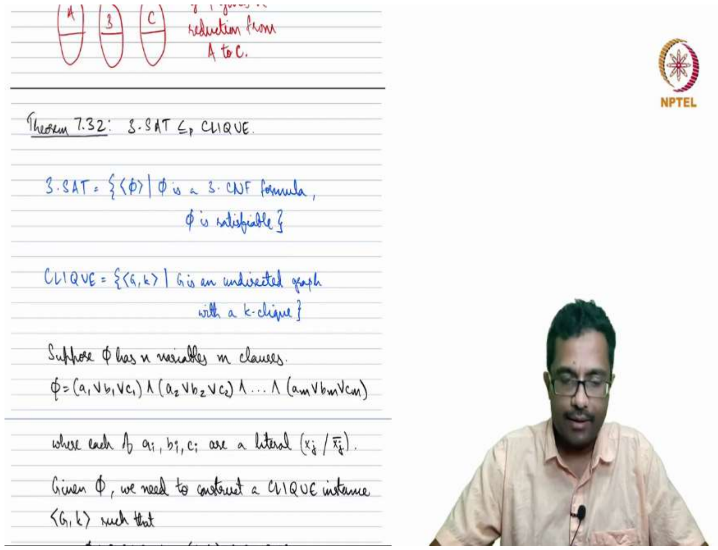
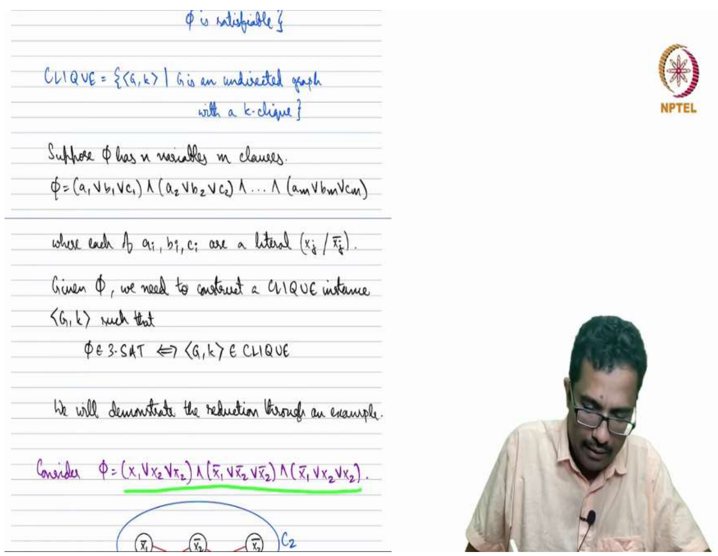
Theory of Computation
Professor Subrahmanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Polynomial Time Reductions - Part 2

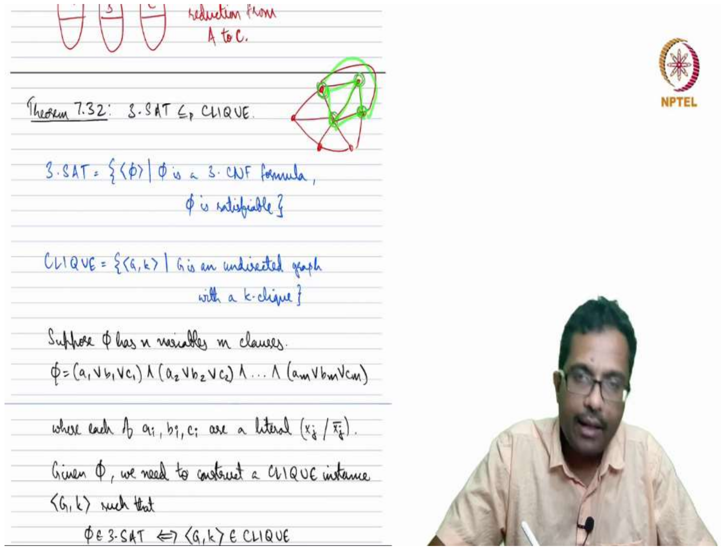
(Refer Slide Time: 00:17)

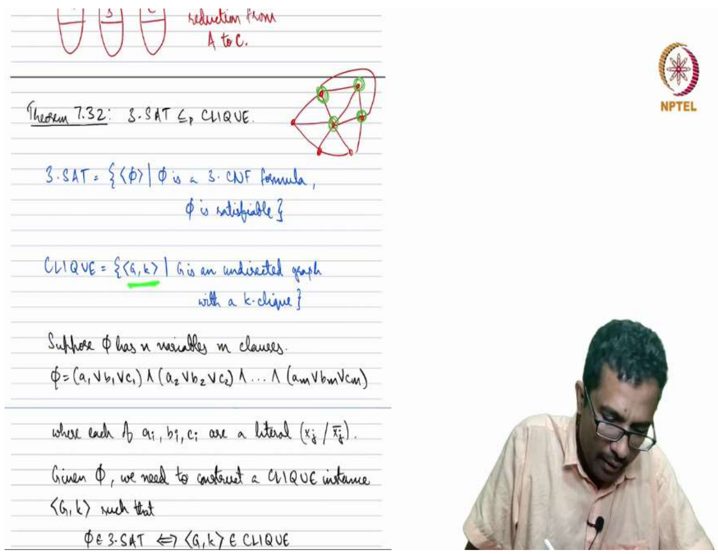


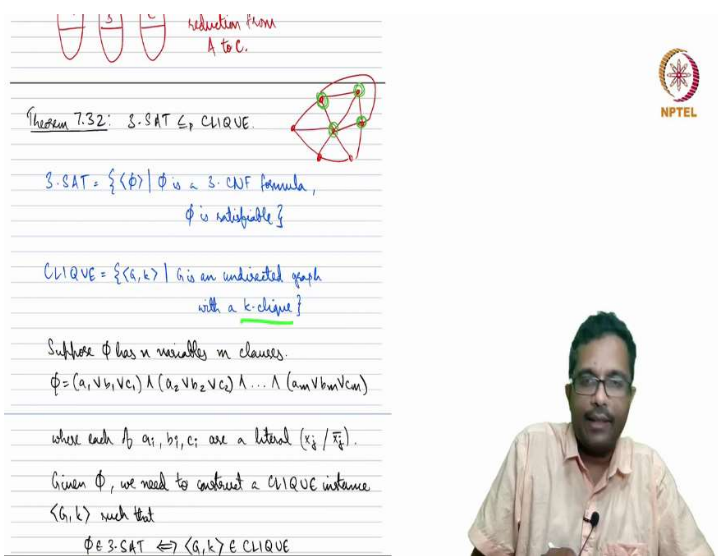


So, the next thing that I want to show is a proof that 3 SAT reduces to CLIQUE. So, 3 SAT we have seen it is a class of all Boolean formula, which are in 3 CNF form and are satisfiable, so 3 CNF means this, you have AND of clauses were each clause is an OR of literals. So, 3 SAT means it is a 3 CNF formula that is satisfiable.

(Refer Slide Time: 00:58)

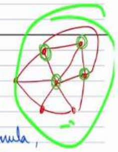






reduction from A to C.

Theorem 7.32: 3-SAT $\leq_p$ CLIQUE.



3-SAT = $\{\langle \phi \rangle \mid \phi$ is a 3-CNF formula, ϕ is satisfiable$\}$

CLIQUE = $\{\langle G, k\rangle \mid G$ is an undirected graph with a k-clique$\}$

Suppose ϕ has n variables m clauses.

$\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_m \vee b_m \vee c_m)$

where each of a_i, b_i, c_i are a literal (x_j / $\overline{x_j}$).

Given ϕ, we need to construct a CLIQUE instance $\langle G, k\rangle$ such that

$\phi \in$ 3-SAT $\Longleftrightarrow \langle G, k\rangle \in$ CLIQUE

reduction from A to C.

Theorem 7.32: 3-SAT $\leq_p$ CLIQUE.

3-SAT = $\{\langle \phi \rangle \mid \phi$ is a 3-CNF formula, ϕ is satisfiable$\}$

CLIQUE = $\{\langle G, k\rangle \mid G$ is an undirected graph with a k-clique$\}$

Suppose ϕ has n variables m clauses.

$\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_m \vee b_m \vee c_m)$

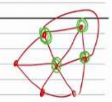
where each of a_i, b_i, c_i are a literal (x_j / $\overline{x_j}$).

Given ϕ, we need to construct a CLIQUE instance $\langle G, k\rangle$ such that

$\phi \in$ 3-SAT $\Longleftrightarrow \langle G, k\rangle \in$ CLIQUE

A to C.

Theorem 7.32: 3-SAT $\leq_p$ CLIQUE.



3-SAT = $\{\langle \phi \rangle \mid \phi$ is a 3-CNF formula, ϕ is satisfiable$\}$

CLIQUE = $\{\langle G, k\rangle \mid G$ is an undirected graph with a k-clique$\}$

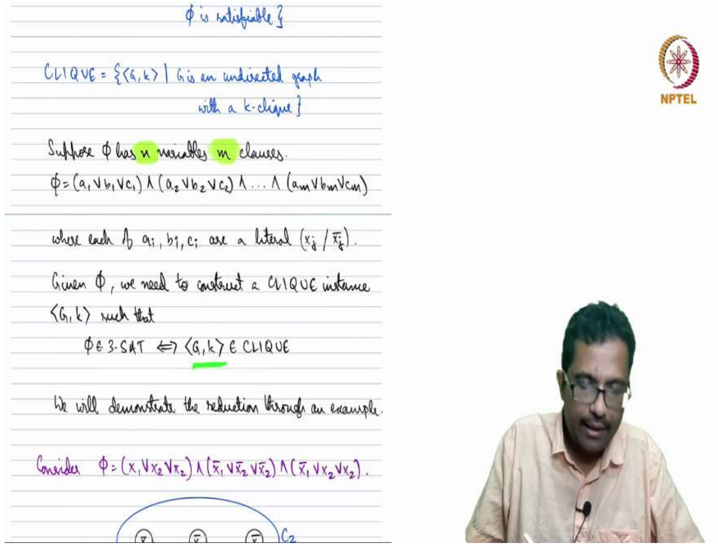
Suppose ϕ has n variables m clauses.

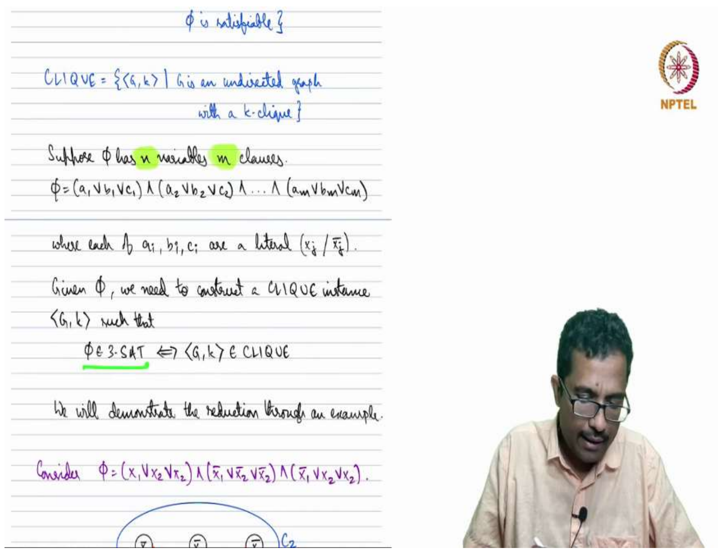
$\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_m \vee b_m \vee c_m)$

where each of a_i, b_i, c_i are a literal (x_j / $\overline{x_j}$).

Given ϕ, we need to construct a CLIQUE instance $\langle G, k\rangle$ such that

$\phi \in$ 3-SAT $\Longleftrightarrow \langle G, k\rangle \in$ CLIQUE





So, CLIQUE is a graph whose all the edges are adjacent. So, for instance, in this graph, the graph that I am just drawing in the corner here, there is a 4 CLIQUE, which of these 4 vertices constitute a 4 CLIQUE. Because if you take these circle vertices, you take any pair of them they are adjacent this and this is adjacent this and this any pair is adjacent.

So, there are 6 possible 1, 2, 3, 4, 5, 6, 6 possible adjacent is there and all 6 exist. So, which means this graph has a 4 CLIQUE, but it does not have it 5 CLIQUE, there are no set of 5 vertices that are adjacent to each other like this. So, the CLIQUE consists of a pair G and K, where G is an undirected graph with K CLIQUE.

