong regularity lemma?

So the sequence of epsilons gives you flexibility on how to apply it, but let's see how to prove it. And the proof is once you understand how this works, conceptually it's pretty short. But let me do it slowly so that we can appreciate this sequence of epsilons.

And the idea is that we will repeatedly apply Szemeredi's regularity lemma. So start with the regularity lemma. We'll apply it repeatedly to generate a sequence of partitions. So first, let me remind you a statement of Szemeredi's regularity lemma. This is slightly different from the one that we stated, but comes out of the same proof.

So for every epsilon, there exists some m0 which depends on epsilon such that for every partition P0, so starting with some partition-- so actually, let me start with just P. So if you start with some partition of the vertex set of g, there exists a refinement P prime of P into at most-- OK, so the refinement has is such that with each part of P refined into at most m0 parts such that P prime, the new partition, is epsilon regular.

So this is a statement of Szemeredi's regularity lemma that we will apply repeatedly. So in the version that we've seen before, we would start with a trivial partition. And applying refinements repeatedly in the proof to get a partition into a bounded number of parts such that the final partition is epsilon regular. But instead, in the proof of the regularity lemma if you start with not a trivial partition but start with a given partition and run this exact same proof, you find this consequence. Except now you can guarantee that the final partition is a refinement of the one that you are given.

So let's apply the statement, and we obtain a sequence of partitions of g-- the vertex set of g-- starting with P0 being a trivial partition, and so on. Such that each partition, each P sub i plus 1 refines the previous one, and such that each P sub i plus 1 is epsilon sub a P sub i regular.

So you apply the regularity lemma with parameter based on the number of parts you currently have. Applied to the current partition, you get a finer partition that's extremely regular. And you also know that the number of parts of the new partition is bounded in terms of the previous partition.

All right. Any questions so far? So now we get this sequence of partitions. We can keep on doing this. So g could be arbitrarily large, but eventually we will be able to obtain the last condition here, which is the only thing that is missing so far.

So since the energy is bounded between 0 and 1, there exists some i at most 1 over epsilon 0 such that the energy goes up by less than epsilon 0. Because otherwise your energy would exceed 1. So now let's set P to be this Pi, and Q to be this, the refinement-- the next term in the partition. And what we find is that the-- so then you have basically all the conditions. So p

is epsilon 0 regular, because it is epsilon-- the previous term, which is at most epsilon 0 regular. And you have this one as well, and this one as well.

And we want to show that the number of parts of Q is bounded. And that's basically because each time there was a bound on the number of parts which depends only on the regularity parameters, and you're repeating that bound a bounded number of times. So Q is-- so it's bounded as a function of the sequence of epsilons-- this infinite vector of epsilons, but it is a bounded number. You're only iterating this bound a bounded number of times. And that finishes the proof. Any questions?

It may be somewhat mysterious to you right now why we do this, so we'll get that application a second. But for now, I just want to comment a bit on the bounds. Of course, the bounds depend on what epsilon i's do you use. And typically, you want the epsilon i's to decrease with more parts that you have. And with almost all reasonable applications of this regularity lemma, the strong regularity lemma-- so for example, with epsilon i being some epsilon divided by, let's say, i plus 1 or any polynomial of the i's-- or you can even let it decay quicker than that, as well.

You see, basically what happens is that you are applying this m0 bound. m0 applied in succession 1 over epsilon times. In the regularity lemma, we saw that the m0 that comes out of Szemeredi's graph regularity lemma is the tower function. So the tower function, that's a tower of i is defined to be the exponential function iterated i times. So of course, I'm being somewhat loose here with the exact dependence, but you get the idea that now we want to apply the tower function i times.

Instead of iterating the exponential i times, now you iterate the tower function i times. And some of you laughing, this is an incredibly large number. It's even larger than the tower function.

So in literature, especially around the regularity lemma, this function where you iterate the tower function i times is given the name wowzer. [LAUGHTER] As in, wow, this is a huge number. So it's a step up in the Ackerman hierarchy. So if you repeat the wowzer function i times, you move up one ladder in the Ackerman hierarchy and this hierarchy of rapidly growing functions. But in any case, it's bounded and that's good enough for us. Any questions so far? Yeah?

AUDIENCE: What do you call like [INAUDIBLE]

YUFEI ZHAO: Yes, so question is, what do you call wowzer iterated? I'm not aware of a standard name for that. Actually, even the name wowzer somehow is very common in the combinatorics community, but I think most people outside this community will not recognize this word. Any more questions? So another way it's a step up in Ackerman hierarchy. So it's enumerated one, two, three, four, you know, if you keep going up.

