Theory of Computation

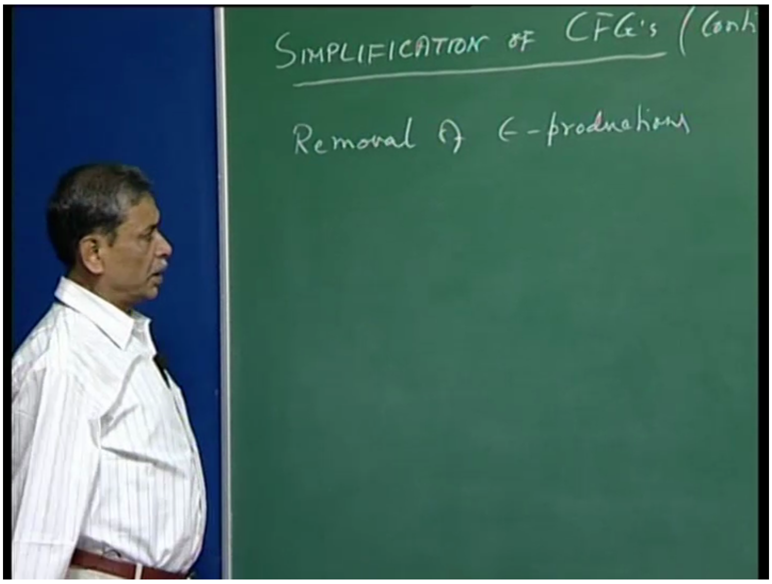
Professor Somenath Biswas
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Lecture 24

Simplification of CFGS Continued, Removal of Epsilon Productions-Algorithm and its Correctness

We will continue our discussion on simplification of context free grammars. Last time we saw how useless symbols can be eliminated from a given CFG to produce a new CFG which would be equivalent in the sense the new CFG also will generate the same language as the old CFG. Today first let us discuss how to remove so called epsilon productions?

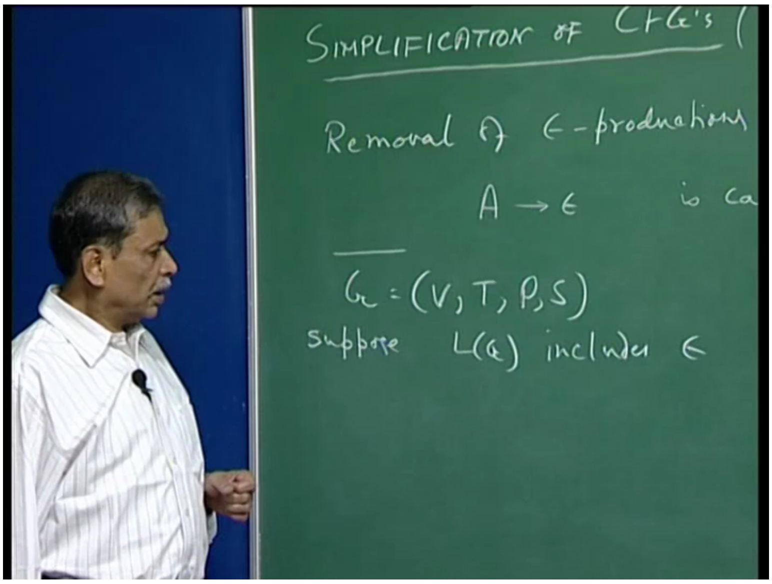
(Refer Slide Time: 01:06)



First of all the definition of epsilon production. Any production of this form A goes to epsilon is called an epsilon production. Such a production the left hand side as usual in case of CFGS is a nonterminal. The right hand side consists only of the empty string. And in general there can be many epsilon productions in a grammar and we would like to eliminate all such productions from the grammar to form a new grammar G.

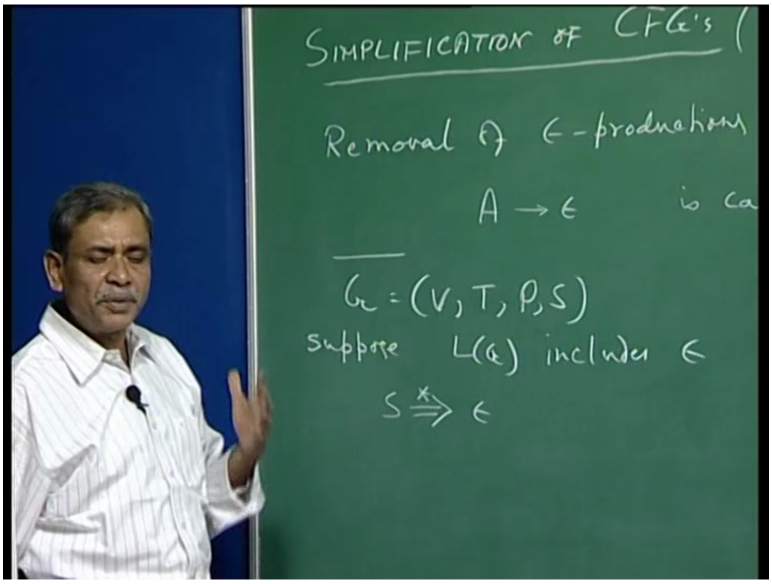
Now suppose the old grammar G produced or generated the string epsilon itself. So let ussay what we want to say. So suppose I had a grammar G, V, T, P and S and suppose L G includes the string epsilon. That means the string epsilon the empty string is in the language generated by the grammar G.

(Refer Slide Time: 02:48)



Now that really means that from S we will derive the string which is the empty string epsilon. Now this derivation is not possible unless you have epsilon productions clearly because somewhere down the line you must be able to remove all nonterminals that you have generated from S andsubstitute them by epsilon to ultimately obtain this empty string epsilon.

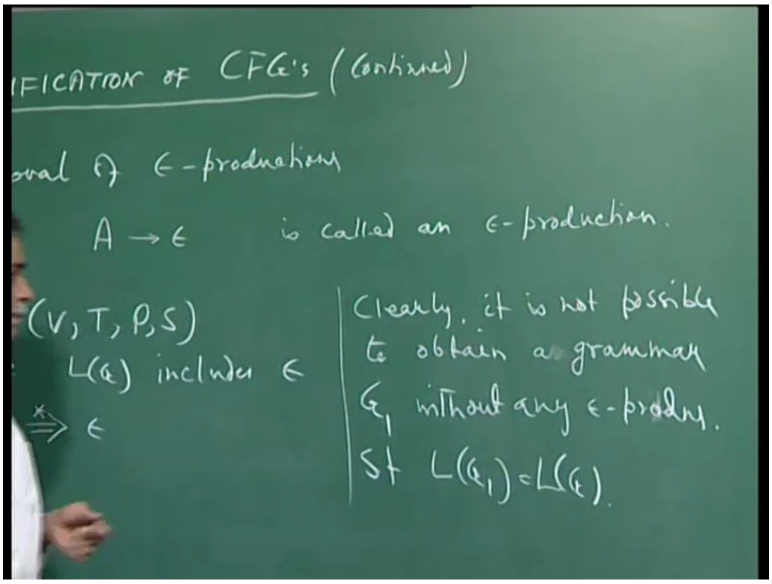
(Refer Slide Time: 03:25)



So it is not possible very clearly to eliminate all epsilon productions from a grammar G and get an equivalent grammarG 1 because G 1 if it does not have any epsilon production then it will not be able to generate the string epsilon itself and in that sense the two grammars will not be equivalent. So let me say it more clearly what I am trying to say.

That suppose L G includes epsilon, right? Then clearly it is not possible to obtaina grammar G 1 without any epsilon productions such that L G 1 is same as L G, right? Because if G 1 does not have any epsilon production it can never generate this string epsilon itself, however much its size.

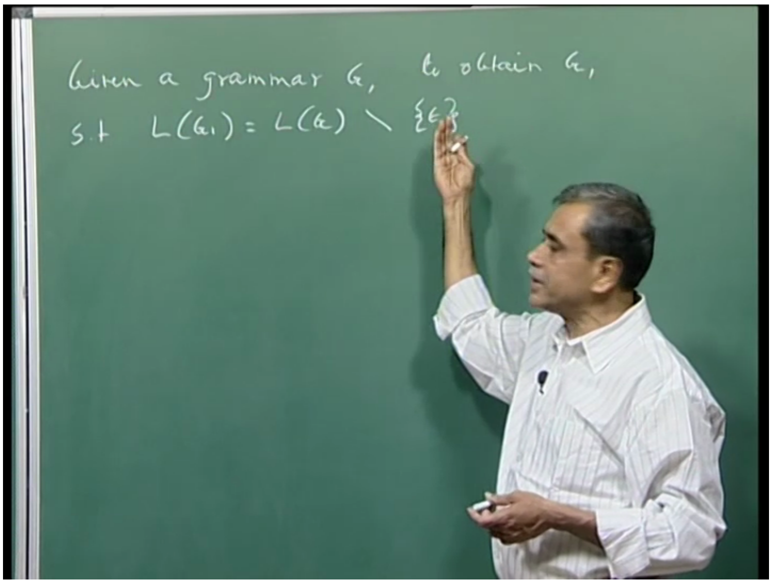
(Refer Slide Time: 05:13)



Because initially any derivation in G 1 starts with a nonterminal and if you say that it finally derives epsilon, somewhere all the nonterminals must be erased and there will be no terminals left because once terminal is written it cannot be erased in the production. So therefore only way you could generate this string epsilon is by having epsilon productions. So clearly we cannot remove epsilon productions from all grammars and obtain an equivalent grammar.

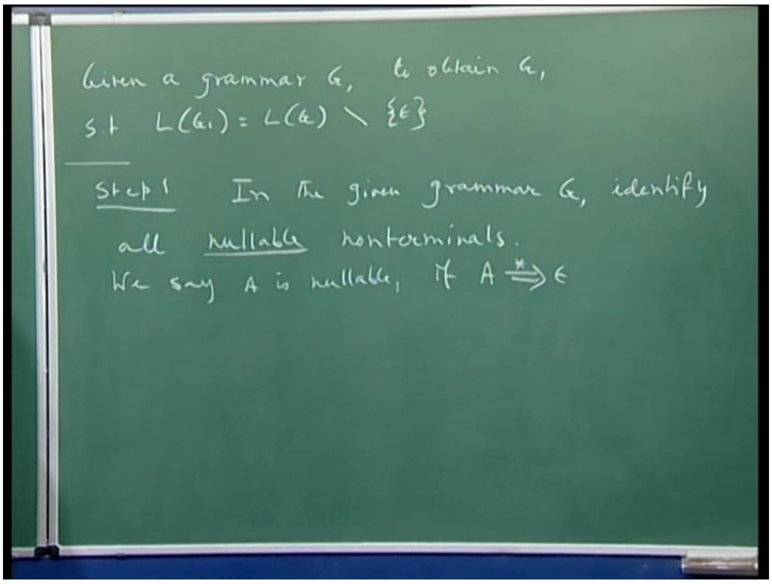
Our goal should be the following that given a grammar G to obtain G 1 such that L G 1 is equal to L G without the string epsilon. In other words if generated epsilon then G 1 should not generate epsilon but G 1 should generate all other strings which this grammar does. On the other hand if Gwould not generate epsilon that means L G did not contain epsilon then this would be equal to L G itself.

(Refer Slide Time: 06:56)



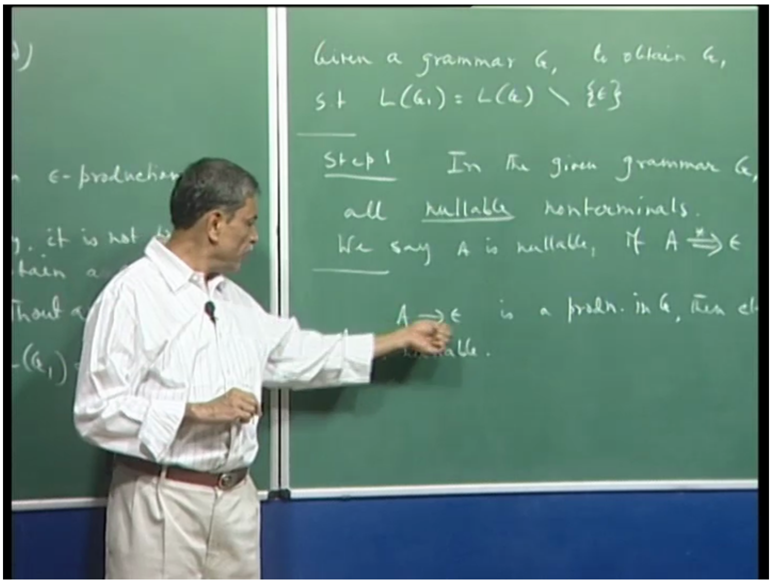
In such a case of course we should get a grammar that is our aim which would be equivalent to the grammar G. So this is our goal. And the way we achieve this is first identify from the given grammar G or step 1 is, in the given grammar G identifyall so called nullable nonterminals. What is a nullable nonterminal? We say A is nullable if A can derive the string which is epsilon that is empty string.