So, if this graph that is that we have drawn over here, is given with the number 4, it will be yes instance with K equal to 4. But if it is given with the number 5, it will be no instance because there is no CLIQUE of size 5, if it is given with the number 3, again, it is a yes instance, because you could take these 3 vertices, and that is a 3, CLIQUE. So that is the problem of CLIQUE.

So given a graph G and a number K, you are asking whether there is a K CLIQUE. And 3 SAT I already mentioned. So, it may be a bit surprising that one of them is a problem on Boolean formulas and satisfiability. The other one is something about graphs and having some number of vertices that are adjacent to each other.

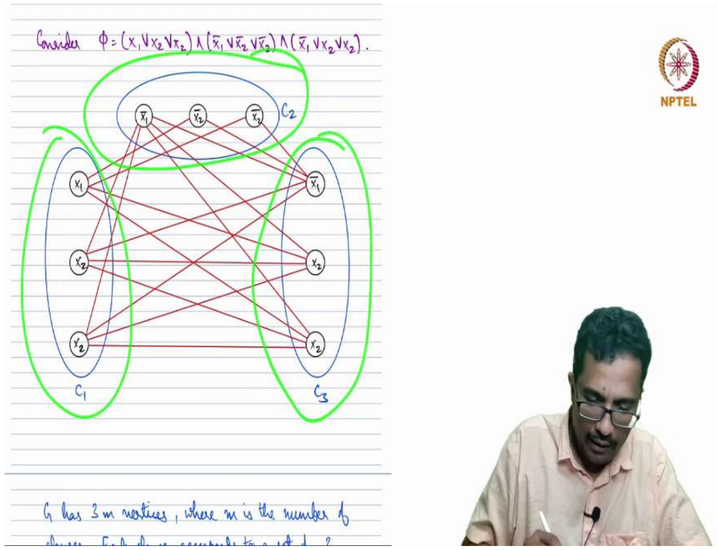
How is it possible that you can transform 1 to other? So that may seem very surprising, so it is not that hard. So, we will see the reduction and maybe another next 15 minutes. So, before that, we will just set up some basic notation. So, let n be the number of variables of the, so what do we have to do? We have to given a 3 SAT instance we have to obtain a CLIQUE instance.

So, if the given 3 SAT instance is satisfiable, then the output CLIQUE instance should be a yes instance, if the given 3 SAT instance is not satisfiable, the output CLIQUE instance should not be yes instance. So, it should be a pair G and K where G does not have a K clique So, how can we do this? So, suppose, we say that n is the number of variables of the given formula, and m is the number of clauses.

So remember, the Boolean formula is 3 CNF form, where we have AND of clauses, where each clause is an OR of literals. And, and further, each clause is an OR of 3 literals. So you will have $(a1 \vee b1 \vee c1) \wedge (a1 \vee b1 \vee c1) \wedge \ldots \wedge (am \vee bm \vee cm)$. So, this is how the how the formula will look like. Where each of $a1, b1, c1$ maybe $x1$, $\overline{x1}$, $x2$, $\overline{x2}$, $x3$, $\overline{x3}$ anything or $xj, \overline{xj}$ in general.

So, given this, we need to construct an output something of the form $< G, K >$, such that this formula is satisfiable if and only if this $< G, K >$ is yes instance of CLIQUE or in other words, G has a K clique. So it is easiest to explain this reduction through an example.

(Refer Slide Time: 04:48)



And since there are I want to draw the picture and explain this example, I am going to resort to a very simple, simplistic kind of formula, but it should be very easy to see how we can deal with a bigger formula as well. Just that the figure will get much more messier. But just for the purpose of understanding this reduction, the example that we are taking is representative enough.

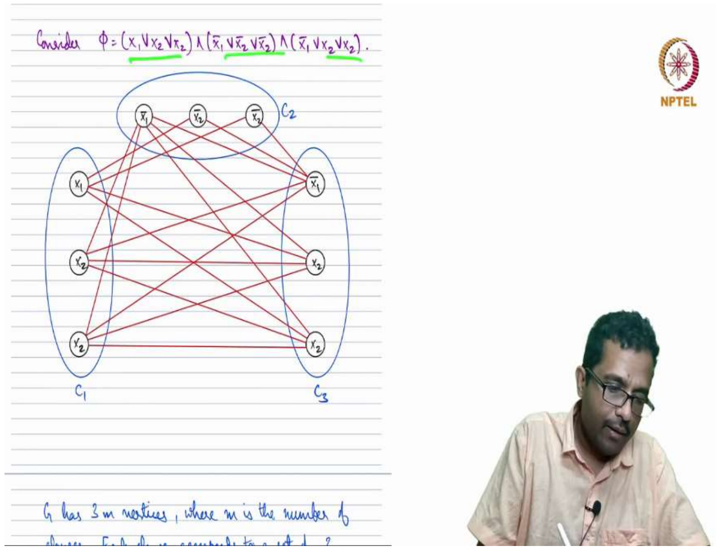
So, suppose φ

$$\varphi = (x1 \ \vee x2 \ \vee x2) \wedge (\overline{x1} \vee \overline{x2} \vee \overline{x2}) \wedge (\overline{x1} \vee x2 \vee x2)$$

So, we have only 2 variables $x1$ and $x2$. And we have only 3 clauses. So, n is 2, m is 3. And this is a property of 3 SAT, every clause has 3 literals,

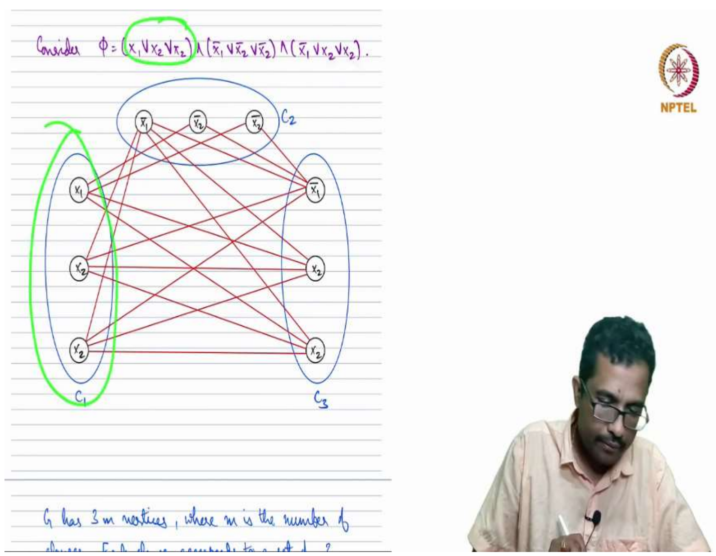
So, given this, our goal is to produce G and m such that this is satisfiable if and only if G has a m CLIQUE. So, let us see how it happens. So, first, the graph G looks like this. So, what do we have here? So, we have 3 groups of vertices. So, the group by group, this is group 1, this is group 2, and this is group 3, and each group has 3 vertices each.

(Refer Slide Time: 06:24)

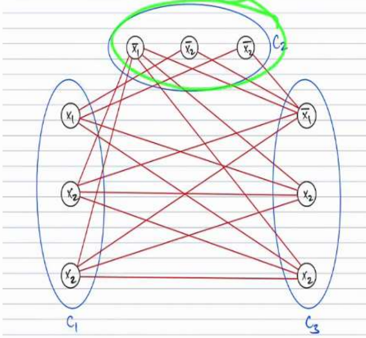


So, basically, we have 3 groups of vertices, because we have 3 clauses in the formula. So, here we have 3 clauses.

(Refer Slide Time: 06:27)

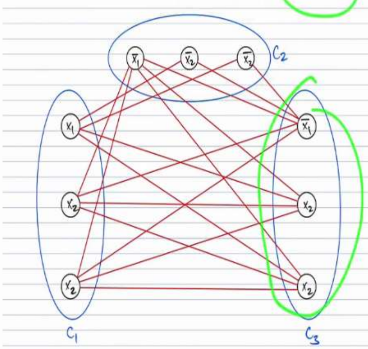


Consider $\phi = (x_1 \vee x_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$.

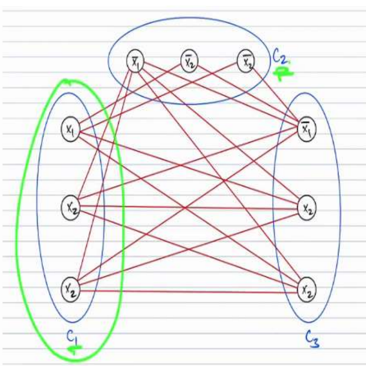


G has 3m vertices, where m is the number of

Consider $\phi = (x_1 \vee x_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$



G has 3m vertices, where m is the number of

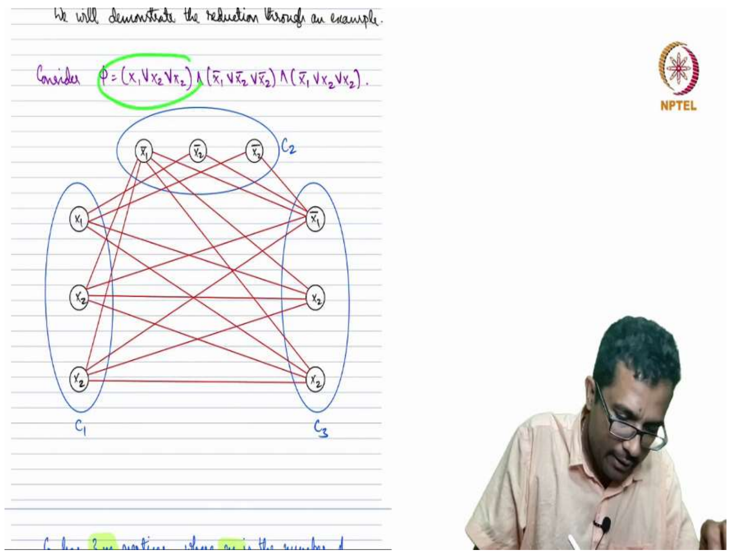


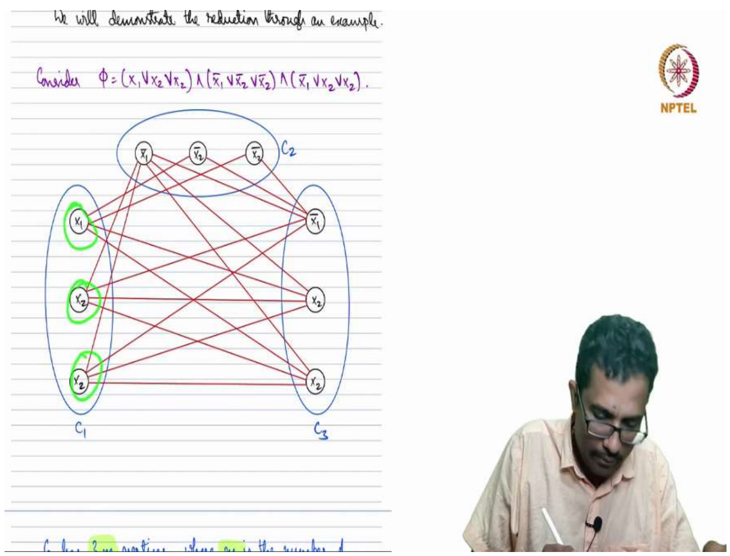
G has 3m vertices, where m is the number of clauses. Each clause corresponds to a set of 3 vertices.

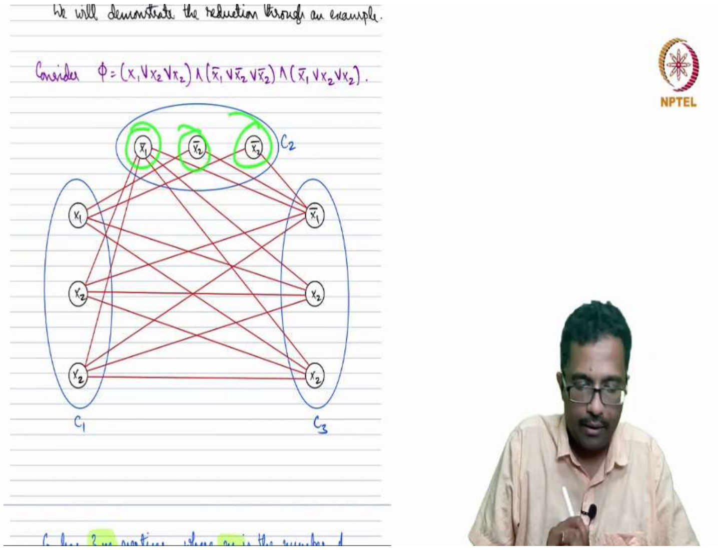
So, the clause 1 corresponds to this group, clause 2 corresponds to this group and clause, clause 3 corresponds to this group. So, the graph will have 3m vertices, basically m groups of vertices, and each group contains 3 vertices. So, the graph will have 3m vertices.