All right. Another remark about this strong regularity lemma is that it will be convenient for us-- actually, some are more essential compared to our previous applications-- to make the parts equitable. So P and Q equitable. And basically, the parts are such that all the-- the partitions are such that all the parts have basically the same number of vertices. So I won't make it precise, but you can do it. It's not too hard to do it. And you can prove it similar to how I described how to modify the proof of the regularity level. So I won't belabor that point, but we'll use the equitable version.

All right, so how does one use this regularity lemma? Let me state a corollary, and let me call this a corollary star because you actually need to do some work to get it to follow from the strong regularity lemma. But the corollary is the version that we will apply that if you start with a decreasing sequence of this epsilon, then there exists a delta such that the following is true.

Every n vertex graph has an equitable vertex partition, call it i through the k, and a subset Wi of each Vi such that the following properties hold. First, all the W's are fairly large. They're at least constant proportion of the total vertex set. Between every pair of Wi Wj, it is epsilon sub k regular.

And this is the point I want to emphasize. So here there are not you regular pairs anymore. So it is every. So no irregular pairs between the Wi's, and also we need to include the case when i equals the j, as well. So each Wi is regular with itself.

And furthermore, the edge densities between the V's are similar to the edge densities between the corresponding W's. And here it is for most pairs for all but at most epsilon k square pairs. Epsilon 0, yeah. At most epsilon 0. Any questions about the statement?

So let me show you how you could deduce the corollary from the strong regularity lemma. So first, let me draw your picture. So here you have a regularity partition. And so these are your V's, and inside each V I find a W such that if I look at the edge sets between pairwise blue

sets, including the blue sets with themselves, it is always very regular. And also, the edge densities between the blue sets is mostly very similar to the edge density between their ambient white sets.

OK, so let me say a few words-- I won't go into too many details-- about how you might deduce this corollary from the strong regularity lemma. So first let me do something which is slightly simpler, which is to not yet require that the blue sets, Wi's, are regular with themselves. So without requiring this as regular so we can obtain the Wi's by picking a uniform random part of the final partition, Q, inside each part of P in the strong regularity lemma.

So you have the strong regularity lemma, which produces for you a pair of partitions like that. So it produces for you a pair of partitions. And what we will do is to pick one of these guys as my W, pick one of these guys at random, and pick one of those guys at random.

Because W is so extremely regular, most of these pairs will be regular. So with high probability, you will not encounter any irregular pairs if you pick the W's randomly as parts of Q. So that's the key point. Here we're using that Q is extremely regular. So all the Wi Wj is regular for all i not equal to j with high probability.

But the other thing that we would like is that the edge densities between the W's are similar to those between the V's. And for that, we will use this condition about their energies being very similar to each other. So the third consequence, C, is-- it's a consequence of the energy bound.

Because recall that in our proof of the Szemeredi regularity lemma there was an interpretation of the energy as the second moment of a certain random variable which we called z. And using that interpretation, I can write down this expression like that. We are here assuming for simplicity that Q is completely equitable, so all the parts have exactly the same size. Z of Q is defined to be the edge density between Vi and Vj for random ij. So this is a random variable z. So you pick pair of parts uniformly, or maybe with some weights if they're not exactly equal. And you evaluate the edge density.

So this energy difference is the difference between the second moments. And because Q is a refinement of P, it is the case that this difference of L2 norms is equal to the second moment of the difference of the random variables. So we saw a version of this earlier when we were discussing variance in the context of the proof of the similar irregularity lemma.

Here it's basically the same. You can either look at this inequality part by part of V, or if you like to be a bit more abstract then this is actually a case of Pythagorean theorem. If you view these as vectors in a certain vector space, then you have some orthogonality. So you have this sum of squares identity. Where does part A come from? So part A, we want the parts, that Wi's to be not too small, but that comes from a bound on the number of parts of Q.

So so far this more or less proves the corollary except for that we simplified our lives by requiring just that the i not equal to j, the Vi Vj's are regular. But in the statement up there, we also want the Vi's-- so the Wi's ice to be regular with themselves, which will be important for application. So I won't explain how to do that, and part of the reason is that this is also one of your homework problems. So in one of the homework problems problem set 3, you were asked to prove that every graph has a subset of vertices that is of least constant proportion such that it is regular with itself. And the methods you use there will be applicable to handle the situation over here, as well.

So putting all of these ingredients together, we get the corollary whereby you have this picture, you have this partition. I don't even require the Vi's to be regular. That doesn't matter anymore. All that matters is that between the Wi's they are very regular, and that there are no irregular parts between these Wi's.