(Refer Slide Time: 08:22)



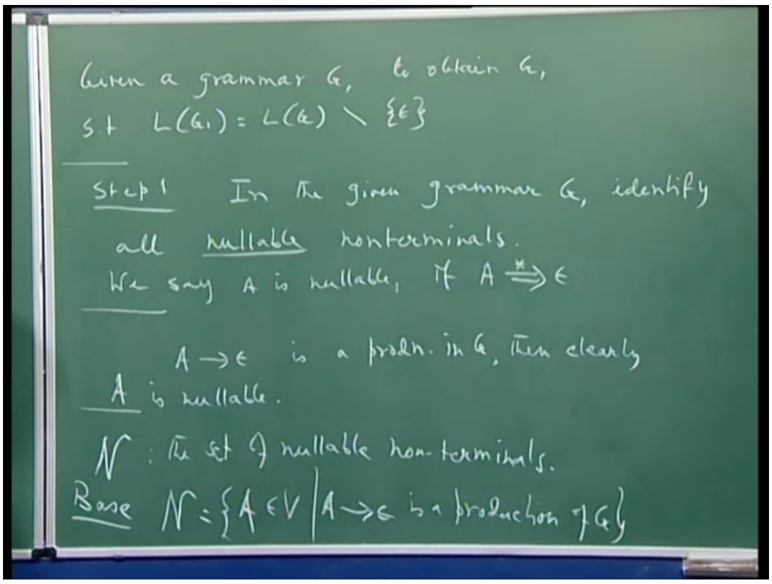
That means in other words A can finally become the string epsilon after some steps. Now clearly if a nonterminal is there in the grammar G such that A goes to epsilon is a production in G then clearly this nonterminal is nullable. Now the way we define the set of or identify the set of all nullable nonterminals of the grammar G is by an inductive process and in the base case of the inductive process, the base of that induction we will start with identifying or finding out all nonterminals which have such a production, okay.

(Refer Slide Time: 09:43)



So let us say the set of nullable terminals, let me call it script N. Let it denotes the set of nullable nonterminals. As we have done in some other cases previously we will define this set script N inductively. And the base is that this N consist of initially the set of all nonterminals A in V such that A goes to epsilon is a production of G.

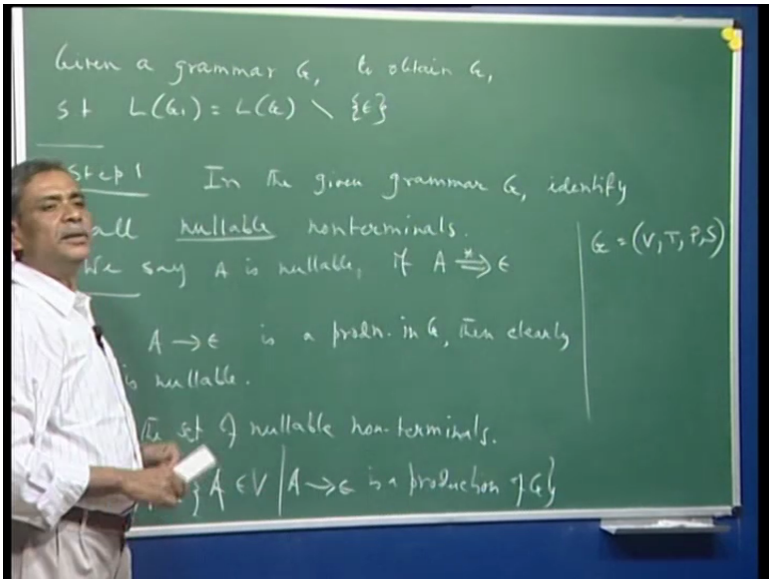
(Refer Slide Time: 10:58)



Remember any inductive definition of a set starts with a base definition of that set and that base definition for this set ofnullable nonterminals is all those nonterminals of the grammar such that each of which contains a production of the form A goes to epsilon. So you know clearly that such nonterminals are visiblyright from the inspection of the grammar G without doing anything I know that these nonterminals are nullable.

Here it is obvious but I should mention once more that my grammar G from which we are trying to remove epsilon productions that was of the form V, T, P, S and this Vtherefore is the set of nonterminals and that is the one that I am using here.

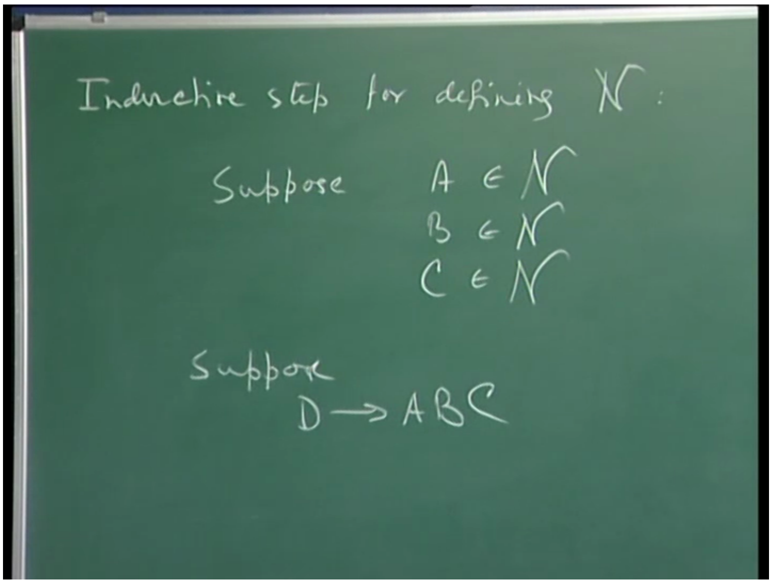
(Refer Slide Time: 12:01)



And now I should say what the inductive step is? Inductive step for defining this script N is quite simple. So let us say at some given time I have defined a set which consists of nullable symbols that I have identified so far. And now at this pointyou know at every stage of this construction we look at all productions, right? Suppose particular non terminal is in mynullable sets that I have defined already. This non terminal is there so I know it is nullable.

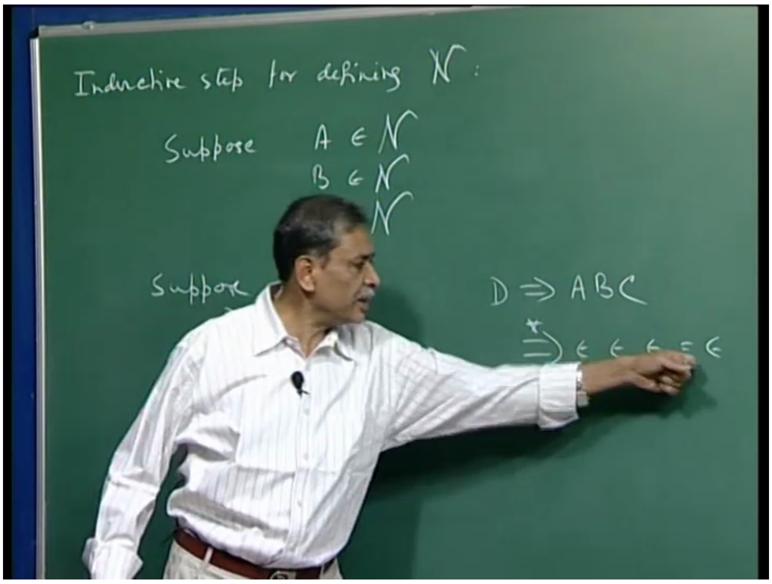
As well as some other ones. So let us say A, B, etc. Sofor the sake of example so let us say C is also there in N and suppose D is a production of this kind A B C.

(Refer Slide Time: 14:07)



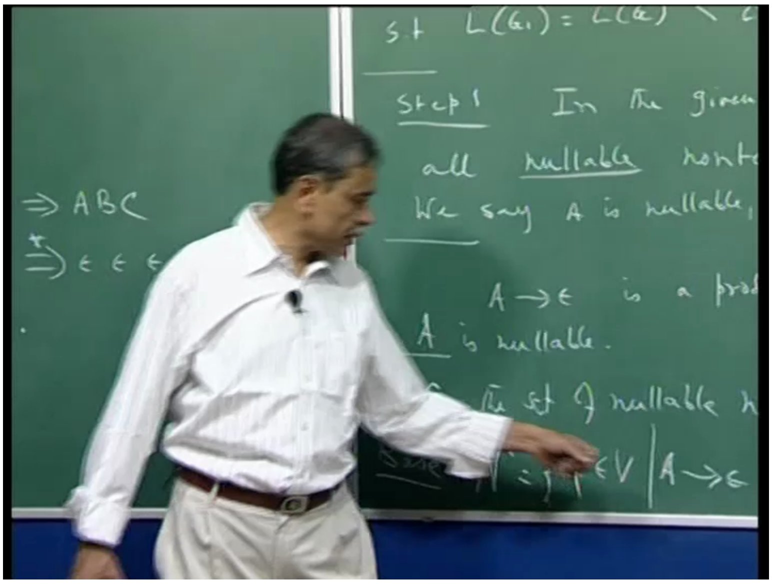
Then you can see clearly that because A is in the set of nullable symbols that you have already defined, I can start a derivation like this D that goes to A B C. I used this production to get this and now afteruse of some productions A will go to epsilon and then B also will go to epsilon, C also will go to epsilon. Therefore I get that D will eventually can generate this string epsilon.

(Refer Slide Time: 14:44)



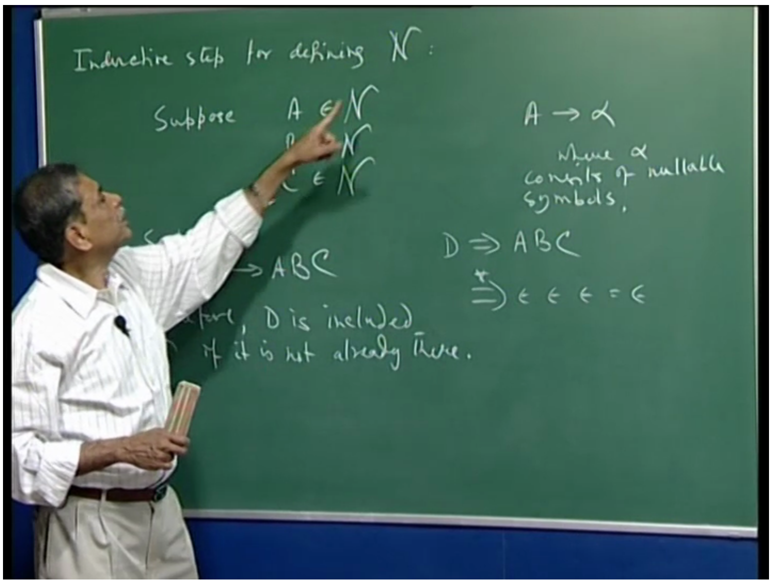
So therefore in such a case D also have discovered to be a nullable symbol and therefore D is included in this set ofnullable nonterminals if it is not already there. Now what is the algorithm corresponding to this? The algorithm is this step is of course clear that is the base step that is easy to see. First of all I defined my initial version of N to be all those nonterminals which have a production of this kind.

(Refer Slide Time: 15:46)



And then whatthe algorithm will do this that it will examine every production andidentify a production. If it comes across a production of the kind that A goes to alpha where alpha consist of nullable symbols then we check if A is there already in the set N.If it is not there then we include this new nonterminal into N. And now I have changed the N from the old one. So that anytime we manage to update non-trivially this set that is the endof one iteration.

(Refer Slide Time: 16:57)

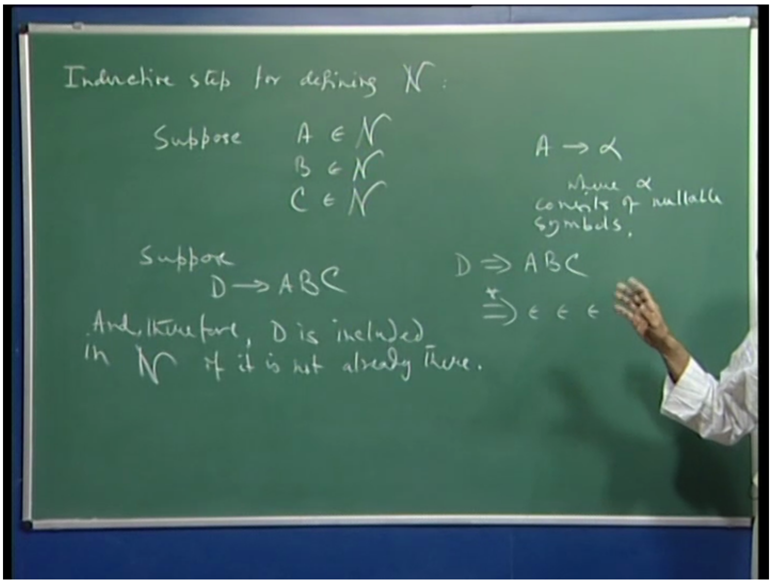


The every iteration we start with some N and then we will look through all the productions, identifyusing basically this rule that if anew member can come into N, anytime I find so that is the end of that iteration. I start the next iteration. How many times this iteration can go on? Clearly atmost thenumber of nonterminal symbols which are there in theset of nonterminals, right? It cannot exceed that.