So, here, m is also 3, so, it is 9 vertices here, it has 3m vertices, where m is the number of clauses and each clause corresponds to a group of 3 vertices, which I have indicated here, this vertex corresponds to clause 1. Clause 1, clause 2, clause 3. So, that are the vertices and it is also labelled as such.

(Refer Slide Time: 07:18)

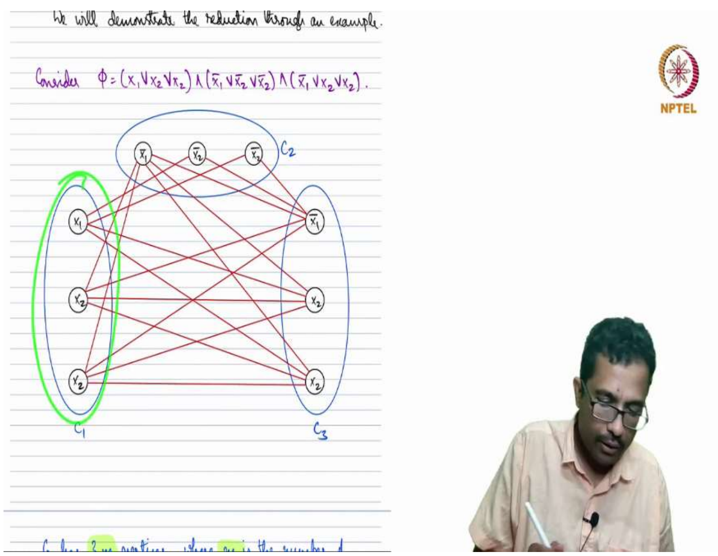


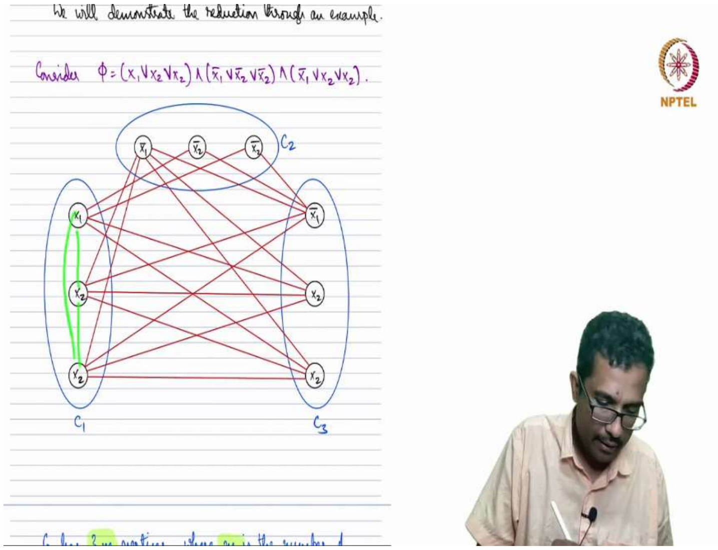


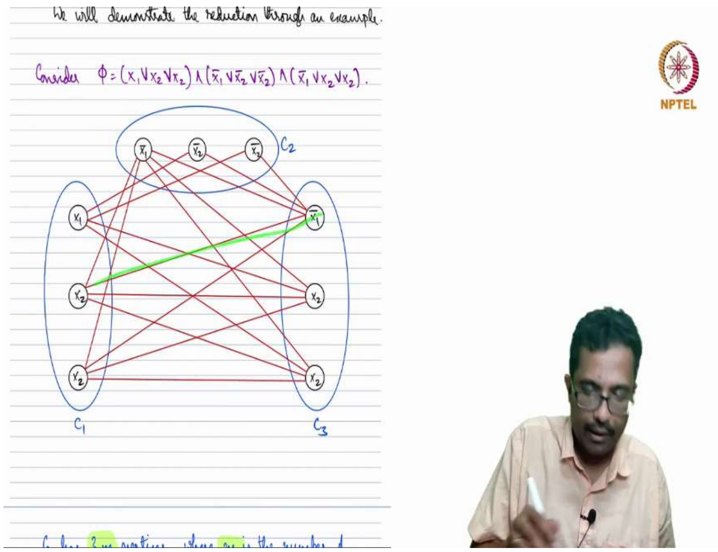


So, clause 1 is $x1 \lor x2 \lor x2$. Clause 2 is $\overline{x1} \lor \overline{x2} \lor \overline{x2}$. Similarly, for clause 3. So, I have told the vertices. Now, how do we define the edges?

(Refer Slide Time: 07:57)

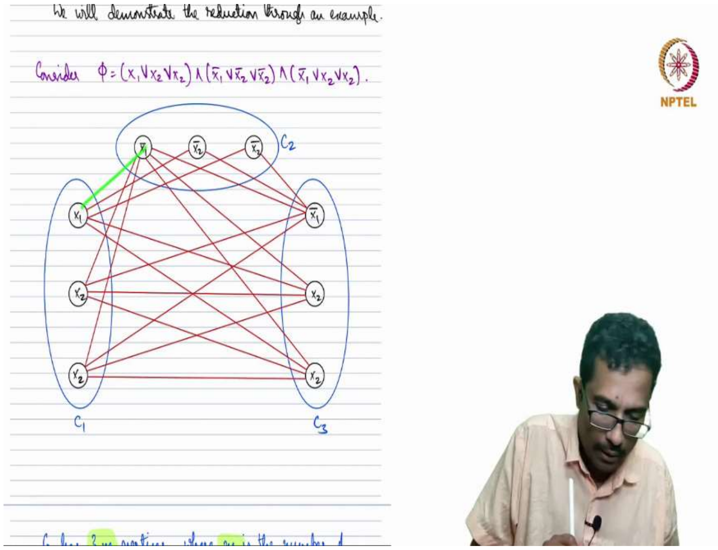


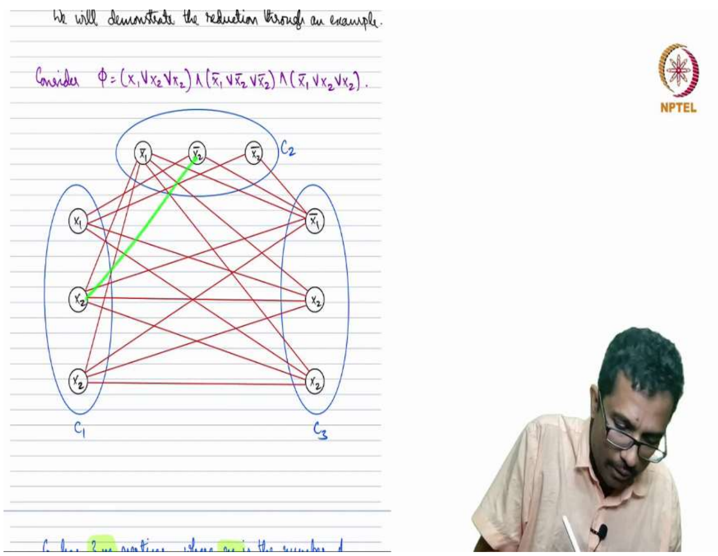


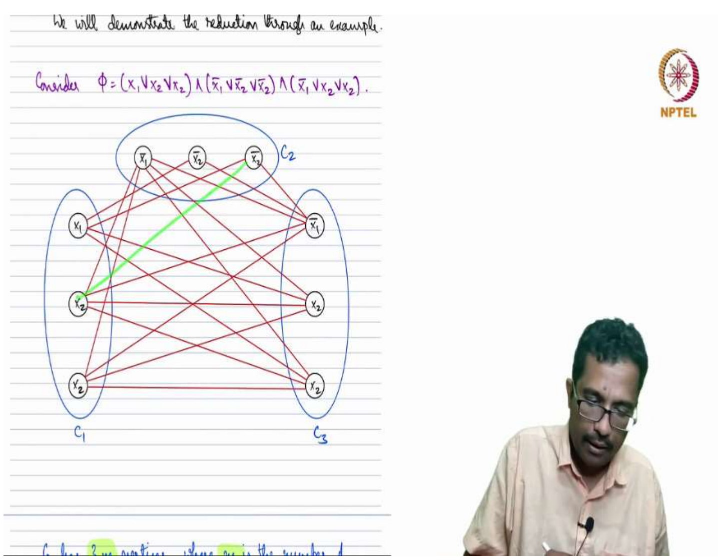


So, edges are very simple, we want to connect all possible vertices, but with 2 exceptions. We do not connect 2 vertices if one of the 2 cases apply. So, one is that within a group, we do not have any internal edges. So, we do not have edges like this. So, in that case, we do not put an edge. So, if you see all the edges are across groups, nothing within the group that is case 1.

(Refer Slide Time: 08:19)

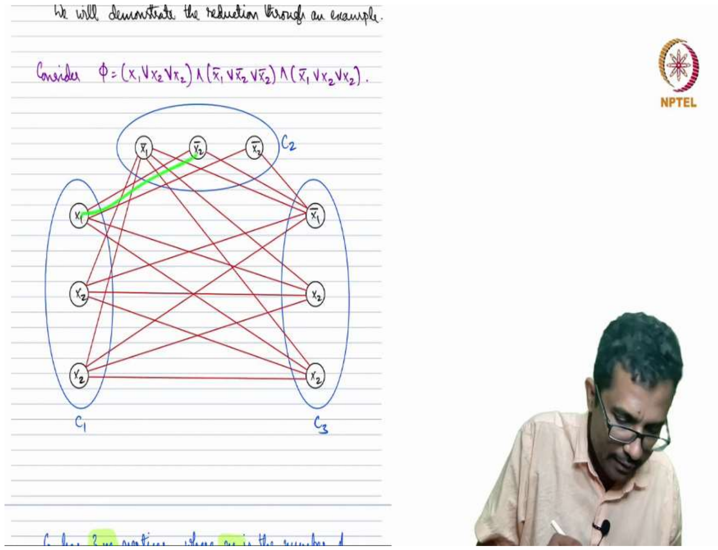


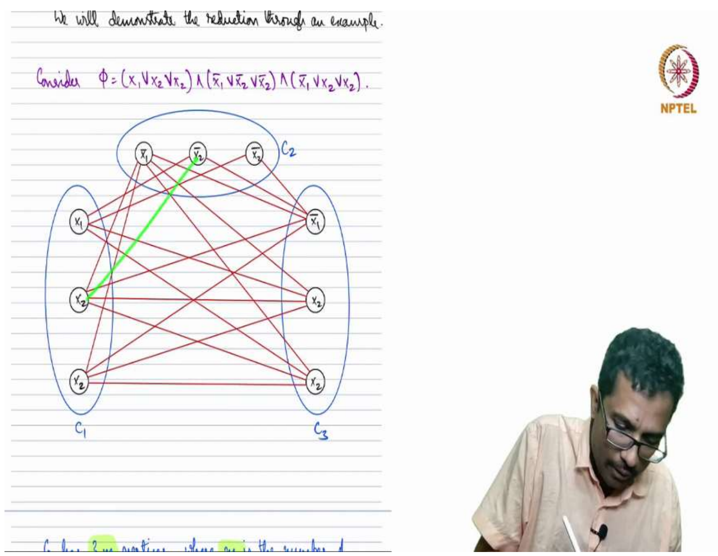




Second is that we do not have edges from $x1$ to $\overline{x1}$ and $x2$ to $\overline{x2}$. So, basically for any variable xi, we do not have edges from xi to $\overline{xi}$. In other words, $x1$ and $\overline{x1}$ are directly conflicting literals, $x1$ and $\overline{x1}$ cannot both be true. So, corresponding to that, I do not want to put an edge between these conflicting things. So, for any i, xi and $\overline{xi}$ cannot be adjacent.