And now we'll be able to go back to the induced graph removal lemma where previously we had an issue with the existence of irregular pairs in the use of Szemeredi regularity partition, and now we have a tool to get around that. So next we will see how to execute this proof, but at this point hopefully you already see an outline. Because you no longer need to worry about this thing here. Let's take a quick break.

Any questions so far? Yes?

AUDIENCE: Why are we able to [INAUDIBLE]

YUFEI ZHAO: OK, so the question was, there was a step where we were looking at some expectations of squares. And so why was that identity true? So if you look back to the proof of Szemeredi's regularity lemma, we already saw an instance of that inequality in the computation of the variance.

So you know that the variance of x, on one hand it is equal to where mu is the mean of x. And

on the other hand, it is equal to this quantity. So you agree with this formula? And you can expand it to prove it, and the thing that-- the question that you raised basically you can prove by looking at this formula part by part. Any more questions?

So let's now prove the induced graph removal lemma. And we'll follow the regularity partition, but with a small twist that Instead of using Szemeredi's regularity lemma, we will use that corollary up there. So let's prove the induced graph removal lemma.

So the three steps. First, we do partition. So let's suppose you have a-- so we suppose g is like above. You have very few induced copies of H. Let's apply the corollary to get a partition of the vertex set of g into k parts. And inside each part I have a W. Satisfying the following properties that each Wi Wj is regular with the following parameter which will come out of later when we need to use the counting lemma. But it's some number, but don't worry too much about it.

So here I'm going to-- so let's say H has little H vertices. So between Wi Wj it is this regular. So we actually have not yet used the full strength of the corollary where I can make the regularity even depend on k. So we will not need that here, but we'll need it in a later application. So the exponent is little H.

OK, so other properties are that the densities between the Vi's and the Wi's do not differ by more than epsilon over 2 for all but a small fraction-- so epsilon k squared over 2-- pairs. And finally, the sizes of the Wi's are at least delta 0 times n where delta 0 depends only on epsilon. Epsilon and H.

This is the partition step, so now let's do the cleaning. In the cleaning step, basically we're not going to-- I mean, there is no longer an issue of irregular pairs if we only look at the Wi's. So we just need to think about the low density pairs or whatever the corresponding analog is.

And what happens here is that for every i less than j, and crucially including when i equals to j, if the edge densities between the W's is too small then we remove all edges between Vi and Vj. And if the edge density between the Wi's is too big, then we remove all edges. So we add all edges between Vi and Vj.

How many edges do we end up adding or removing? So the total number of edges added or removed from g is-- in this case, so if the edges density in g between the Vi's and Vj's is also

very small, then you do not remove very many edges. But most pairs of Vi and Vj have that property.

So you tidy up what kind of errors you can get from here and there, and you find that the total number of edges that are added or removed from g is less than, let's say, epsilon n squared. Maybe even get an extra factor of 2, but you know, upon changing some constant factors, it's less than epsilon n squared. So this is some small details you can work out. Here we're using-- asking, how is the density between Vi and Vj related to Wi and Wj? Well, for most pairs of i and j they're very similar. And there's a small fraction of them that are not similar, but then you lump everything in to this bound over here.

So maybe I need to-- let me just put a 2 here just to be safe. All right. So we deleted a very small number of edges, and now we want to show that the graph that has resulted from this modification does not have any induced H sub-graphs. And the final step is the counting step. So suppose there were any induced H left after the modification. So I want to show that, in fact, there must be a lot of H's-- induced H's originally in the graph, thereby contradicting the hypothesis.

So where does this induced H sit? Well, you have the V's, and inside the V's you have the W's. So suppose my H is that graph for illustration. And in particular, I have a non-edge. So I have an edge, and I also have a non-edge. So between these two, that's the non-edge.

So suppose you find a copy of H in the cleaned-up graph. Where can that cleaned up-- this copy of H sit? Suppose you find it here.

The claim now is that if this copy of H existed here, then I must be able to find many such copies of H in the corresponding yellow parts. Because between the yellow parts you have regularity, and you also have the right kinds of densities. Because if they didn't have the right kind of density, we would have cleaned it up already. So that's the ideal. If you had a copy of this H somewhere, then I zoom into the yellow parts, zoom into these W's, and I find lots of copies of H in between the W's.

So suppose-- let me write this down. So suppose the little V's, so the vertices, lies in the-- so I'm just indexing where a little v lies. The little v lies in big V sub phi V for some phi which since the vertices of H2 went through k. So now we apply counting lemma to embed induced copies of H in g where the vertex V in H is mapped to a vertex in the corresponding W.