It will be less than that because you already have some members to begin with. And onceall the iteration stops then I claim that we have identified the set of all nullable symbols, right? So thealgorithm is very simple. We start with a base case and then myiterative process starts. Every iteration examines the set of all productions and if it finds a production such that the right hand side of the production consists only of nullable symbols that you have already identified and the left hand side nonterminal is not there inthe current N.

Then we stop that iteration because we have already discovered an update for N. We include that symbol that we have just found to be nullable into N and start the next iteration and we go on like this. And it is not difficult to prove if you wish that this algorithm will correctly identify the set of all nullable nonterminal.

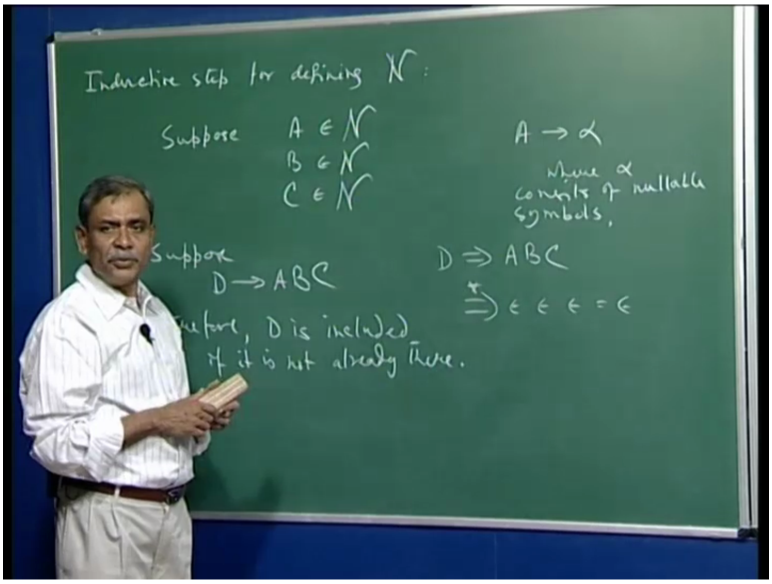
(Refer Slide Time: 19:14)



Firstly it is clear that the way we have described that no nonterminal which is not nullable can get into N, right? Becauseany symbol which came into the seteither because it was there in the (ba) base case.Because then clearly in such a case of course that symbol is nullable or it came in the set Nin one of those iterations. And because of application of such a case and then we found in factthat is the witness that A is nullable.

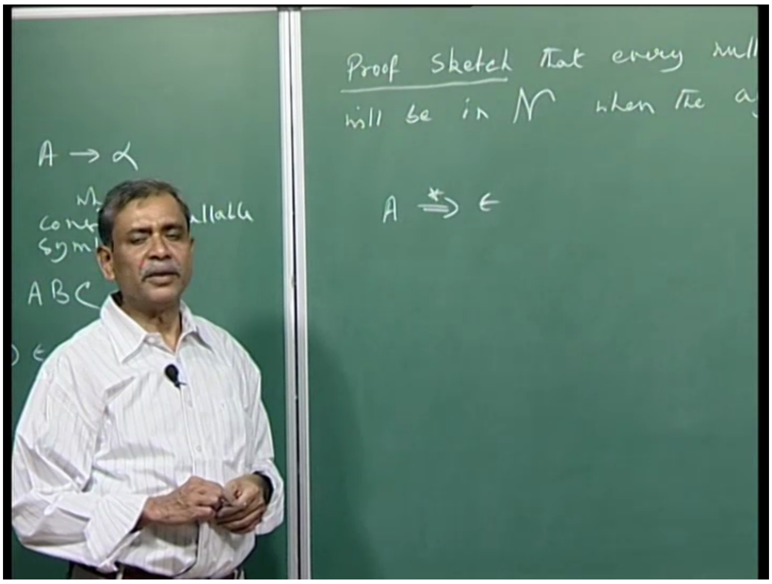
So every non terminal which we put in the set N is clearly nullable. Only question that you may ask is have we identified or is there any nonterminal is it possible that which is nullable but which did not or would not get into the set Nbecause of the algorithm that we have used.

(Refer Slide Time: 20:30)



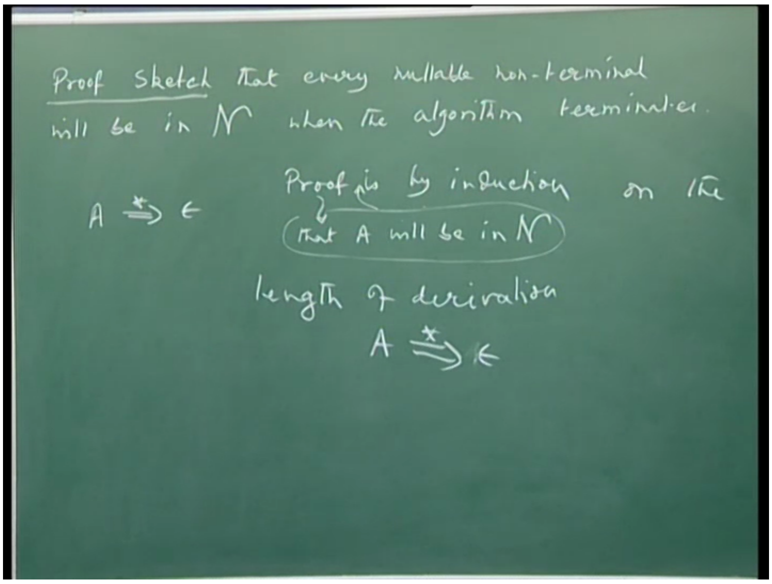
Let us provide a proof sketch that every nullable nonterminal will be in N when the algorithm terminates. The algorithm which we have used to define script N. Now by definition a nullable symbol is something which can generate a string epsilon. The way we prove this assertion that N contains all nullable nonterminals is by showing such an A will eventually get into N. A is nullable by definition therefore there is such a derivation that will take A and lead to an empty string. And our proof is on the basis of the number of steps required in this derivation.

(Refer Slide Time: 21:44)



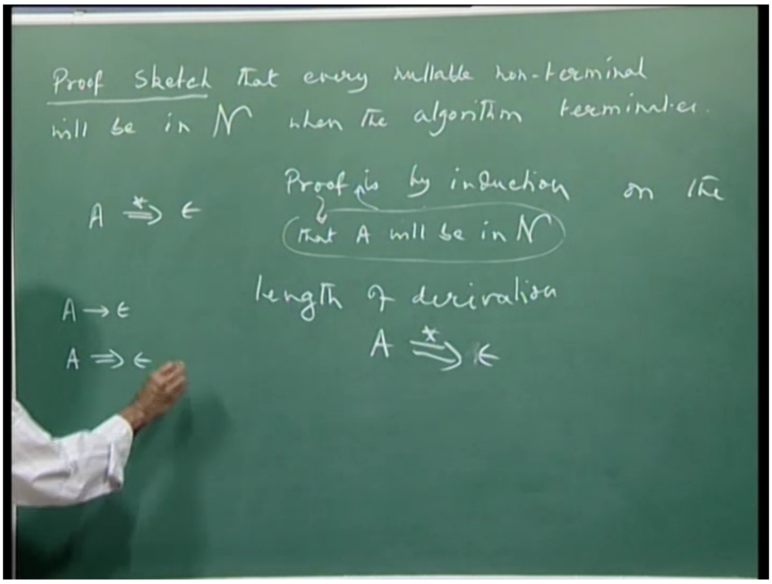
You know it is not surprising that we are using induction to prove this assertion because N was defined inductively so therefore it is not unnatural that we will use induction to justify the main claim about the algorithm. So whatis it? The proof is byinduction, proof of what? Proof, so let me write this, proof that A will be in N. This proof is by induction on the length of derivation which gives the empty string from A.

(Refer Slide Time: 23:01)



The base case is thatthis length is 1. How is that possible? That is possible if A goes to epsilon is a production of the grammar. So in that case we will have a derivation of length 1 because given such a production there we will start with A, use that production to simply generate epsilon, right?

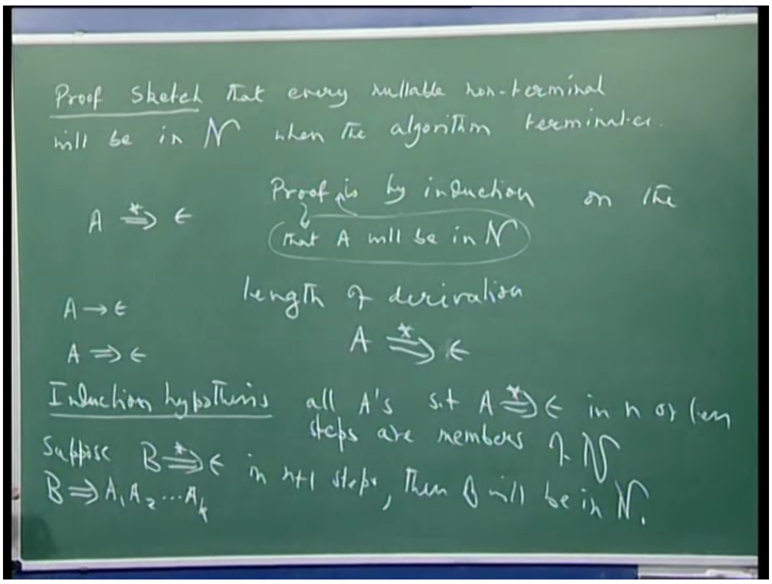
(Refer Slide Time: 23:30)



Now therefore the base case is clear because you see that in the beginning the (ba) inductive definition of N, the base case of the inductive definition then would contain all such A's which had a one-step derivation to generate epsilon. Now induction hypothesis is all A's such that A derivesepsilon in n or less steps,all such A's are members of script N. And using this induction hypothesis I would like to show that suppose B generates epsilon in n plus 1 steps then B will bein N.

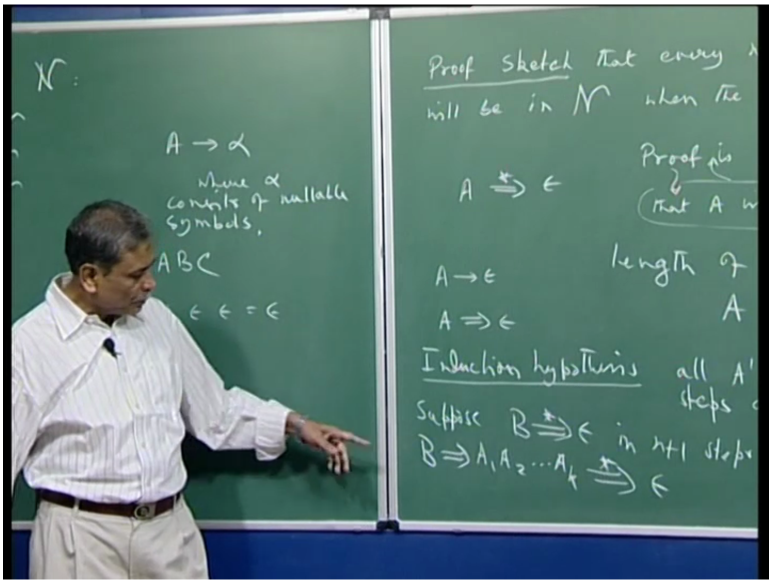
That proof is simple you see because consider such a derivation. Consider the very first step of that derivation. That will be thatyou will replace B by the right hand side of production whose left hand side is B and that left hand side may have a number of nonterminals.

(Refer Slide Time: 25:51)



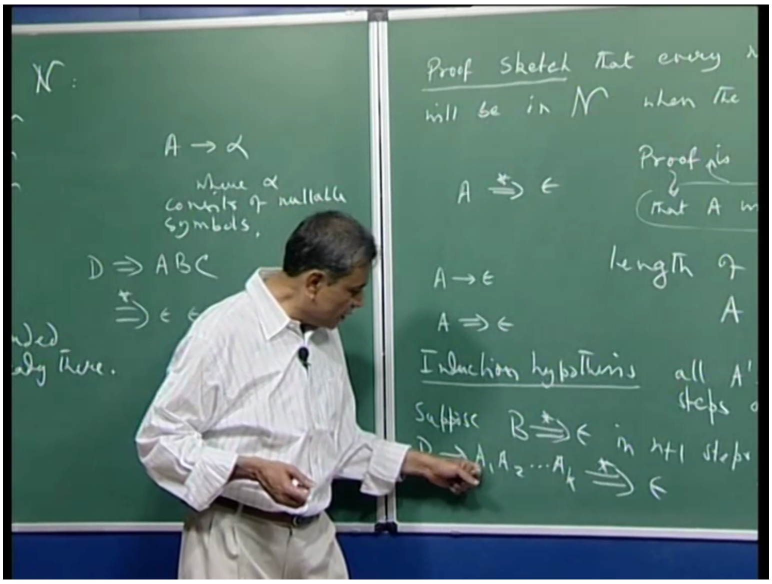
It is clear that the right hand side of a production which contains a terminal symbol could not be used in the derivation because the terminal symbol cannever become part of an empty string there. So let us say and the very first step was using a production of the kind B goes to A 1, A 2, A k, right? And then eventuallyall these A i's must be nullable themselves. So they are finally written off as epsilon and therefore B derivesepsilon.