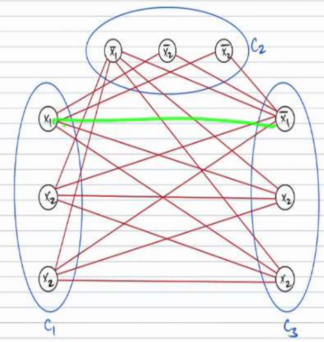
(Refer Slide Time: 08:52)





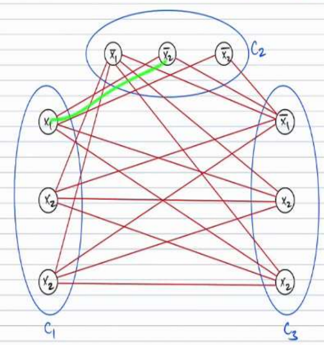
We will demonstrate the reduction through an example.

Consider $\phi = (x_1 \vee x_2 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$.



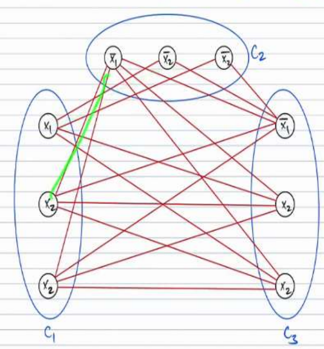
We will demonstrate the reduction through an example.

Consider $\phi = (x_1 \vee x_2 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$.



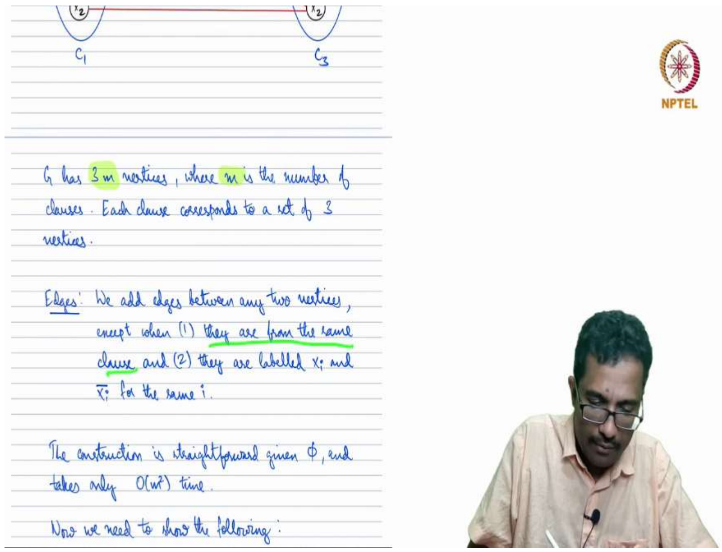
We will demonstrate the reduction through an example.

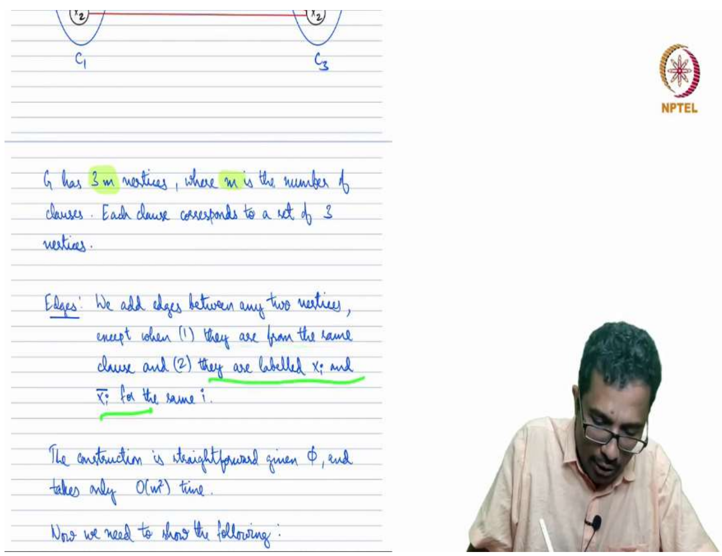
Consider $\phi = (x_1 \vee x_2 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$.

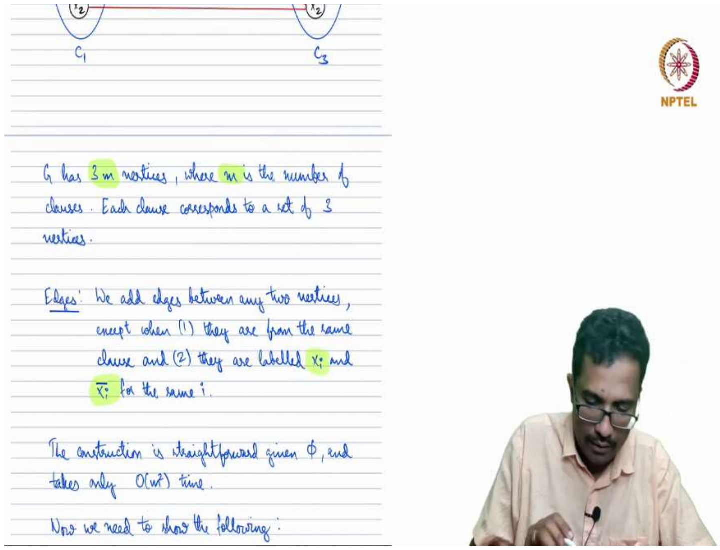


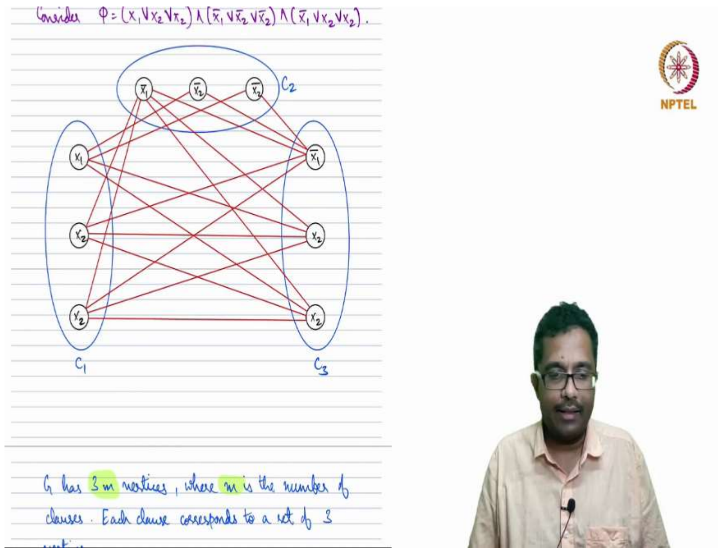
So, here $x1$ and $\overline{x1}$ cannot be adjacent, $x2$ and $\overline{x2}$ cannot be adjacent. But $x2$ and $\overline{x1}$ can have an edge no problem, because there is no issue with that $x2$ and $\overline{x1}$ can also have an edge, but for the same i, xi and $\overline{xi}$ cannot have an edge.

(Refer Slide Time: 09:19)





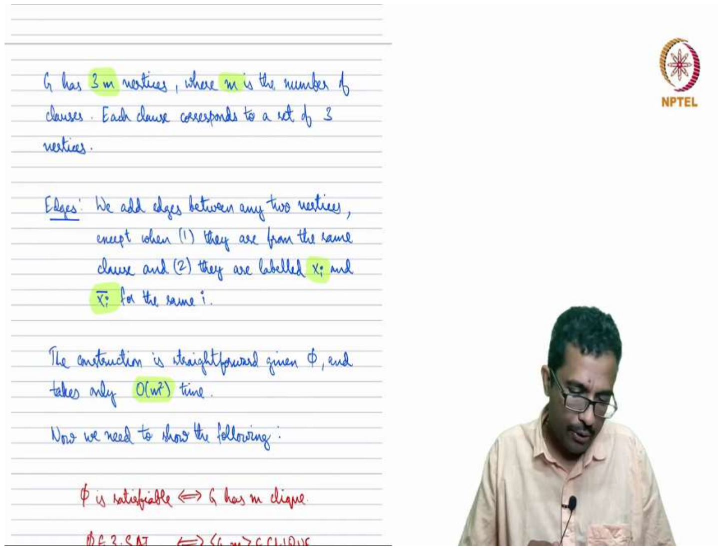




So, this is the same thing that I said here, we add edges between any 2 vertices, except when they are from the same clause. And they are labelled xi and $\bar{xi}$ for the same i. And everything else is here, you can just have a look everything else is here. And it is it should be evident that the construction is straightforward.

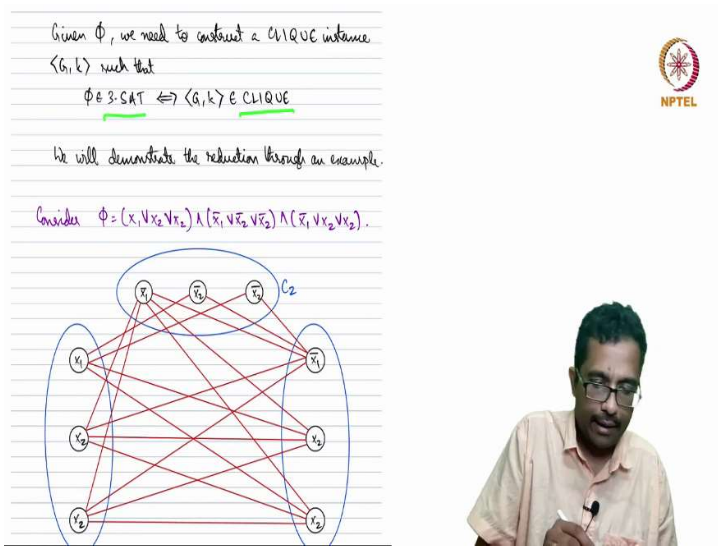
So I have 3m vertices. And I connect them by just doing this. So I have these groups. So I can have a for-loop, which outputs the edges, which output the adjacency matrix or something of the graph. And we can have maybe at most $(3m)^2$ edges. So, $9m^2$.

(Refer Slide Time: 10:05)



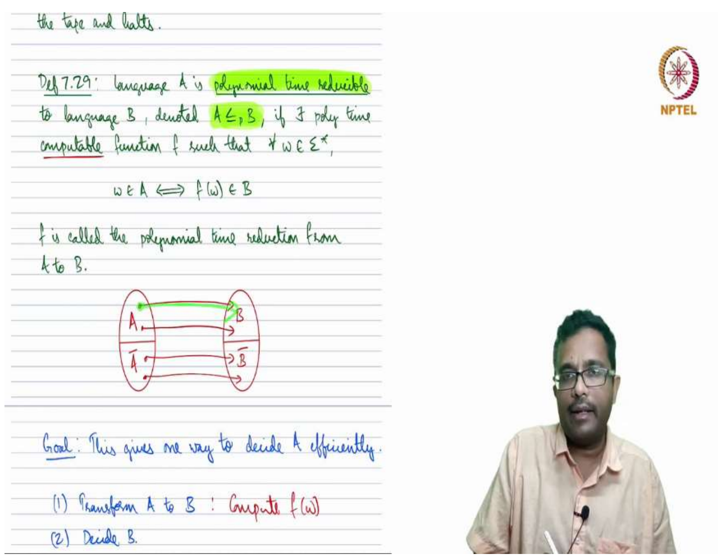
So, the construction is straightforward, the running time of this construction is order m^2 . So, where order m^2 is the size of the output and that is pretty much the running time the running time is not significantly more.