And we would like to know that there are lots of such copies.

And the counting Lemma-- or rather, some variant, but I should read the counting lemma that we did last time and view it as a multi-partite version. Apply this so far part to part. So we find that the number of such induced copies is within a small error.

So that regularity parameter multiplied by the number of edges of H, which we already canceled out, multiplied by the product of these Wi's. So it's within this error of what you would suspect if you naively multiply the edge densities together along with the vertex densities.

So these factors are for the edges that you want to embed, and then I also need to multiply the densities for the long edges. So 1 minus these edge densities. So one way you can think of it is just consider the complement in g. So consider the complement of g to get this version here. And then finally, the product of the vertex set sizes.

And the point is that this is not a small number. So hence the number of induced copies of H in g is at least on the order of-- well, OK? So it's at least some number, which is basically this guy over here. So epsilon over 4 raised to-- all of these are constants, so that's the point. All of these guys are constants, minus-- so here is the main term, and then the error term.

And then the product of these vertex set sizes, and we saw that each vertex set is not too small. So you have lots of induced copies of H in g. Yep?

AUDIENCE: How do you do in the case where the density between [INAUDIBLE]

YUFEI ZHAO: OK, so can you repeat your question?

AUDIENCE: How are you dealing with the [INAUDIBLE]

YUFEI ZHAO: OK. So question, how do we deal with the all but epsilon over two pairs? So that comes up in the cleaning step in what I wrote in red in dealing with the number of total edges that are added or removed. So think about how many edges are added or removed.

In these non-exceptional pairs, the number of edges that are added or removed-- let's just think about added edges. So if the density of V is controlled by that of W, then the number of edges added-- or removed, in that case-- from all such pairs along with-- yeah. So you have epsilon n squared edges changed.

On the other hand, if this is not true then you only have epsilon k squared such pairs ij for which this cannot be true. So you also only have at most epsilon n squared edges added or removed in such cases. That answers your question? Yes?

AUDIENCE: Is that number 0?

YUFEI ZHAO: Is which number 0?

AUDIENCE: The number of induced edges for the [INAUDIBLE]

YUFEI ZHAO: The--

AUDIENCE: Yeah, the top board.

YUFEI ZHAO: Top board? Good. So asking about this number. So that should have been 2. Yes?

AUDIENCE: I don't see k anywhere.

YUFEI ZHAO: OK, so question, you don't see k appearing anywhere. So the k in the corollary, do you mean?

AUDIENCE: Yeah.

YUFEI ZHAO: So that hasn't come up yet. So it comes up implicitly because we need to lower bound the sizes of these W's. So this is partly why we need a bound on the number of parts, but it is true that we do not need epsilon k to depend on k in this application yet. I will mention a different application in the second where you do need that k.

OK, so the number of induced H in g is at least this amount. And that's a small lie. You need to maybe consider this is the number of homomorphic. Well, actually, no, we're OK. Never mind. So you can set delta to be this quantity here, and then that finishes the proof. So you have lots of induced copies of H in your graph which contradicts the hypothesis.

So that finishes the proof of the induced removal lemma, and basically the proof is the same as the usual graph removal lemma except that now we need some strengthened regularity lemma which allows us to get rid of irregular parts but in a more restricted setting. Because we saw you cannot completely get rid of irregular parts. Any questions? Yes?

AUDIENCE: [INAUDIBLE]

YUFEI ZHAO: So I want to address the question of why did I state this corollary in this more general form of a

decreasing sequence of epsilons? So first of all, with strong regularity lemmas, the strength is sometimes always nice to-- it's always nice to state it with this extra strength. Because it's the right way to think about these types of theorems. That the regularity on the parts depends-- you can make it depend on the number of parts so that you get much stronger control on the regularity.

But there are also some applications. For example, whether I will state next, an application where you do need that kind of strength. So here's what's known as the infinite removal lemma. Here we have not just a single pattern or a finite number of patterns we want to get rid of. For now we have infinitely many patterns. So for every curly H, which is a possibly infinite set of graphs. The graphs themselves are always finite, but this may be an infinite list. And an epsilon parameter.

There exists an H0 and a delta positive parameter such that every n vertex graph with at most delta-- so less than delta-- V to the H induced copies of H for every H in this family with fewer than H0 vertices. So every graph with this property can be made curly H free. So it means free of-- induced curly H free by adding or removing fewer than epsilon n squared edges.