(Refer Slide Time: 26:37)



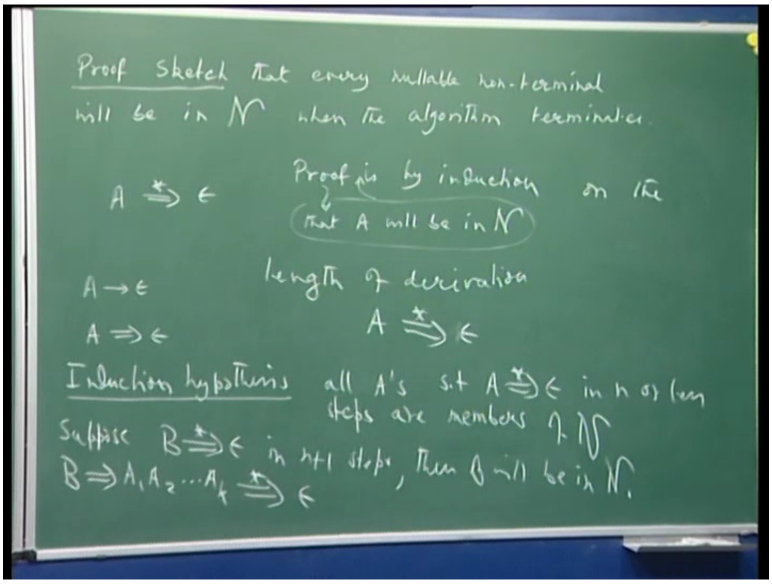
The point is since this derivation is of length n plus 1, every derivation of the kind that A 1 goes to epsilonsoin that I must be using the fact somewhere that A 1 you know ultimately derives epsilon and that derivation will have a number of steps which is n or less. And therefore by induction hypothesis each of these A i's will be already there in N, right? So and that time you know when we are in our algorithm there will be a time when I would find all these symbols are nullable.

(Refer Slide Time: 27:22)



Algorithm toidentify all nonterminals which are nullable and in that algorithm when I examine A 1 through A k, this particular production. After we have identified A 1 through A k are inthe set script N then clearly we will add B also. So therefore B also will get into the set of nullable nonterminals which is what we wanted to prove.

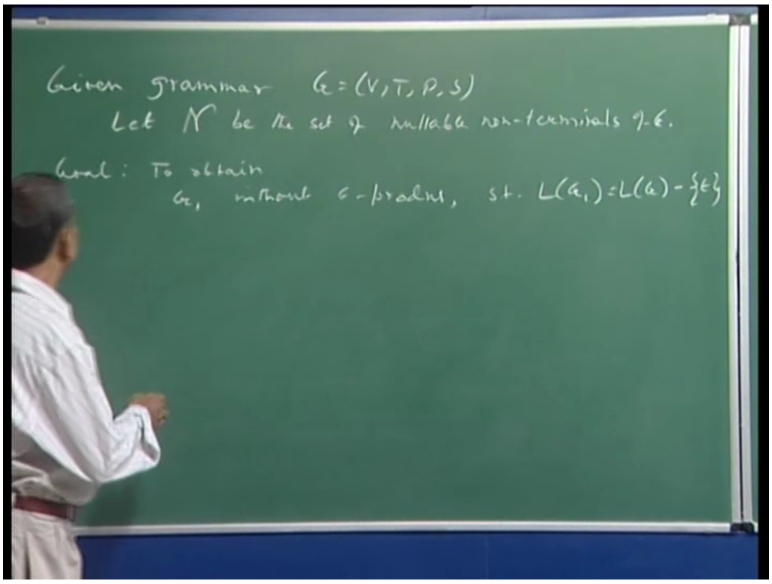
(Refer Slide Time: 28:06)



What we have achieved so far is the identification ofall nullable nonterminals of a given grammar G. So now we should proceed to obtain what we wanted to, essentially a grammar without any epsilon productions and which will generate the language which is same as the old grammar language except possibly the string epsilon. So let us write this down. Given grammar is G V, T, P, S and let N be the set of nullable nonterminals of G.

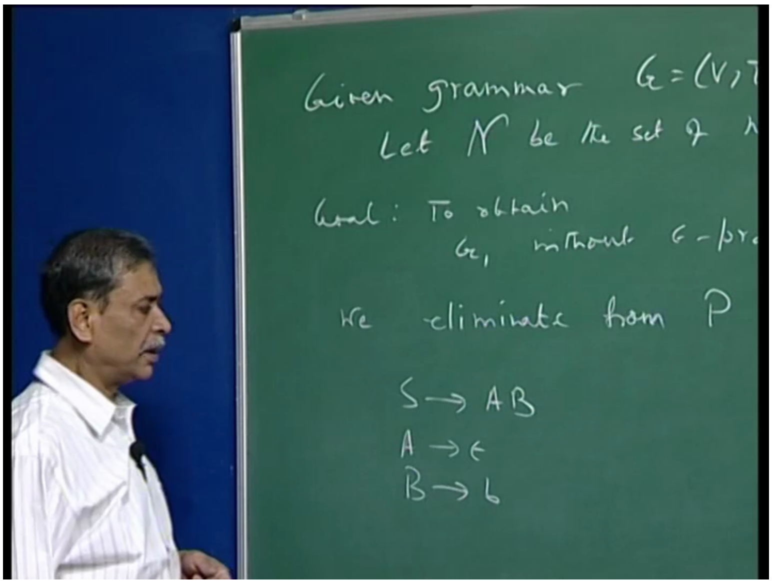
And our goal is to obtainG 1 without epsilon productions such that L G 1 is L G without the string epsilon. If it is as we said already that if L G had epsilon, L G 1 should contain everything other than that stringepsilon. If L G did not have epsilon then the new grammar and old grammar they generate identical languages, alright.

(Refer Slide Time: 30:17)



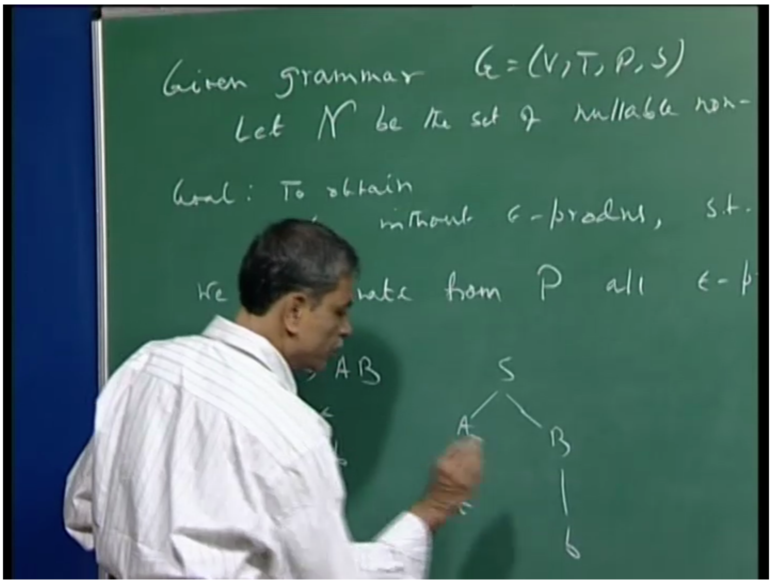
So how we do this?first of all what we do is we eliminate from P, which is the set of productions of G, all epsilon productions. Now clearly at this time the grammar does not have any epsilon productions but the grammar is not the grammar that I want. The reason is if you remove all epsilon productions it may be that you are blocking some non-epsilon strings which are in the language from being generated. A very simple example suppose S goes to A B and A goes to epsilon, right? And B goes to b.

(Refer Slide Time: 31:36)



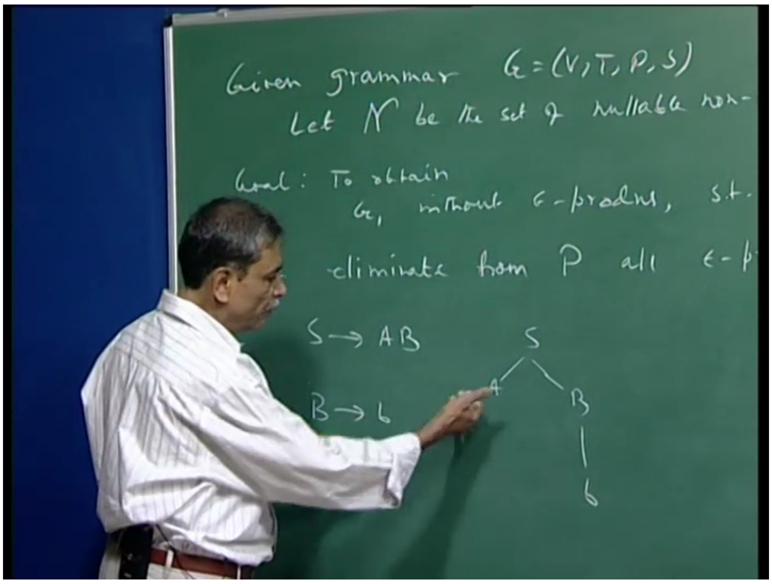
Supposingthis is the grammar that you have. These are the set of productions in the grammar then what this grammarcan generate? You can see this A B. A goes to epsilon and then this B goes to b.

(Refer Slide Time: 31:58)



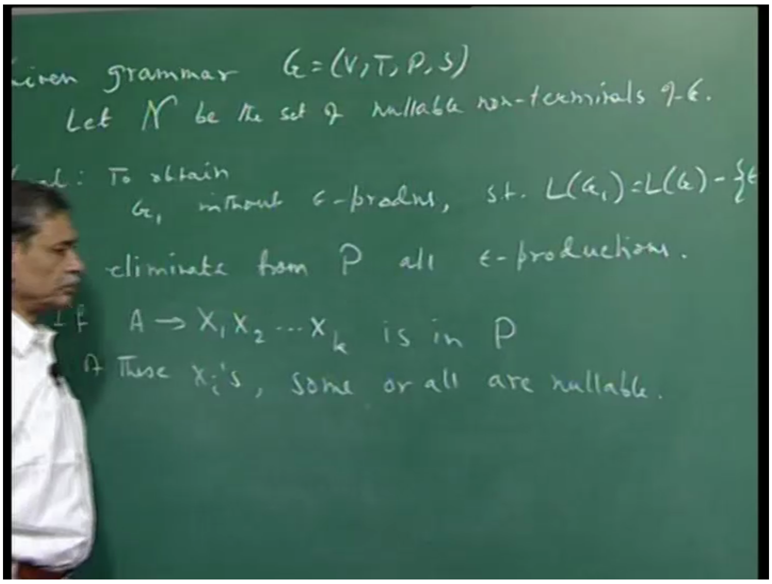
Now if I remove from this the set ofthe epsilon productions then this part of the derivation tree cannot be there. So therefore I will not be able to generatethe string b whichoriginally I could generate. Because you see that S goes to A B and then there is no way of getting rid of this A.

(Refer Slide Time: 32:33)



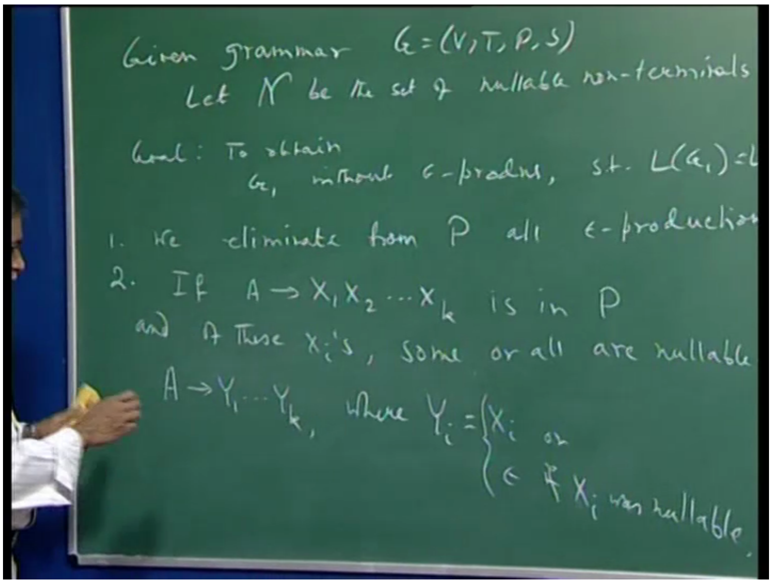
So clearly we need to do something more and that is done by adding some new productions and that rule is fairly simple. So this is first thing we do. Second thing that we do is if A goes to let us say X 1, X 2, X k is in P and let us say that of theseX i's some of these X i's are nullable. Some or all are nullable. It can happen.