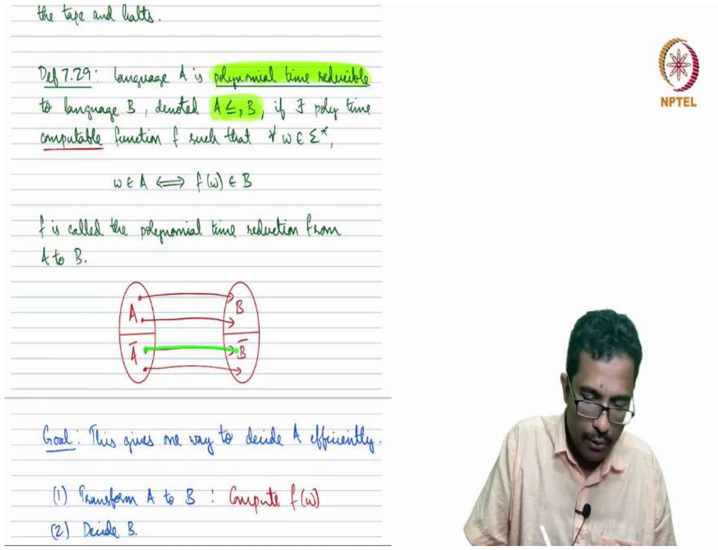
(Refer Slide Time: 10:22)

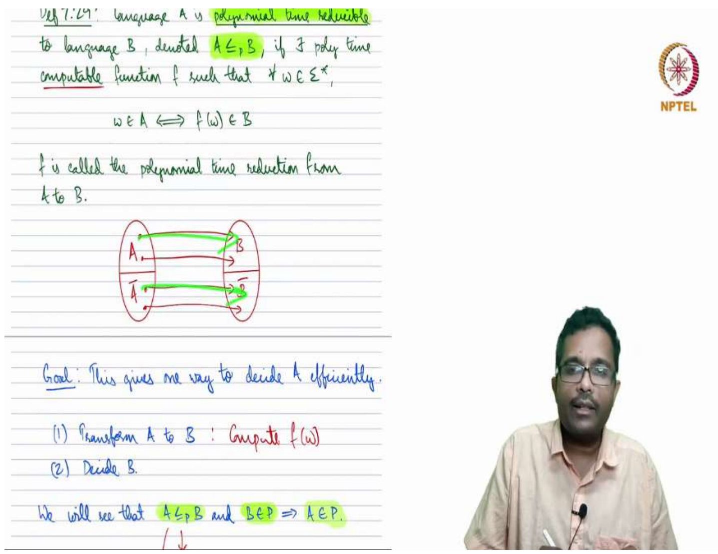


So, like in any reduction, we have to show 2 things, one is that given the 3 SAT instance, we have to produce a CLIQUE instance in polynomial time. So, that is already shown, because I said the construction is straightforward, and I explained the construction. So, given the formula we can easily produce the graph.

(Refer Slide Time: 10:52)



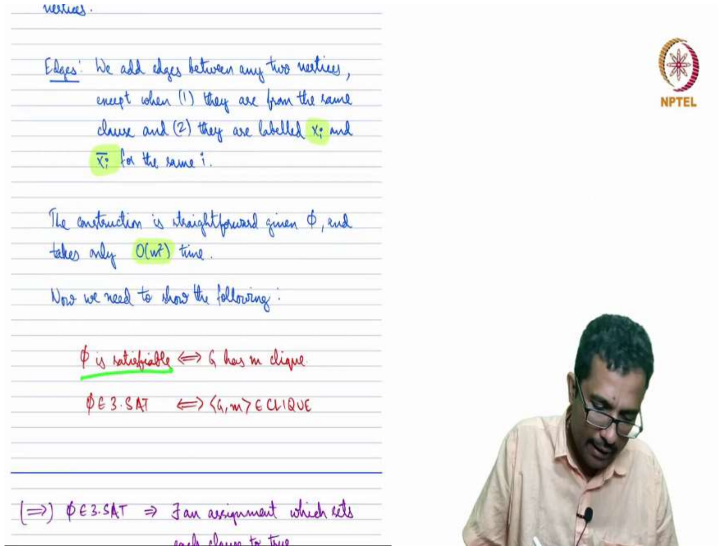


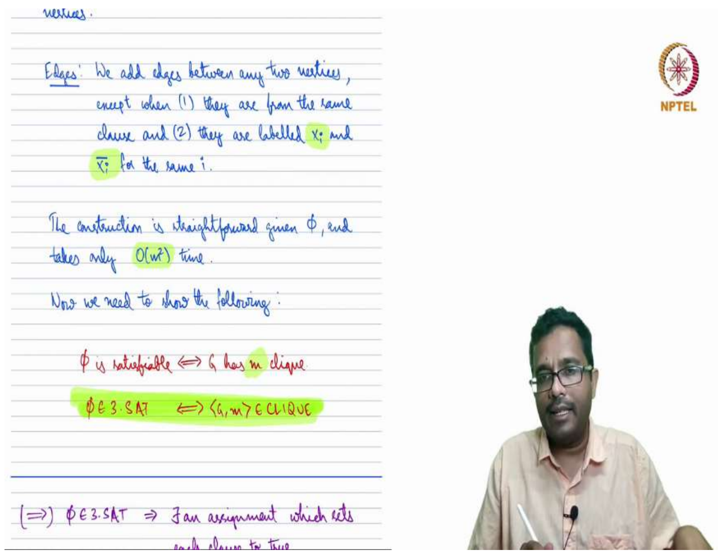


Now, the second thing to show is this thing that any member from A goes to a yes instance of B, anything from $\bar{A}$ goes to $\bar{B}$ or in our case, any formula that is satisfiable gives us $< G, K >$ where G has a K clique. Any formula that is not satisfiable gives us a $< G, K >$ where G does not have a K clique.

So, if φ is satisfiable, we get $< G, K >$ where G with a K clique, if φ is not satisfiable we get $< G, K >$ where G does not have K clique, this is the thing that we need to show.

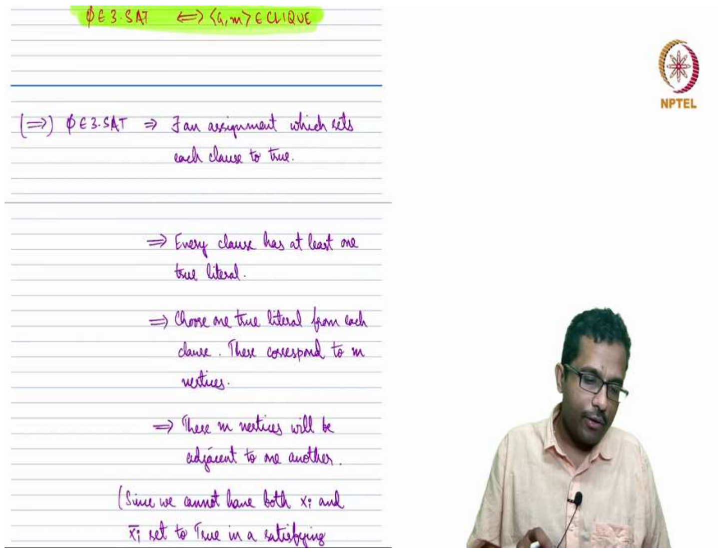
(Refer Slide Time: 11:37)

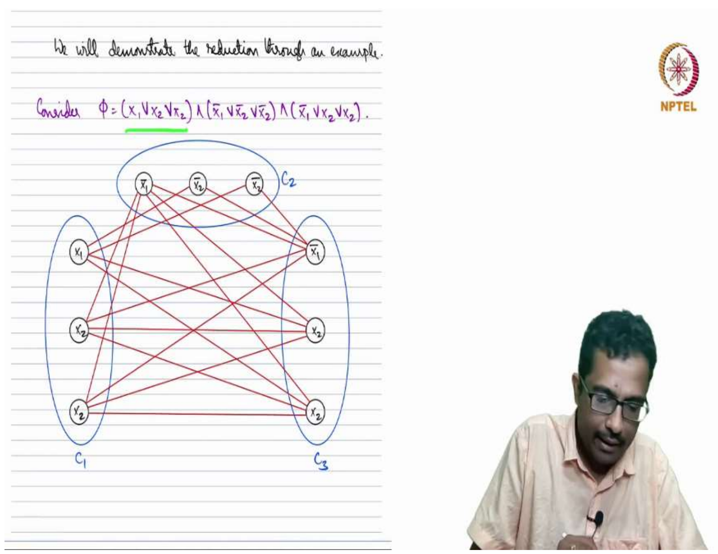


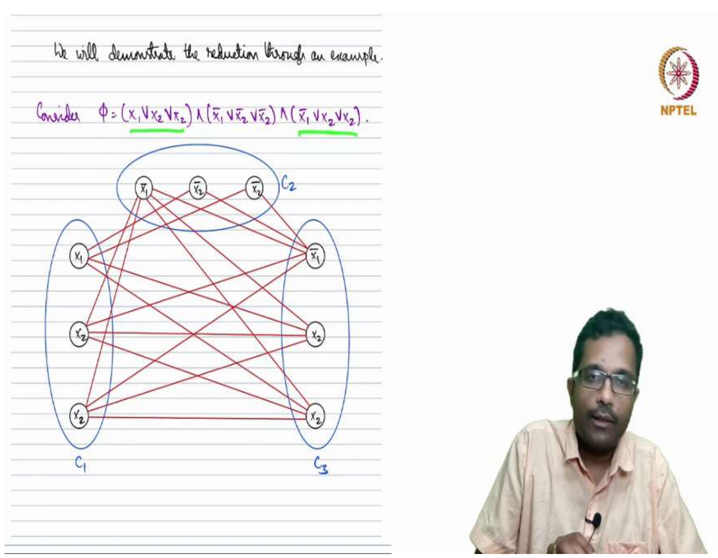


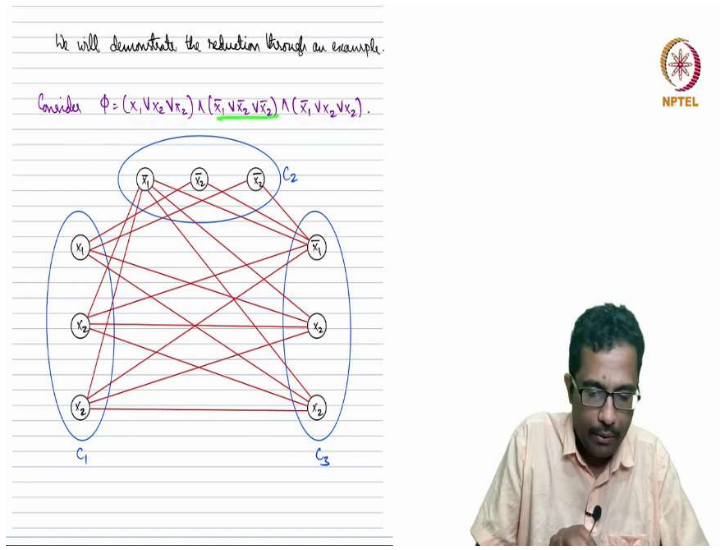
So, φ is satisfiable if and only if G has a K clique, so our K is also going to be m. So, this is our claim φ is in 3 SAT, if and only if $< G, m >$ is in CLIQUE. So, whenever φ is in 3 SAT, G will have an m CLIQUE, whenever φ is not satisfiable G will not have an m CLIQUE. So that is a claim. Let us see why this is true.

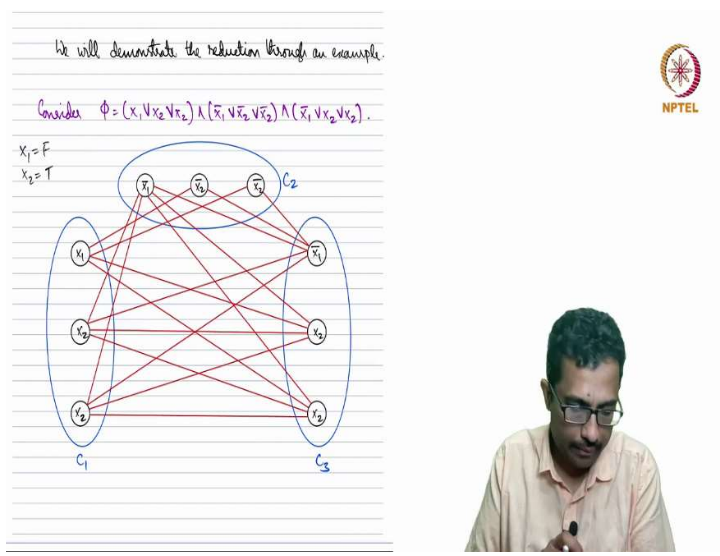
(Refer Slide Time: 12:10)