So now instead of a single pattern you have a possibly infinite set of induced patterns and a want to make your graph curly H free-- induced curly H free. And the theorem is that if there exists some finite bound, H0, such that if you have few copies-- so for all the patterns up to that point-- then you can do what you need to do.

So take some time to even digest this statement, but it's somehow infinite versions-- the correct infinite version of the removal lemma if you have infinitely many patterns that you need to remove. And I claim that the proof is actually more or less the same proof as the one that we did here, except now you need to take your epsilon case, as in this corollary, to depend on k. You need to in some way look ahead in this infinite pattern. So here in proof, this epsilon k from corollary depends on k. And also it depends on your family of patterns H.

Finally, I want to mention a perspective-- a computer science perspective on these removal lemmas that we've been discussing so far. And that's in the context of something called property testing. And basically, we would like an efficient-- efficient meaning fast-- randomized algorithm to distinguish graphs that are triangle-free from those that are epsilon far from triangle-free. Where being epsilon far from triangle-free means that you need to change more than epsilon n squared edges here. n is, as usual, the number of vertices to make the graph

triangle-free. So the distance, the [INAUDIBLE] distance is more than epsilon away from being triangle-free.

So somebody gives you a very large graphing. n is very large. You cannot search through every triple vertices. That's too expensive. But you want some way to test if a graph is triangle-free versus very far away from being triangle-free.

So there's a very simple randomized algorithm to do this, which is to just try randomly sample a random triple of vertices and check if it's a triangle. So you do this. And just to make our life a bit more secure, let's try it some larger number of times. So some c of epsilon some constant number of times. And if you find a triangle-- so if you don't find a triangle, then we return that it's triangle-free. Otherwise we return that it is epsilon far from triangle-free.

So that's the algorithm. So it's a very intuitive algorithm, but why does it work? So we want to know that, indeed, somebody gives you one of these two possibilities. You run that algorithm, you can succeed with high probability. Question?

AUDIENCE: [INAUDIBLE]

YUFEI ZHAO: So let's talk about why this works. So theorem, for every epsilon, there exists a c such that algorithm succeeds with probability bigger than 2/3, and 2/3 can be any number. So any number that you like because you can always repeat it to boost that constant probability.

So there are two cases. If g is triangle-free, then it always succeeds. You'll never find this triangle, and it would return triangle-free. On the other hand, if g is epsilon far from triangle-free, then triangle removal lemma tells us that g has lots of triangles. Delta n cubed triangles.

So if we sample c being, let's say, 1 over delta times-- delta here is a function of epsilon from the triangle removal lemma. So we find that the probability that the algorithm fails is at most-- so you have a lot of triangles. So very likely you will hit one of these triangles. So the probability that the algorithm fails is at most 1 minus delta n cubed divided by total number of triples raised to 1 over delta. And this is 1 minus at most 1 minus 6 delta raised to 1 over delta, and it's at most e to the minus 6. So less than 1/3 in particular. So this algorithm succeeds with high probability.

Now, how big of a c do you need? Well, that depends on the triangle removal lemma. So it's a constant. So it's a constant, does not depend on the size of the graph. But it's a large constant, because we saw in the proof of regularity lemma that it can be very large. But you

know, this theorem here is basically the same as a triangle removal lemma. So it's highly non-trivial if it's true. Even though the algorithm is extremely naive and simple.

I just want to finish off with one more thing. Instead of testing for triangle-freeness, you can ask what other properties can you test? So which graph properties are testable in default in that sense? So distinguishing something which has the property, so P versus epsilon far from this property P.

And you have this tester which is you sample some number of vertices. So this is called the oblivious tester. So you sample k vertices, and you try to see if it has that property.

So there's a class of properties called hereditary. So hereditary properties are properties that are closed under vertex deletion. And these properties are-- lots of properties that you're seeing are of this form. So for example, being H3 is this form being planar so this one being induced H3, so this one being three-colorable, being perfect, they're all examples of hereditary properties. Properties that if your graph is three-colorable, you take out some vertices, it's still three-colorable.

And all the discussions that we've done so far, in particular the infinite removal lemma. If you phrase it in the form of property testing given the above discussion, it implies that every hereditary property is testable. In fact, it's testable in the above sense with a one-sided error using an oblivious tester. One-sided error means that up there if it's triangle-free, then it always succeeds. So here one of the cases that always succeeds.

And the reason is that you can characterize a hereditary property by a curly H induced H3 for some curly H. Namely, you're putting everything into H that do not have this property. This is a possibly infinite set of graphs, and that completely characterizes this hereditary property. And if you read out the infinite removal lemma, it says precisely, using above this interpretation, that you have a property testing algorithm.