(Refer Slide Time: 33:46)



Then what we will do is the following. We will addanother production of this kind that A goes to Y 1 to Y k, right, where Y ican be X i or epsilon if Y i or let me say if the symbol X i was nullable. So what we are saying is this looks a little clumsy the way we are writing it but the idea is very simple.

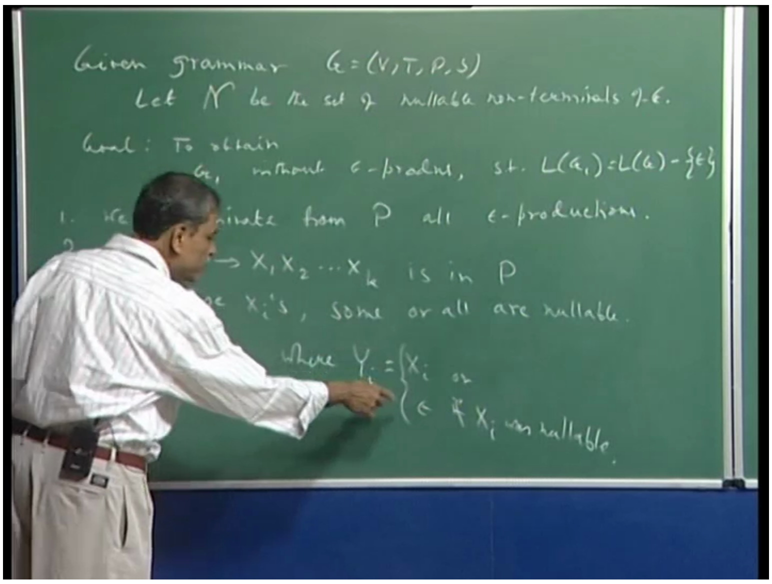
(Refer Slide Time: 34:50)



You look at a production and then what you are doing is these kinds of productions you will add. So letus take a simple example that suppose I have A goes to B C D and then out of this let ussayB and D are nullable. What we are sayingis that this is a production that we will add.

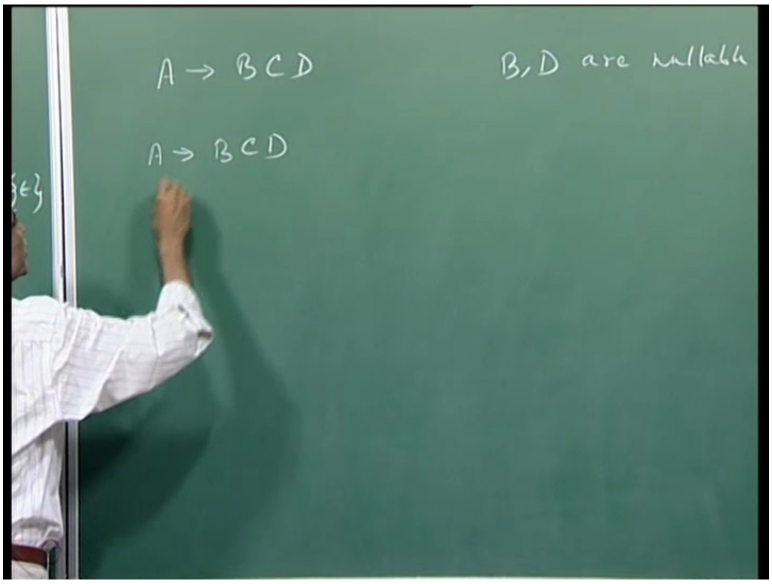
Now the way we have written it you should realize that it is not really one production that we are adding. In general we are adding many because this choice is there, okay. If X i is nullable thenthe right hand side of the production can have either epsilon or X i itself in its place.

(Refer Slide Time: 35:59)



So in this simple example what can happen is, so this is my X 1, this is my X 2, this is my X 3. So how many new things that I can get out of this? I canofcourse keep everything. So this is the production that we will keep because that comes by never using this choice epsilon for any nullable symbol. Or like let us take this first one.

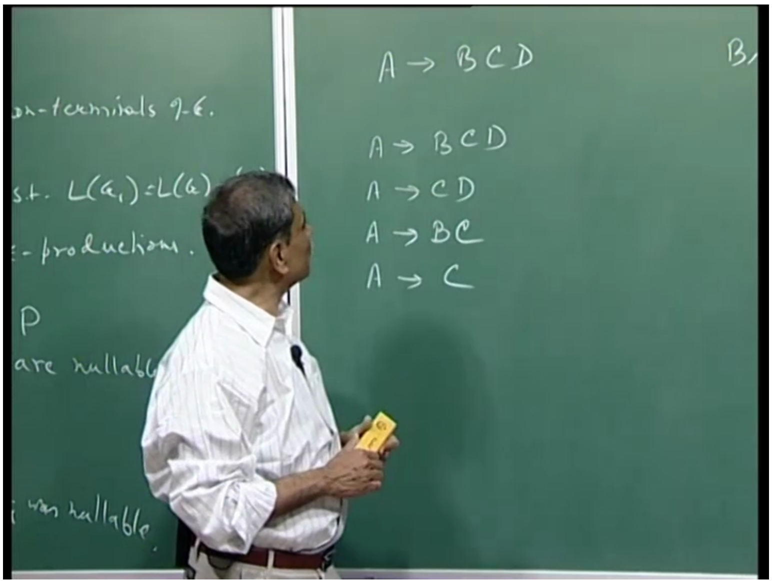
(Refer Slide Time: 36:37)



So I can say that A goes to C D also be a production because according to this rule thatBcould beeither epsilonor itself. So here it is itself and here we are choosingto make it to be epsilon. Long and short ofthis thing is that all those strings which can be obtained from this by substituting one or morenullable nonterminals by epsilons such right hand sides will also be a production of right hand side of A.

So this is a new production. So the kinds of new things that I am getting from here you can see I can get C D. I can get of course so B was replaced by epsilon. So I can also get B C, right? And also I can get A goes to C because at that time I replaced both B and D together by epsilon.

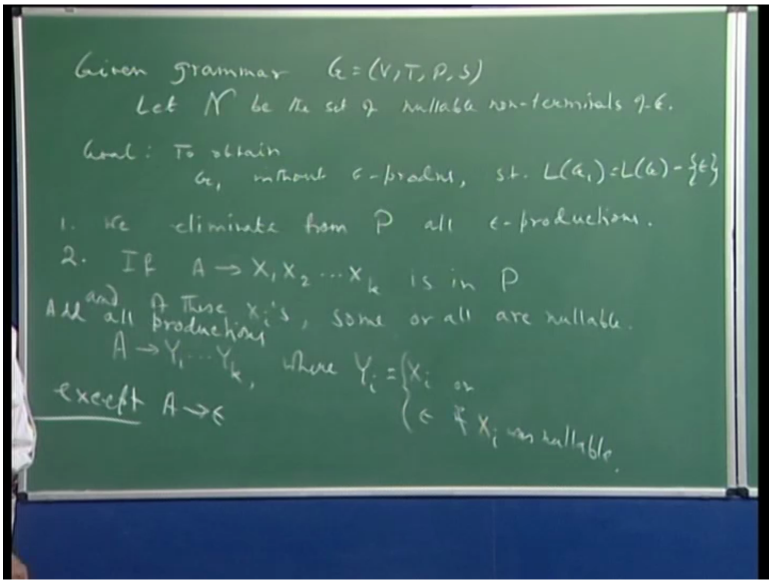
(Refer Slide Time: 37:58)



Now these are the new productions that we are adding in this case. Now it is possible that all of them are nullable, right? In the way we did that possibility is also there. Now if we do this what I saidI will be allowed to replace each one of them by epsilon and then I will get an epsilon itself on the right hand side. Now that is not allowed. So in other words my rule is obtain all productions of the form orlet me say it this way, add all productions of the form A goes to Y 1 up to Y k where each Y i is either X i or epsilon if X i is nullable.

The ith place could contain epsilon which is fine except A goes to epsilon, right? Except this production andthat possibility is there when all of them are nullable. So if I had just said this much then you could have replaced each X i with epsilon and then the right hand side would have been epsilon itself. So that possibility we are removing.

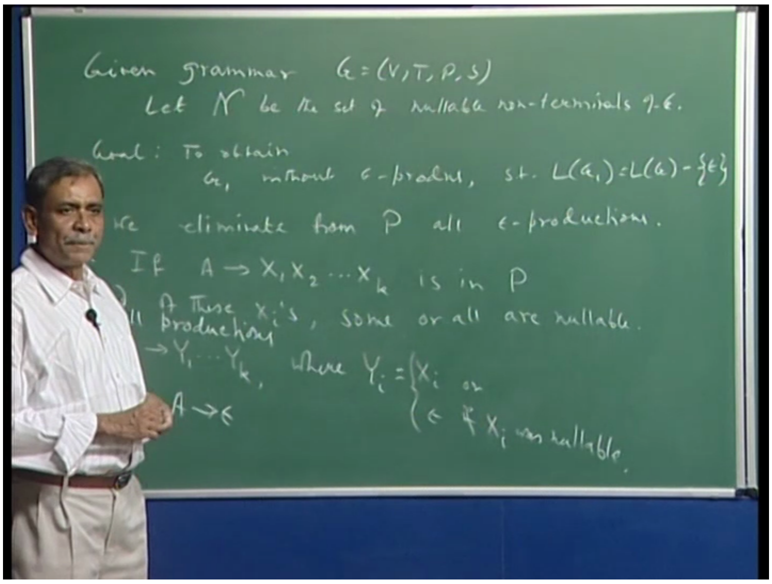
(Refer Slide Time: 39:39)



So what is our G 1 which has the property that it generates every string of the grammar G except epsilon string, right? The string epsilon. So as we said that first we identified all the nullable symbols of G and then we eliminate from P all the epsilon productions. If there are nullable symbols thenthere will be some epsilon production. Sothis new G 1 we are creating by first removing all epsilon productions and then we are adding some new productions and these are the new productions that we add.

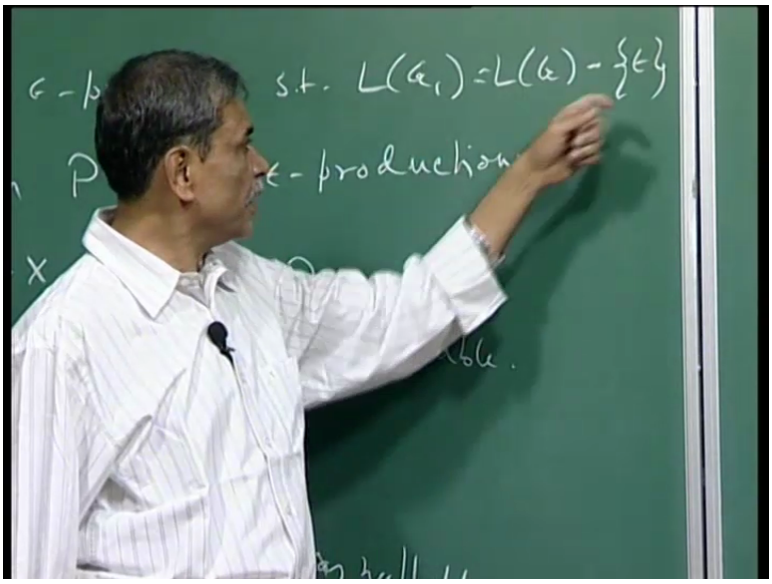
After the identification of all nullable symbols we said that suppose this is a production in P then we add some new productions by removing one or more of these symbols which are nullable and obtain a newproduction A goes to something except you knowwe will not add any production of the form A goes to epsilon, right?

(Refer Slide Time: 41:04)



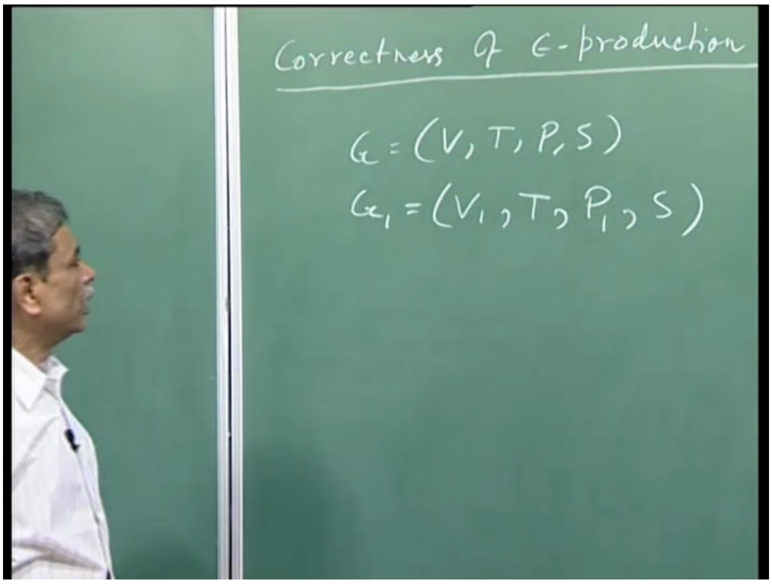
So this ishow we define our new grammar G 1. Very briefly again that we identify all the nullable symbols, then we remove all epsilon productions from G and then we add some new productionsto the set of productions and then the final form of the grammar that we have now can be shown to generate all strings of G except the empty strings.