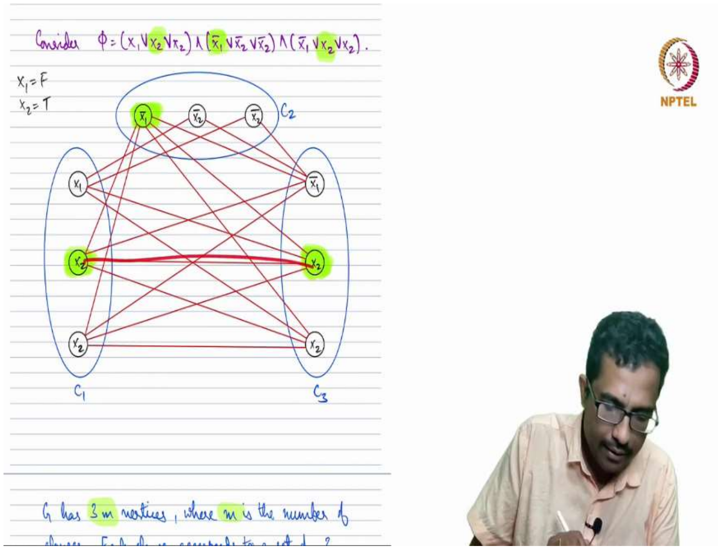


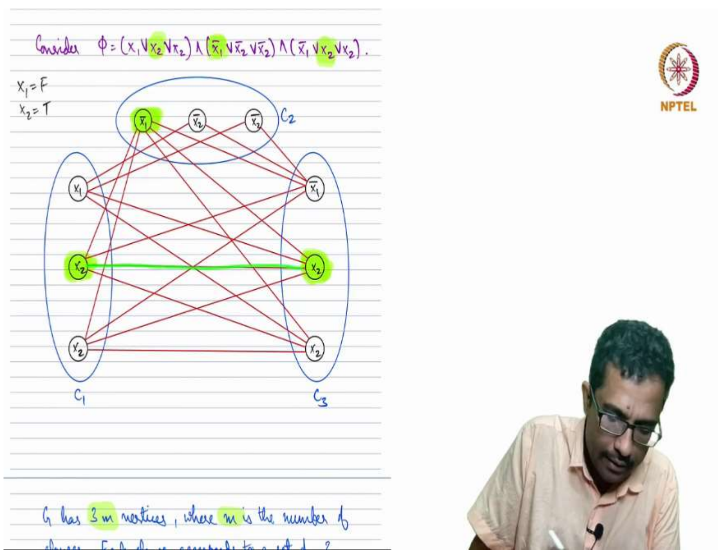


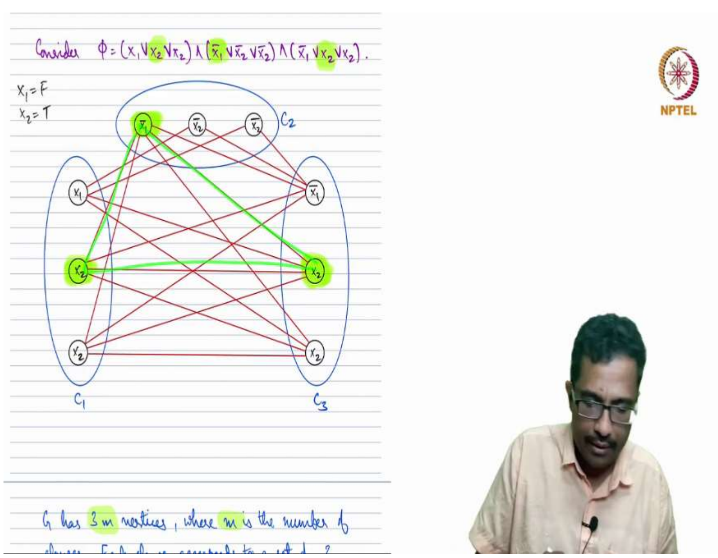
So, suppose φ is in 3 SAT. So this formula is in 3 SAT actually, because it is a very simple formula by making x1 true you can set you can make the first clause satisfied. And by making x2 false, no, sorry, you have to make x2 true to make the first and third clause satisfied. And you can make x1 false, this is satisfiable. So, x1 equal to false and x2 equal to true is a satisfying assignment.

So x2 equal to true satisfies the first and third clauses. x1 equal to false satisfies the second clause. So we have to show that φ is in 3 SAT if and only if G has an m CLIQUE. So let us assume that φ is in 3 SAT, and then show that G has an m CLIQUE. Later, we will show the reverse direction. So, φ is in 3 SAT implies that there is an assignment which sets each clause to true which means every clause has 1 true literal.

(Refer Slide Time: 13:54)







And now, so let us see here. So here, every clause has 1 true literals. So here for this satisfying assignment x2 is a true literal in the first clause, $\overline{x1}$ is a true literal in the second clause, and let us say x2 is the true literal in the third clause. So, the highlighted ones are the 2 literals.

And what we do is choose the same literals from the graph. So here we choose x2 from the first clause, $\overline{x1}$ from the second clause and x2 from the third clause. These are the true literals and the claim is that these 3 will be adjacent. Why is that? So, in the figure, you can see that they are indeed adjacent.

But why is why are they adjacent in general? They are adjacent in general because first of all, they are in 3 clauses. So, they are not in the same group. So, we do not put edges only if 2 conditions are satisfied or only for 2 reasons. One is if they are in the same group and two is if they are of the form x1 and $\overline{x1}$ or x2 or $\overline{x2}$ and so on.

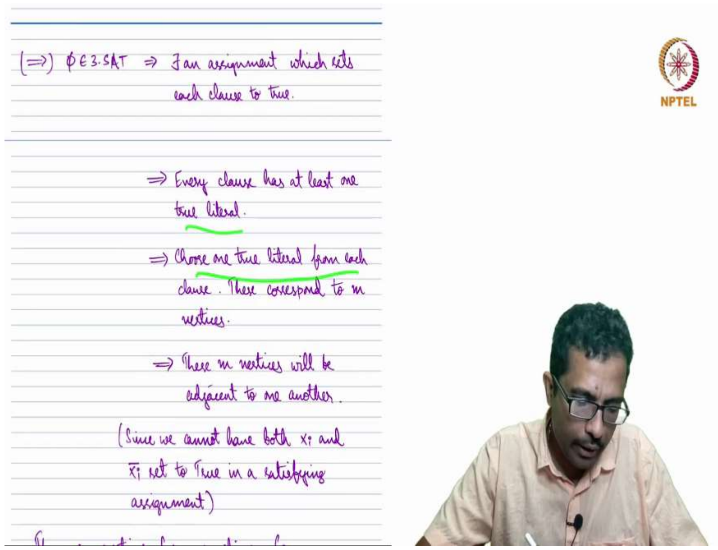
So, none of them are in the same group, because we picked each 1 from a different clause. So, the only way the there will not be an edge between two of these, the only way there can be an absence of an edge between two of these is if they are of the form xi and $\overline{xi}$.

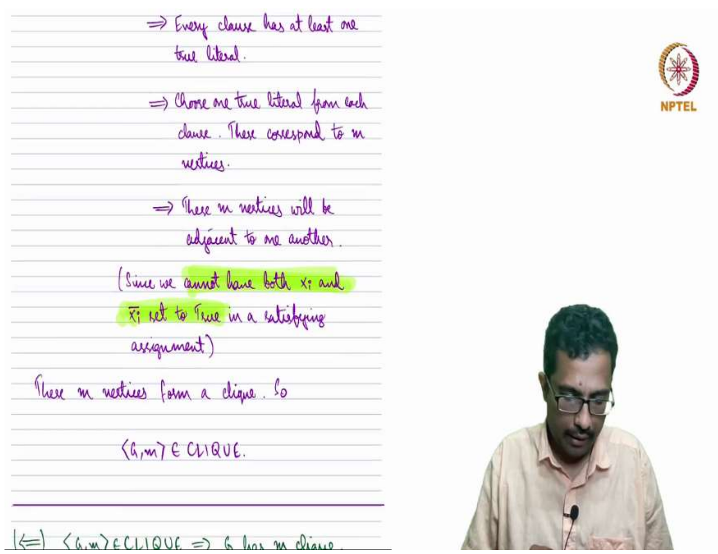
But then, we cannot have xi and $\overline{xi}$ because we pick true literals from each clause. So, if our assignment xi was true, then xi will be the true literal, $\overline{xi}$ will not be true literal. So, hence, if we picked x2 here, x2 compliment will not be true. So, we cannot pick xi and $\overline{xi}$ from 2 different clauses. So that is the other way in which there may not be an edge.

So, there will be an edge between any pair of vertices here because, firstly, they are all from separate groups. And secondly, we do not have any pair xi and $\overline{xi}$ because if it was there, then it will be a contradiction to the way we chose these variables. So these literals were chosen since they are all true.

So, if xi is true, $\overline{xi}$ cannot be true, if $\overline{xi}$ is true, then xi cannot be true. So, both of them will not be true together. So, either for a variable xi, the xi was a true variable and only xi was picked or $\overline{xi}$ was picked, but not both. So, that means that the vertices pick from the 3 groups or from the m groups in general. So, in general, we have m clauses, all of them will be adjacent to each other, and that gives us an m CLIQUE. And here we have we have a 3 CLIQUE because m is 3.

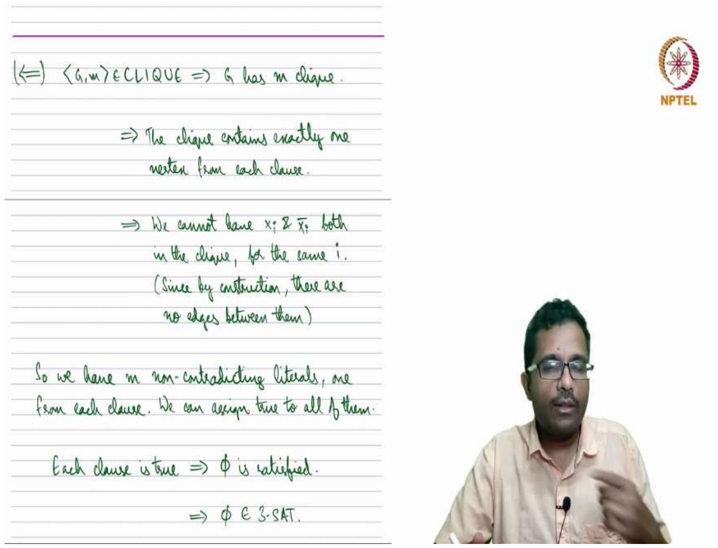
(Refer Slide Time: 17:03)

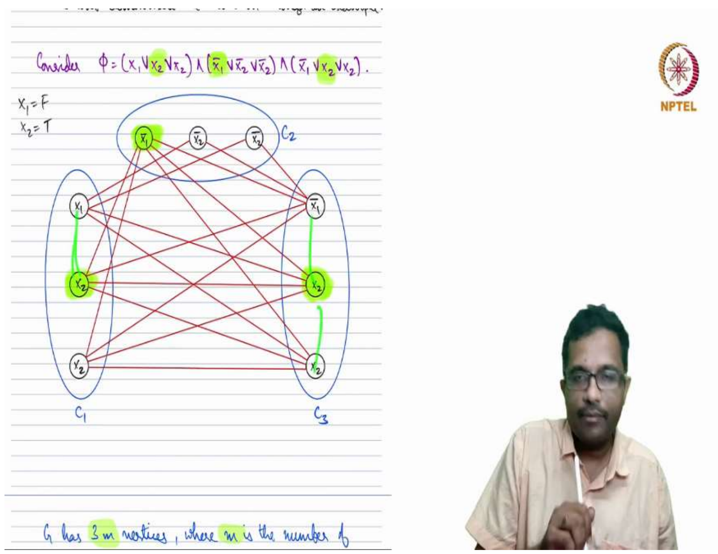


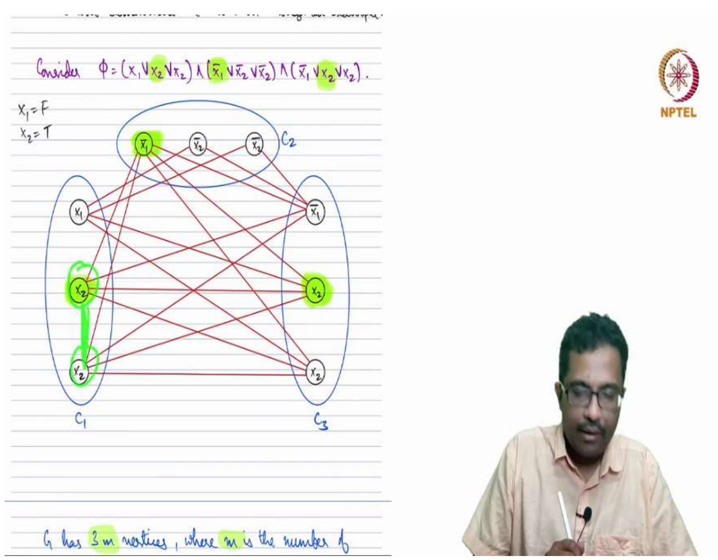


Same thing here. So, every clause has at least one true literal. And so, we choose 1 true literal from each clause and these correspond to m vertices, because there m clauses and these m vertices will be adjacent, because we cannot have both xi and $\overline{xi}$ set to true. This is a key thing we cannot have both xi and $\overline{xi}$ set to true. Hence, the graph has an m CLIQUE.