(Refer Slide Time: 41:38)



Let us try to prove this. To establish the correctness of what we are doing let us consider the original grammar to be as we said V, T, P, S and the new grammar that we got after removal of all epsilon productions and in the process adding a few more productions,that grammar let us call G 1. Possibly wemight have removed somenonterminals. Terminalswill not have removed and let us call the new set of productions for the grammarthat we have is P 1 and of course S.

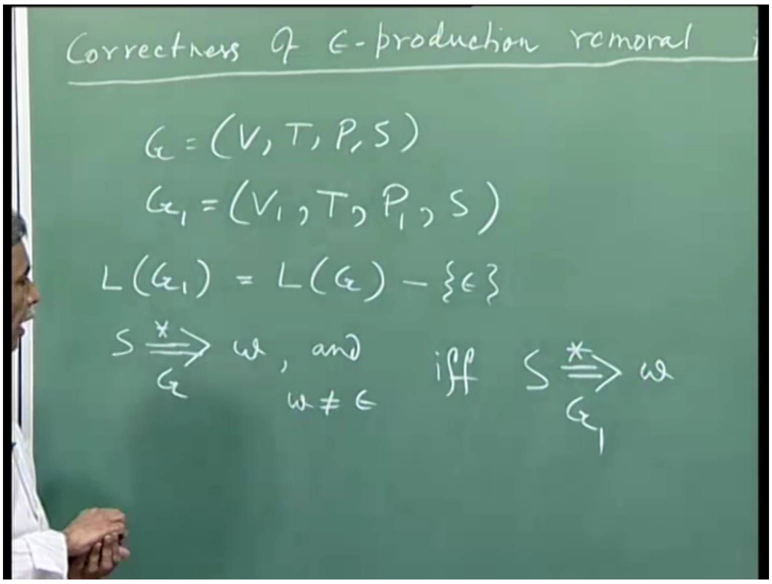
(Refer Slide Time: 42:34)



The way what we want to show is L G 1 is same as language generated by the original grammar minus possibly the string epsilon, right? So what we need to show so we can see that suppose S derives in G w which is a string of terminals and w is not epsilon. This is the case if and only iffrom S you can derive the same string w in the grammar G 1. Recallsince we are talking about two grammars and this symbol that we have been using before needs to be now qualified to indicate derivations in which grammar we are talking of.

So this is easy to say that what we are trying to say that suppose in the original grammar we derived some string w and w is not epsilon,that string will generate in the new grammar also as well as if in the new grammar we generate any string w then clearly we want that stringbecause there is no way we can generate the epsilon string because there are no epsilon productions in G 1. So w is not epsilon and that wwe should be able to show that it can be generated in G as well.

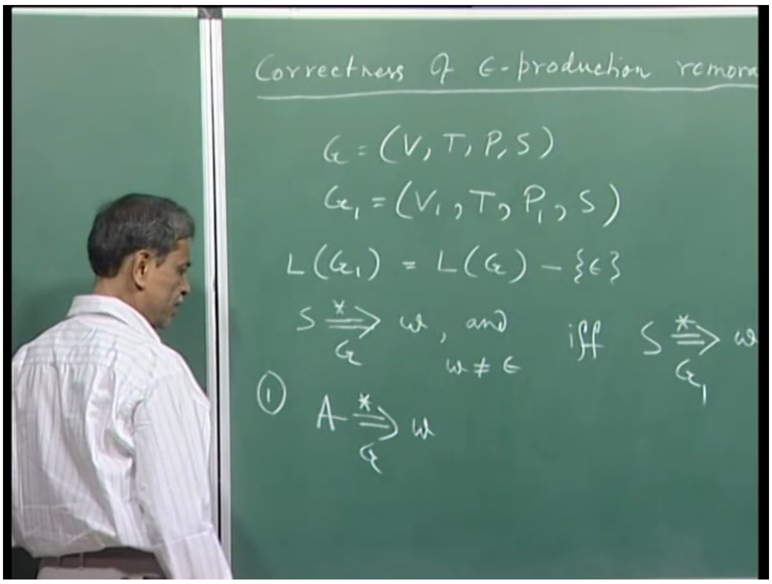
(Refer Slide Time: 44:36)



To establish this we will use our standard method that of induction and induction will be on the length of derivation. So as we see there are two things to establish. First one way that is if Sderives w and w is not epsilon then this and the reverse way. However instead of trying to show only for S it will be more convenient for the proof to establish something stronger and that isestablish the same thing not just for S but for every nonterminal.

So let us say what we want to show that Abeing a nonterminal and in G that derives some string w, from the nonterminal A you can get this terminal string w.

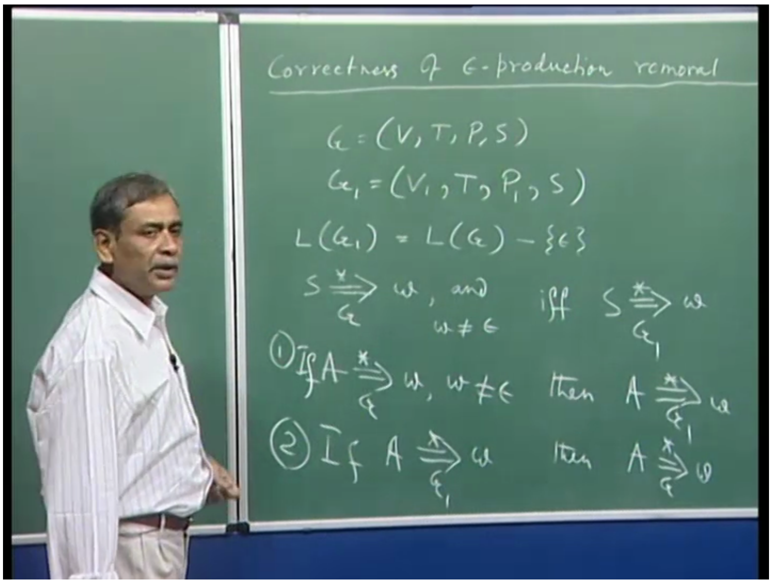
(Refer Slide Time: 45:42)



W is not epsilon then implies, so let me not use this symbol because that might be confusing with our derivation symbol. So let me write if this is the case thenA would generate in G 1 w and the other way we would like to show that if A generates or derives in G 1 the string w then A derives the same string in G as well.

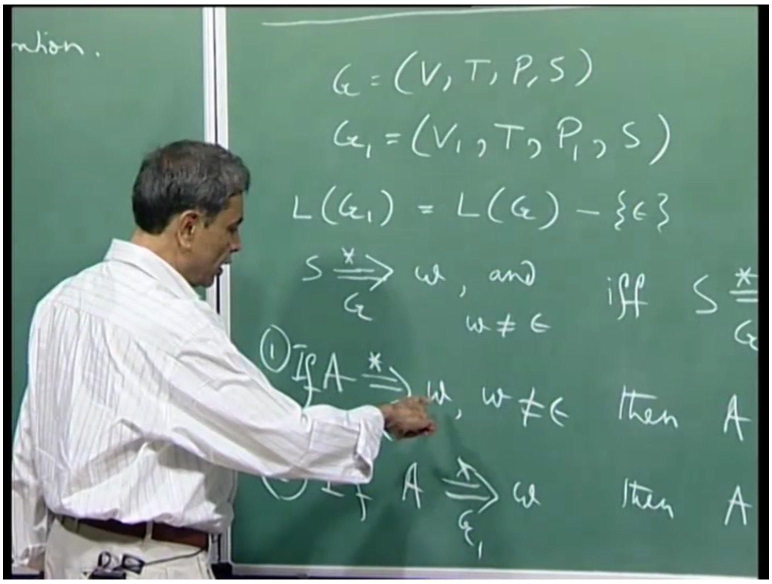
It is not difficult to say that if we prove this for all nonterminals A then of course, I mean not only it is not (trivi)difficult, it is obvious then we necessarily prove this because S is one of the nonterminals. So let us try to establish these two separately.

(Refer Slide Time: 46:57)



Now we come here. As we said the proof of 1. We will carry out this proof by induction on the length of derivation. Here hypothesis is that we are deriving w inG and let us say that what we are trying to show that for every n, n equal to 1, n equal to 2 and so on that if there is a derivation of length n, thisstatement will be true for allderivations of length n. Basically the induction on length n.

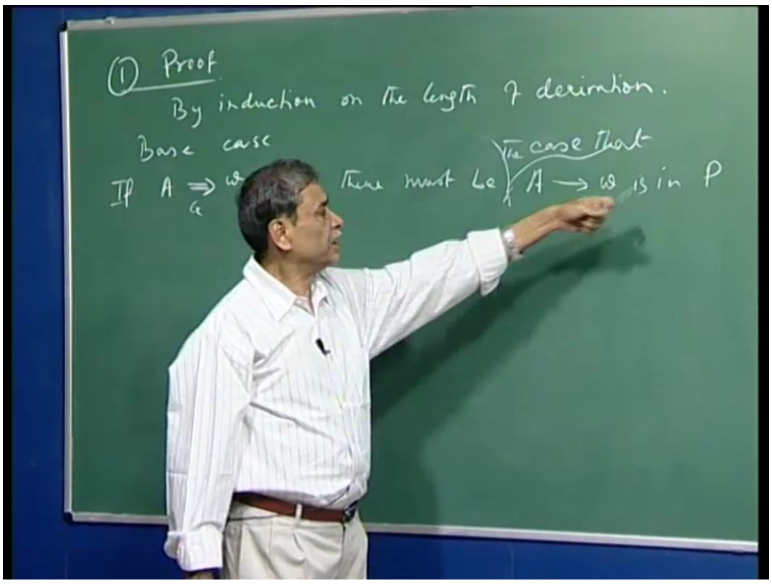
(Refer Slide Time: 48:00)



So what is the base? Base is some nonterminal derives in one stepthe string w in G. What does it mean? How can you derive in one step in G thestring w? That means there must have been, so let me say if in one step we derive from A the string w then there must be A goes to w in theset of productions P.

So there must be the case, right, that A goes to wis inP, is not it? This is clear that if we derive in one (sep) step some string that meanswe can use only one production and therefore that is the production we must be having in the set of productions P.

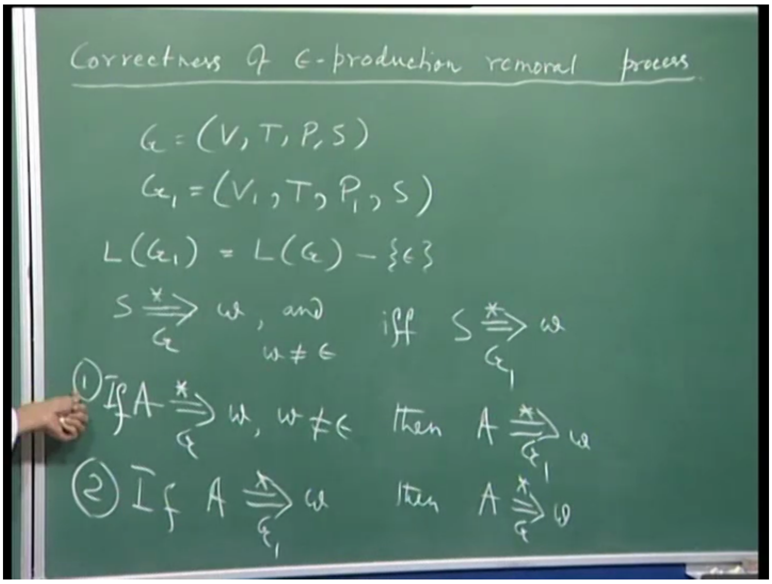
(Refer Slide Time: 49:23)



Nowif younotice that w is we have assumed we areshowing this and w is not epsilon. So therefore this production would not have been removed, right? The process that we discussed of gettingset of productions for this new grammar G 1, that removed all the epsilon productions and added some other productions.