(Refer Slide Time: 17:31)





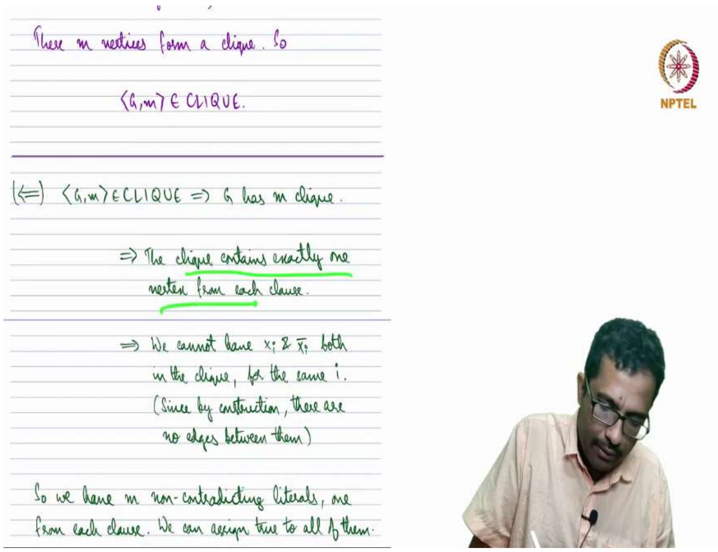


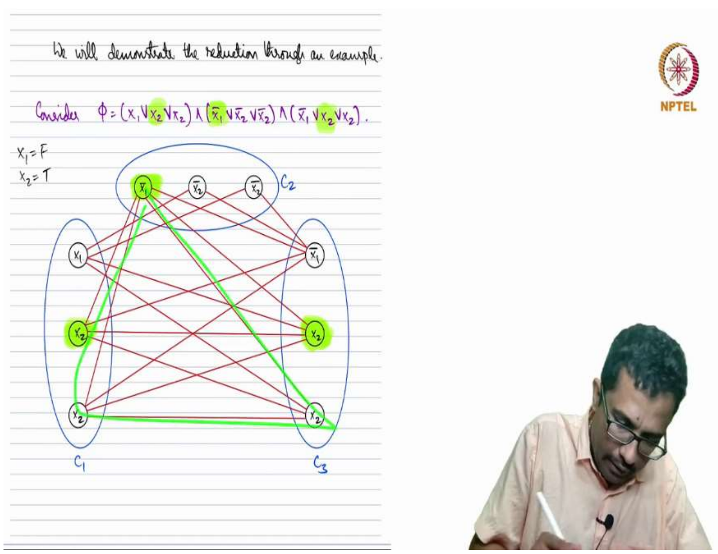
Now, the reverse direction suppose the graph has an m CLIQUE, then we have to show that the formula is satisfied. So, suppose the graph has an m CLIQUE. Now we know that there are m groups and we know that we do not have any edges within the same group these edges do not exist.

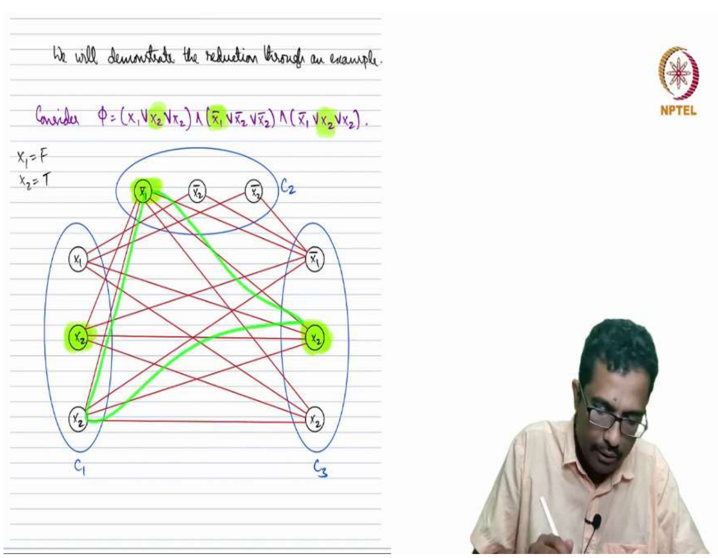
So, which means I can have at most 1 vertex in this m CLIQUE from each group, I cannot have more than 1 vertex, because if I pick 2 of these vertices, then this is not going to be a CLIQUE. So, I can have at most 1 vertex from each group, but then there are only m groups, which means I have to have 1 vertex from each group.

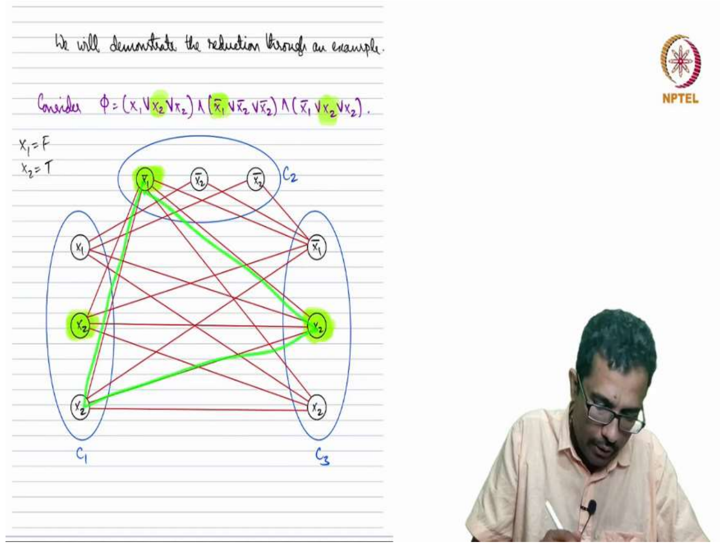
So, if this graph has a 3 CLIQUE, that it has to be 1 vertex from each of these 3 groups, because inside I am not putting internal edges, so we cannot have 2 from 1 group. So, it has to be distributed 1111. So, G has an m CLIQUE means that each vertex in the CLIQUE comes from a distinct group, which means each group contributes exactly 1 vertex.

(Refer Slide Time: 18:54)









So, the CLIQUE contains exactly 1 vertex from each clause. And we know that we cannot have xi and $\overline{xi}$ in the CLIQUE, because if I picked for some i, xi and $\overline{xi}$ from 2 separate groups, we know there is no edge between them that is the rule that we have. So, if for any variable we picked xi then $\overline{xi}$ was not picked, if any for any variable, we picked $\overline{xi}$, then xi was not picked.

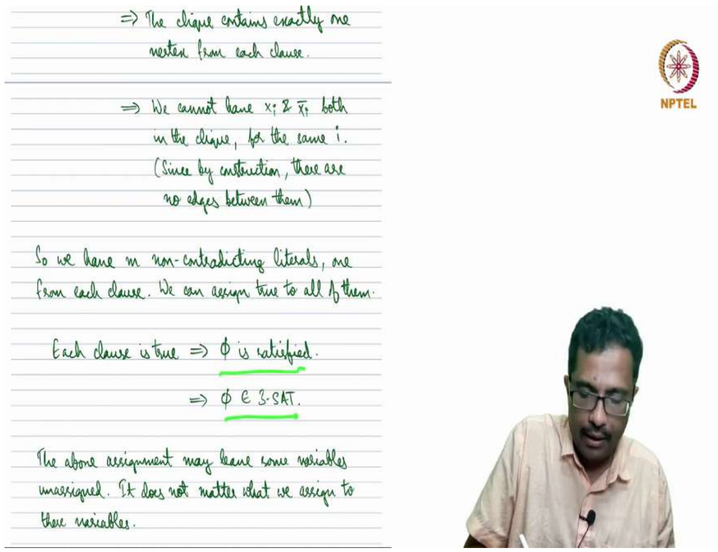
So, because we did not put edges between them, so, what we have is some set of m literals. So, these m vertices that form a CLIQUE correspond to some set of m non contradicting literals. So here, if you look at the CLIQUE, so maybe another CLIQUE that we can identify is this one. But in this case, there is only 1 satisfying assignment.

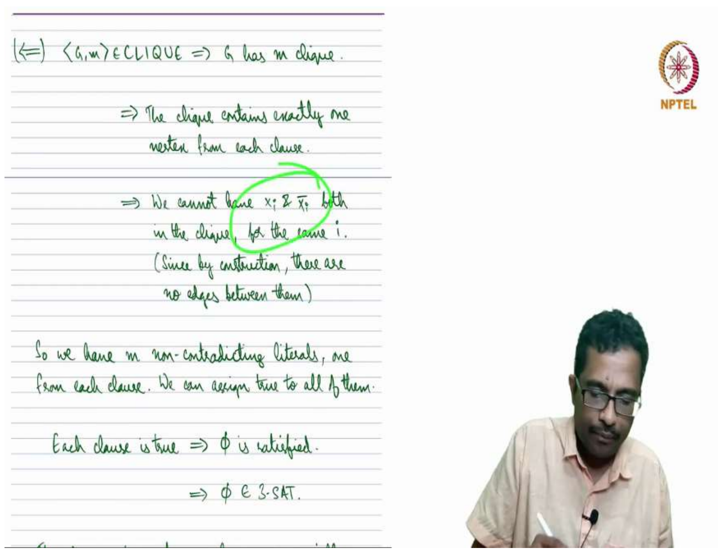
So all the clicks that you identify will be of the form $\overline{x1}$, x2, x2, because that is only satisfying assignment. So $\overline{x1}$, x2, x2. So, the thing is that we do not have $x1$ and $\overline{x1}$, we only have 1 of them. And same, we do not have both $x2$ and $\overline{x2}$, we only have 1 of them in the CLIQUE. So, for some variables, for some xi we have xi, for some xi, we have $\overline{x1}$.

So, we have some, let us say for $x1$, $x1$ is picked, for $x2$, $x2$ is picked, for x3, let us say $\overline{x3}$ is picked. Now, what we do is we set all these to be true. So, if xi is picked, we make xi to be true, if $\overline{xi}$ is picked, we make $\overline{xi}$ to be true. In other words, we make xi to be false. So, we make all the selected literals to be true.

And this will be completely fine because we do not have conflicting literals, if xi was set to true, $\overline{xi}$ will never have been set to true, because these 2 cannot be part of the same CLIQUE. So, we assign true to all the m non contradicting literals. And by choice we, so by the choice of the literals, we picked exactly 1 true literal from each clause or the CLIQUE contained 1 vertex from each clause, which we set to true. So, we have 1 true literal from each clause. So, each clause is satisfied.