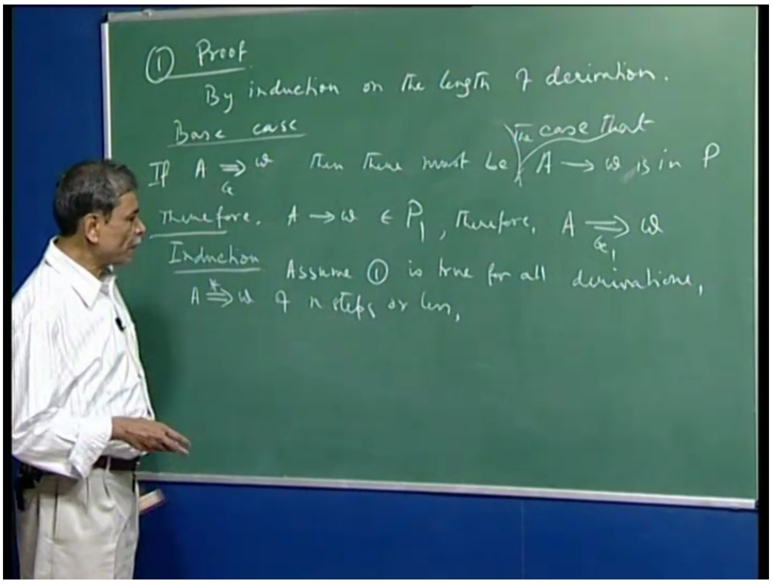
So this particular production would survive for G 1 as well. Therefore A goes to w is in P 1 as well and therefore it means that A in one step derives in G 1 the same string w. So this takes care of the base case for case 1 here.

(Refer Slide Time: 50:38)



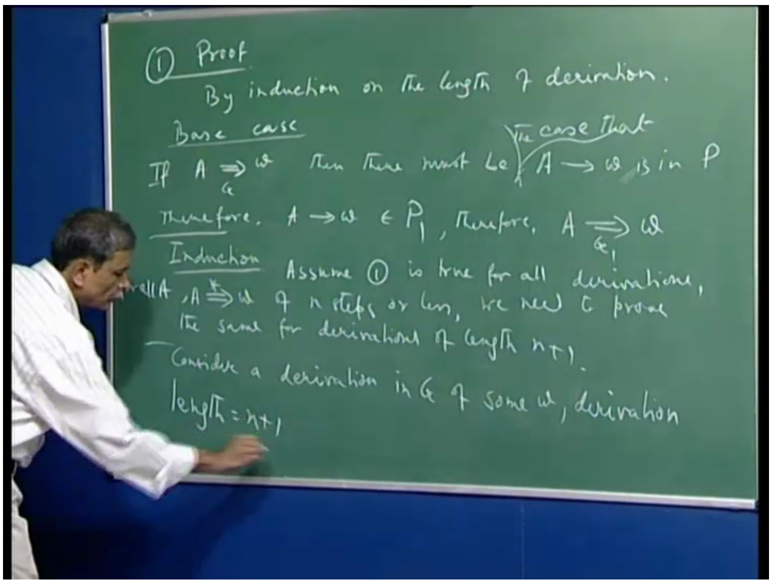
Now what is the induction step? Induction step is we assume the induction hypothesis which is that assume 1 is true for all derivations of n steps or less as well as for all A. So it is a kind of simultaneous induction that we are carrying out. If thisis general A, for any A we prove this. So in particular we proved the base case for every nonterminal and now we are carrying out the induction step. So 1 is true for all derivations of n steps or less.

(Refer Slide Time: 51:59)



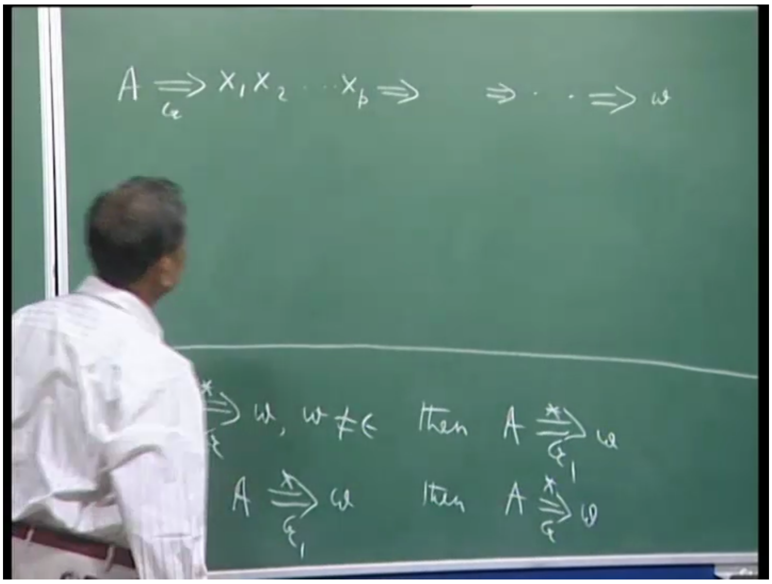
We need to prove the same for derivations of length n plus 1. Nowas I said that it is not only true for all derivations of single symbol A. So we should write for all A, right? So suppose we assume the induction hypothesis then we need to prove for n plus 1.Induction hypothesis or link forderivations of length n or less. And this step is also actually the induction is fairly simple. This carrying out this step. Solet us say consider a derivation in G ofsome w let me say, this derivation being of length n plus 1.

(Refer Slide Time: 53:46)



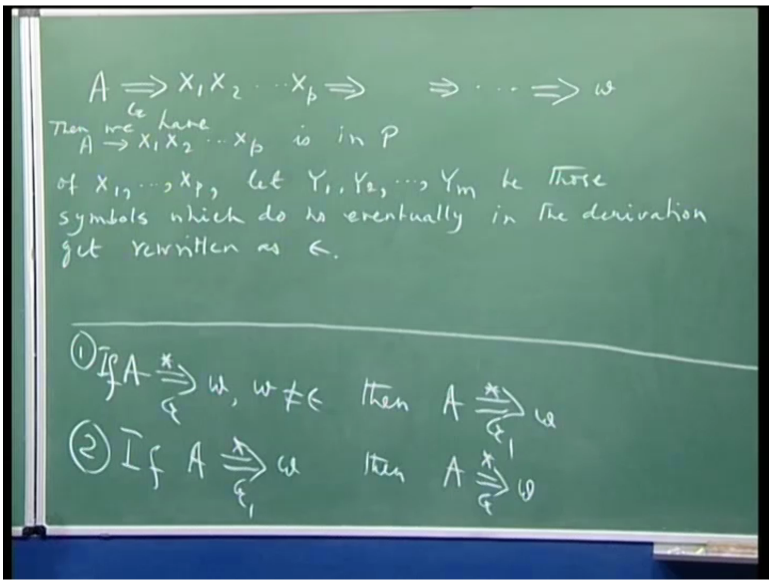
So such a derivation of length n plus 1 of some w starting from A, let us say right in the beginning in G the production that we used was X 1, X 2 and X p and then we have other steps finally leading to w. So what we are saying is that here I have in steps and the first step being A isrewritten by X 1 through X p.

(Refer Slide Time: 54:33)



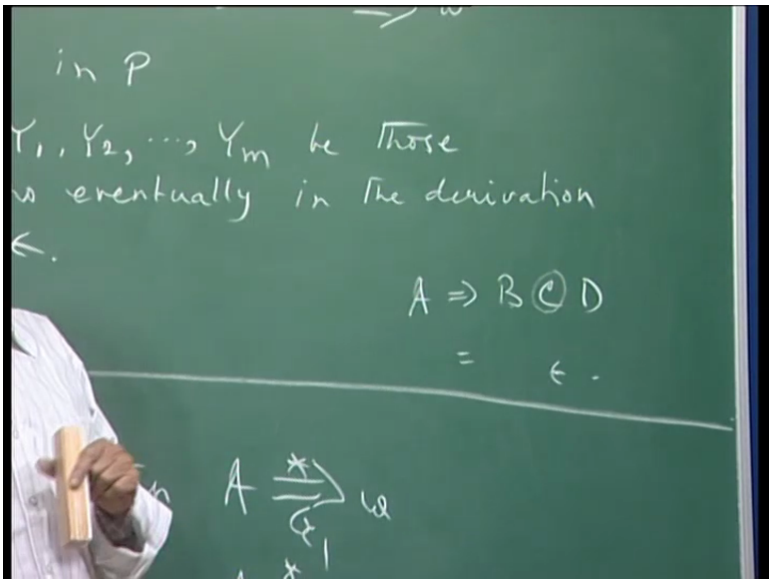
Now that must be because you can do this only because A goes to X 1, X 2, X p is a production in P, right? So ifthis is the case then we have that A goes to X 1 through X p is in P. Now what can happen is some of these X i's they generate null symbol, right? It is possible?So let us see of these X's, X 1, X 2, X p, so let us say of let Y 1, Y 2, Y m be those nonterminals or those symbols which do not eventually in the derivation get (re)rewritten as, is it clear what is happening?

(Refer Slide Time: 56:34)



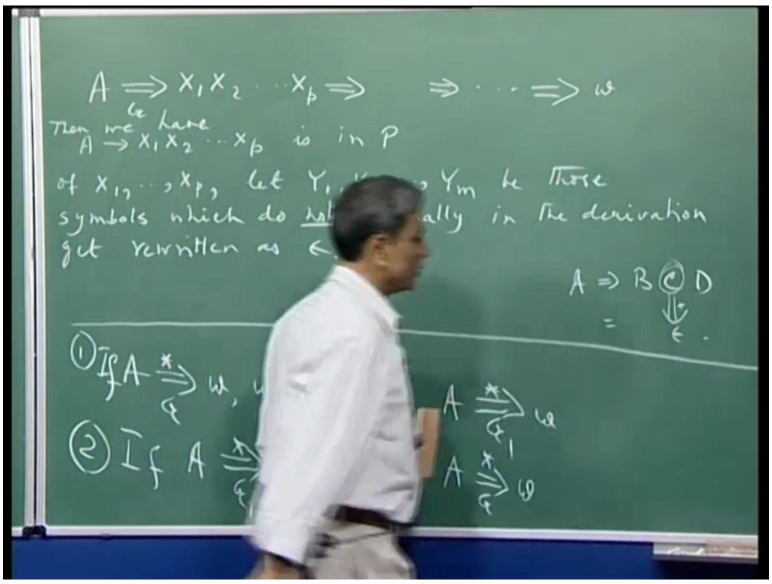
See for example that first production that you might have used is B C D, right? Now what may happen then during the rest of the thing C is a nullable nonterminal and C became epsilon. So nowother these two B and D, they generated (noll) non null strings.

(Refer Slide Time: 56:59)



So we are corresponding to B and D these are the symbols that we are saying that they do not derive epsilon.So those symbols which do not eventually in the derivation get rewritten as epsilon. And this is Y 1 through Y m arein the same order. So for example in this case my Y 1 would have been B and D would have been Y 2because C was getting rewritten as finally epsilon.

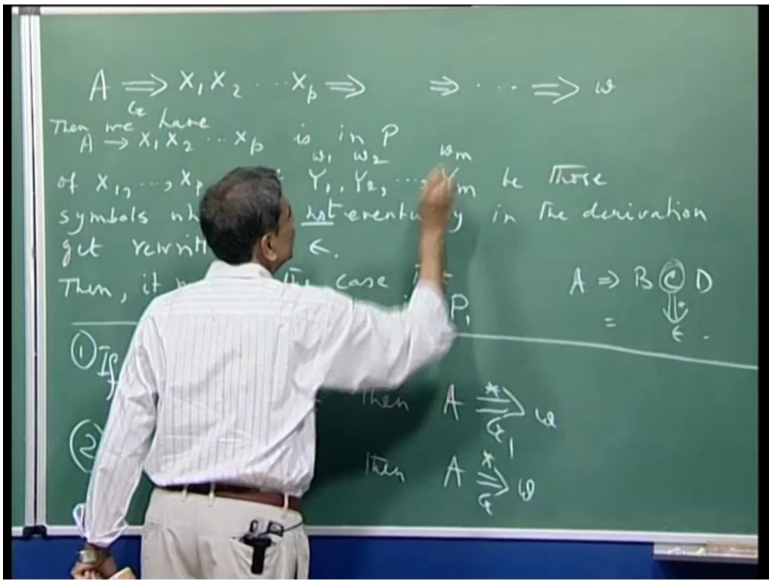
(Refer Slide Time: 57:45)



Soif that is the case then it must be the case that first of all A goes to Y 1 through Y m is in P 1 that is in the set of productions for the grammar G 1. Why? Because you know we will create all kinds of productions removing nullable symbols of G to get new productions for G 1 and therefore this will survive.

And here now it is very clear you see, let Y 1, Y m they are not getting rewritten as epsilon eventually. So each Y 1 through Y m they generate strings which are non-null. So let me say this string is w 1, this string generated by Y 2 is w 2, this is w m.

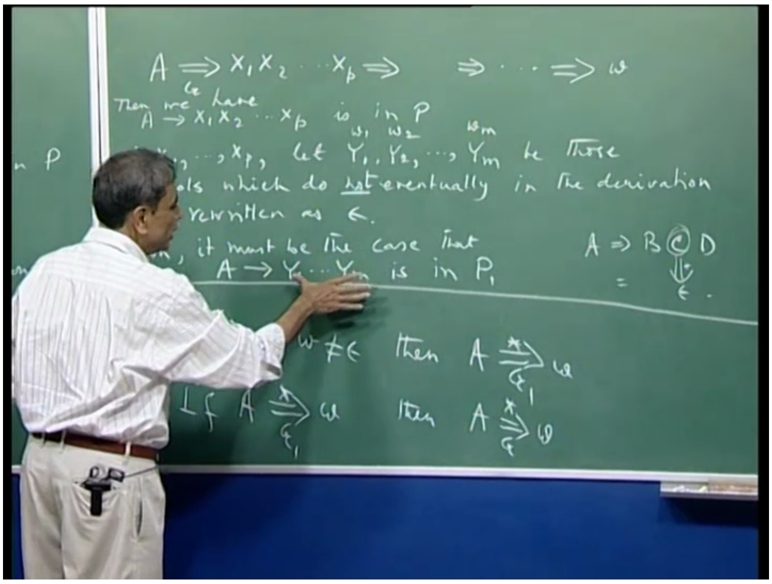
(Refer Slide Time: 59:00)



So then clearly w must be equal to w 1, w 2, w m. And in other words the process is such that Y 1 eventually is rewritten as w 1 non-null string, Y 2 as w 2 and so on. But the derivations for each of these to go from Y i to w i they must be using steps less than n.

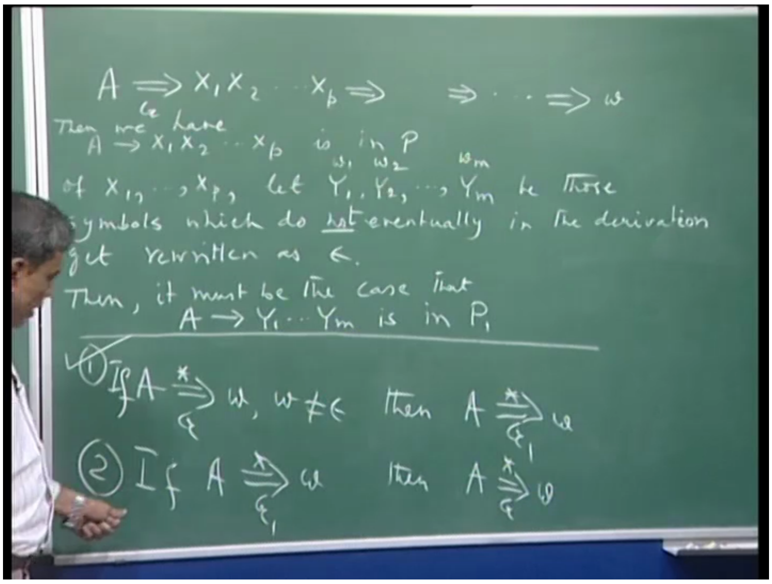
And therefore we can use the induction hypothesis to say that we will be generating the same string w in G 1 also becausethe idea isto show that we are generating in G 1 the same string w. We first usedthis production and then we used the derivation to obtain w 1 from Y 1, w 2 from Y 2 and so on.

(Refer Slide Time: 01:00:02)



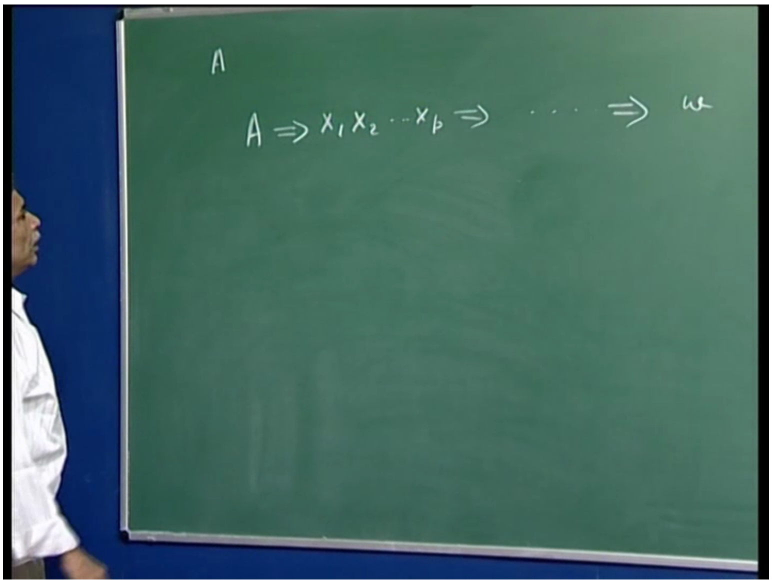
And therefore finally I will get w 1 through w m which is nothing but w. So we have completed this and to show 2 the idea is kind of very similar. Now again assume through induction hypothesis I am not proving the base case. Just clear that we can prove the base case here too very simply. So suppose we havethis assumptionthis result is true for all derivations of length n or less for the grammar G 1.

(Refer Slide Time: 01:00:44)



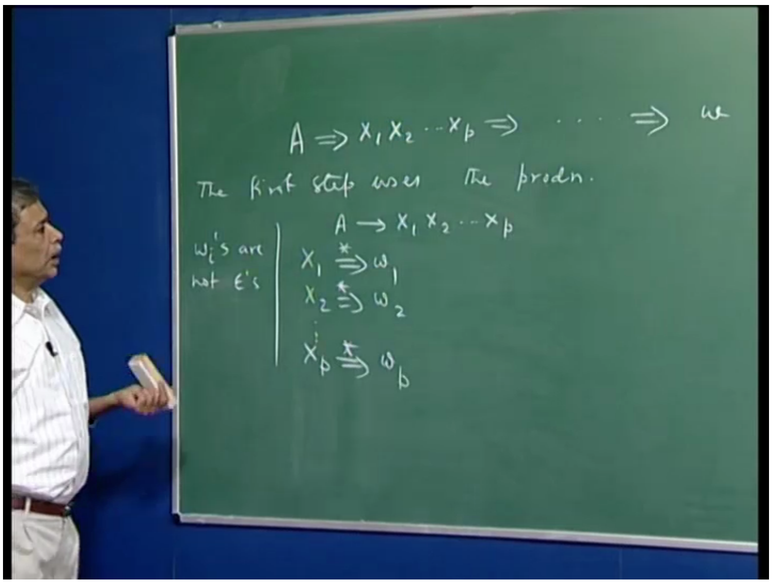
Then I need to show that suppose I derive some w from (non) some nonterminal A in n plus 1 steps in G 1 and I should be able to derive that same string in G also. So consider such a derivation starting from some nonterminal A. Sothe first step that will happen is A will be rewritten by using a production of G 1, right, because we are talking of derivations in G 1. So let us saythe first step happens is X 1, X 2, X p and then I have n more rewriting steps to eventually get w.

(Refer Slide Time: 01:01:44)



Sothe firststep uses the production A goes to X 1, X 2, X p, right? And let us say that eventually X 1 gets rewritten in this derivationas w 1, X 2 as w 2 and this X p gets rewritten as w p. Remember thatnothingcan giveyou epsilon in the grammar G 1, right? So all of them will generate each of these w i's. So w i'sarenot epsilon, okay.

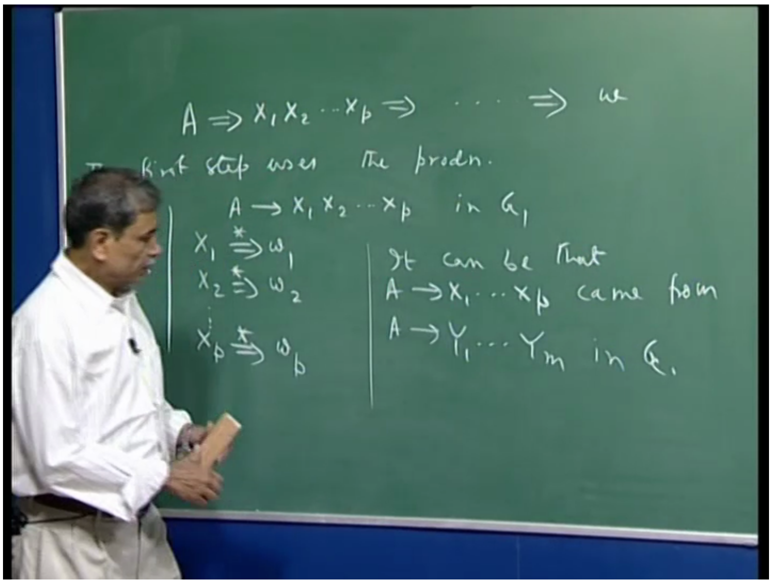
(Refer Slide Time: 01:02:54)



So this is a production in G 1. Now it could be that same production is there in G. So then we have no issue. Weshow this in G we use that same production to come to this point and then use the induction hypothesis. But what might happen that the production that you are using came from a larger production.

With (produ)A on the left hand side and the right hand side there were some more symbols which were removed because they are nullable symbols. Socan be that A goes to X 1, X p came from A goes to Y 1 through Y m in G, right?

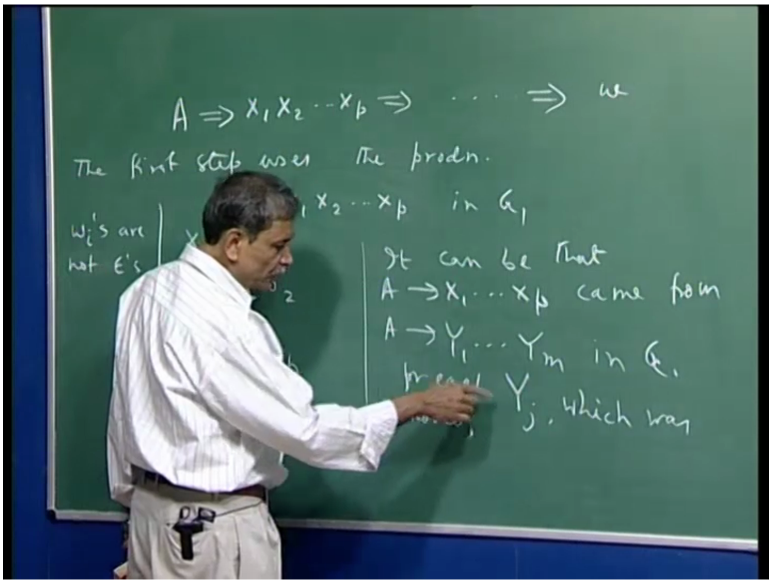
(Refer Slide Time: 01:04:06)



Remember some of these Y i's were removed to get this production. So m is larger than p, okay. So now we want to show that same w can be derived in G. So what we do as the first step ofthis derivation, we use this production and those symbols which were removed to get this particular production from the production of G to a production in G 1, those removed symbols must be the ones which are nullable symbols.

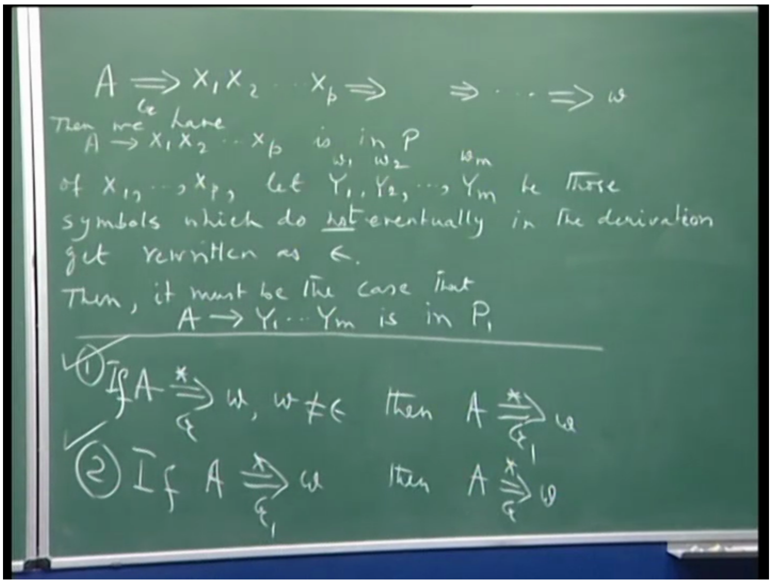
So what I would do is those symbols here which are nullable for each Y j which got removed to obtain this particular production whichwas removed. We start with this production.

(Refer Slide Time: 01:05:22)



Those Y j's we rewrite as epsilon. I know that I can do that because those are nullable symbols. So eventually what I will have is after some steps that in G itself I will have X 1 through X p and then wejust follow the steps of A. Now use the induction hypothesis for this part because each w i from X i, they would be obtained by using number of productions which is less than n. Therefore now we have completed the second step also.

(Refer Slide Time: 01:06:06)



And put together what we have shown that our process of getting a grammar from an old grammar such that the new grammar does not have any epsilon productions. At the same time it generates all non-epsilon strings which are derivable from the old grammar. That particular process is correct. And we will still have one more kind of simplification to do that is called removal of unit productions which we will do in the next lecture.