(Refer Slide Time: 21:41)

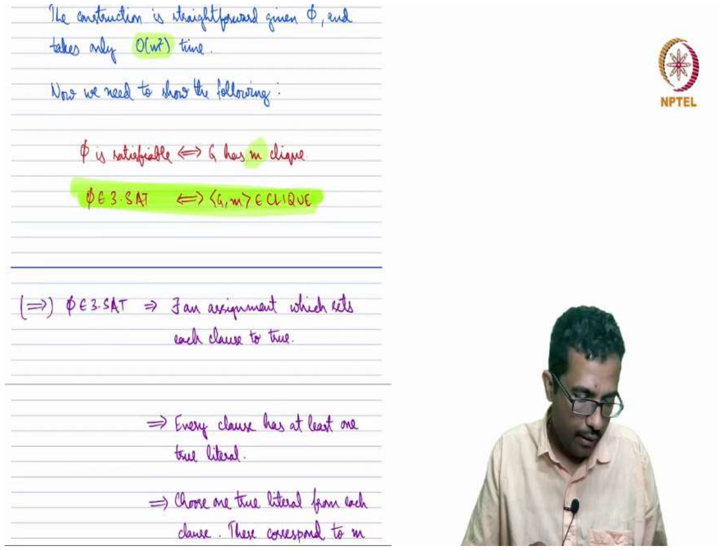


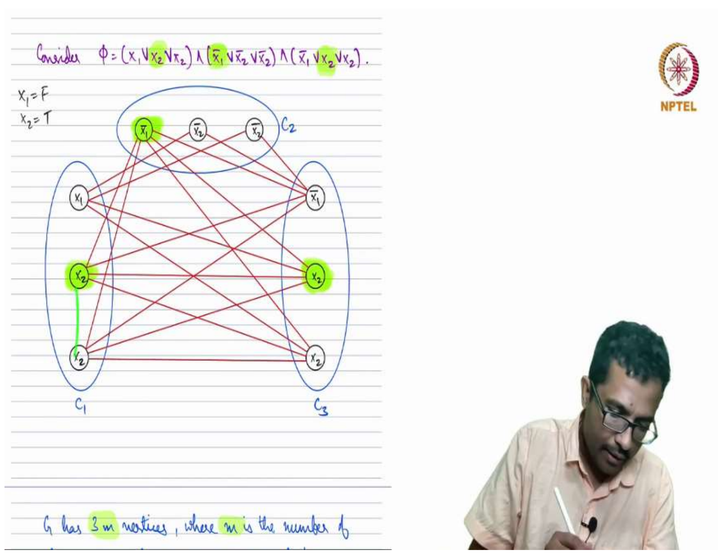


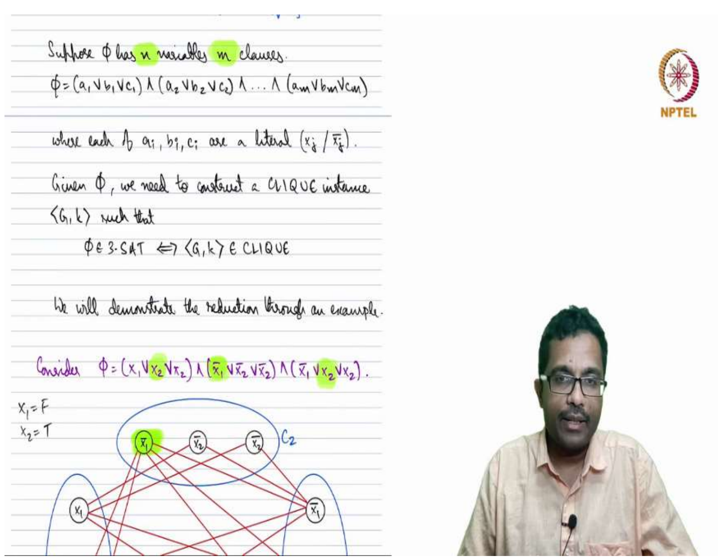
And since each clause is satisfied, that means that the formula is in 3 SAT. There is a small point. So, it is possible that the since we assigned some xi to be true, some xi to be false. It is possible that some other some xj's, or some xk's may not have been assigned anything, maybe the same, let us say x1 may satisfy 2 clauses, x2 may satisfy 3 clauses. So maybe it so happens that x4 is never assigned by this process.

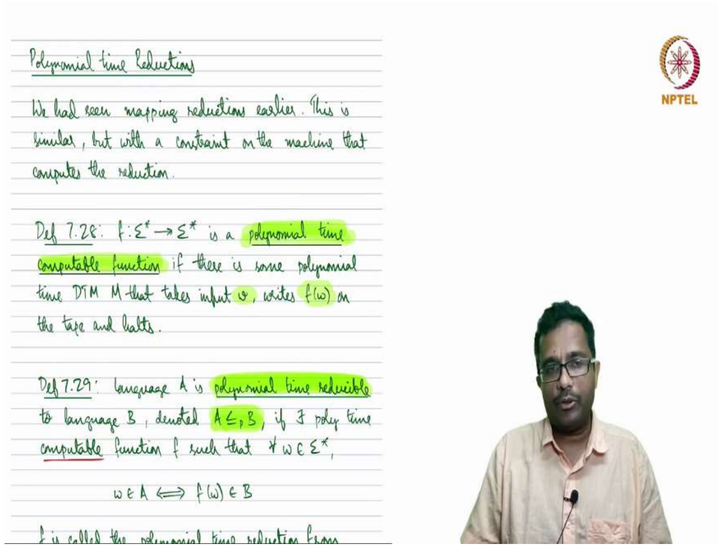
But what it means that we can assign this x4 to true or false, it does not matter. Whatever we assign, we have already satisfied all the clauses. So in any clause that contains x4, there is some other variable that has satisfied that clause. So, whatever we assign x4 does not matter, so the unassigned variable, we assign anything, it does not really matter, but the point is that by the virtue of assigning these literals that were chosen from the CLIQUE, we have satisfied all the clauses, and hence, the formula is satisfiable.

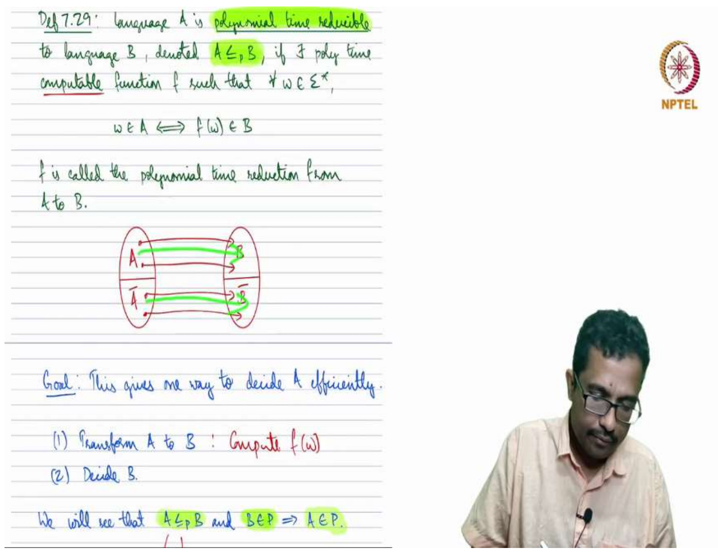
(Refer Slide Time: 22:55)











So, should we just recap we have to show that formula is satisfiable if and only if the constructed graph has an m CLIQUE, if the formula is satisfiable, we consider a satisfying assignment. And we pick 1 variable or 1 literal from each clause that is satisfied. And the claim is that, this forms an m CLIQUE.

Because they are from m clauses. And because it is a satisfying assignment, it will not have both xi and $\overline{xi}$ for any i. So this forms m CLIQUE. So, if φ satisfiable G has an m CLIQUE for other direction, if G has an m CLIQUE, we know that each group or each clause has only 1 vertex cannot have more than 1 vertex, because there are no internal edges.

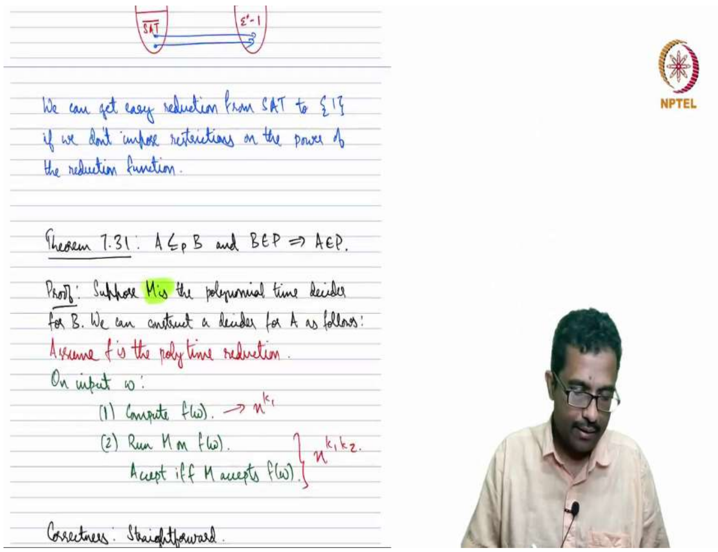
So, which means each group has exactly 1 vertex. And now, we just select those groups. And because we do not put edges between xi and $\overline{xi}$, we know that all these selected literals or all the selected vertices or the literals that correspond to these vertices, they are not conflicting, and we can set all of them to be true.

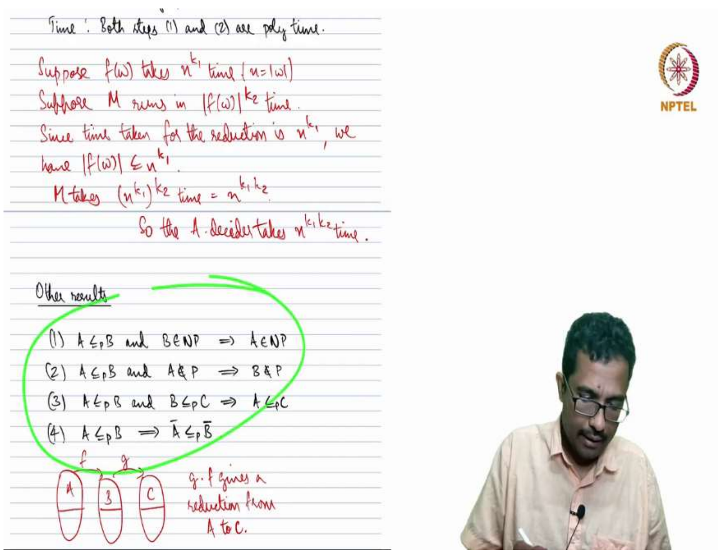
So, suppose, x1, x2, $\overline{x3}$, $\overline{x4}$ complement are picked, then we set x1 and x2 to be true and x3 and x4 to be false. So now that translate to at least 1 satisfying literal in each clause, so each clause is satisfied, hence, the formula is satisfied. So there may be variables that we do not set either way, but we can set in any way it does not matter.

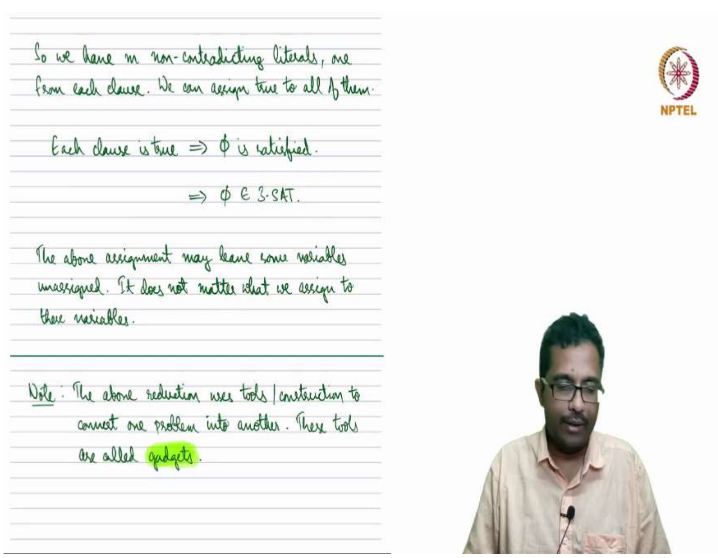
So, that completes the proof of correctness of this reduction. So, the reduction is straightforward, the proof of correctness is a bit more involved. So we saw that, given a 3 SAT instance φ, we can construct a graph and m says that if φ is in 3 SAT if and only if G has an m CLIQUE.

And prior to that we defined polynomial time reductions. So, it is a transformation from an A instance to B instance such that every yes instance is mapped to a yes instance and every no instance is mapped to a no instance. And further the computation of this reduction function should be in polynomial time. So, we had all these results.

(Refer Slide Time: 25:33)







A reduces to B and B is in P implies A is in P, A reduces to B in A not in P implies B not in P and all of this and finally, that kind of concludes the lecture 48. And I just want to make 1 small remark before concluding. So, the construction that we had for showing that satisfiability reduces to CLIQUE. So, this was there is a certain pattern or structure to this.

So, many times when we are reducing 1 problem to another, sometimes we need to make use of some of these types of constructions. And these constructions are like commonly called as gadgets. So, these tools for making the reduction, so sometimes they are called gadgets, just to familiarise with the term. We will not be seeing that many reductions in this course, maybe like 4 or 5 reductions, maybe in the next week.

But if you look at a book on reductions, and if it uses the word gadgets, I am just saying this so that you can be familiar with this usage. So that is it. So, we saw reductions, we saw the definition. We saw 1 example that 3 SAT reduces to CLIQUE. And that is all for lecture number 48. And next we will see NP completeness in the coming lectures. So, see you in lecture 49. Thank you.