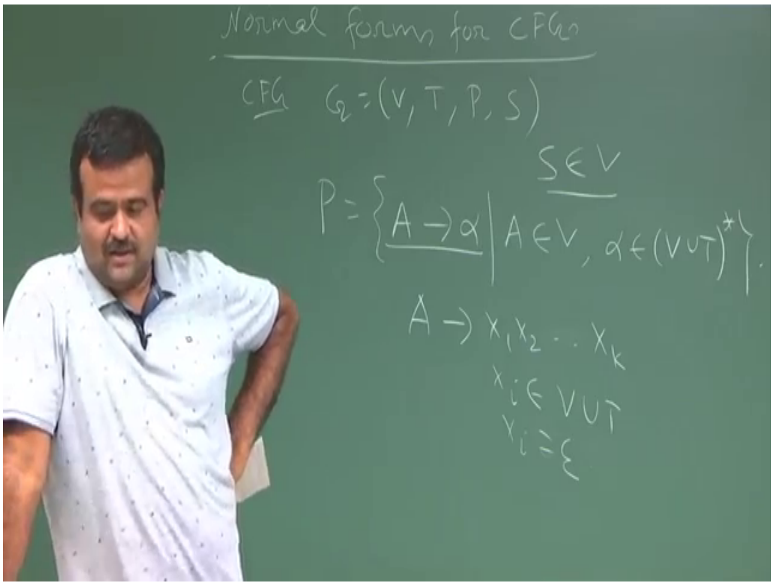
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 46
Chomsky Normal Form (CNF)

So, we will talk about normal form in context we gamma. So, there are 2 normal form, one is Greibach normal form and one is Chomsky Normal Form CNF GNF. So, let us start with the normal form what are those normal forms?

(Refer Slide Time: 00:35)

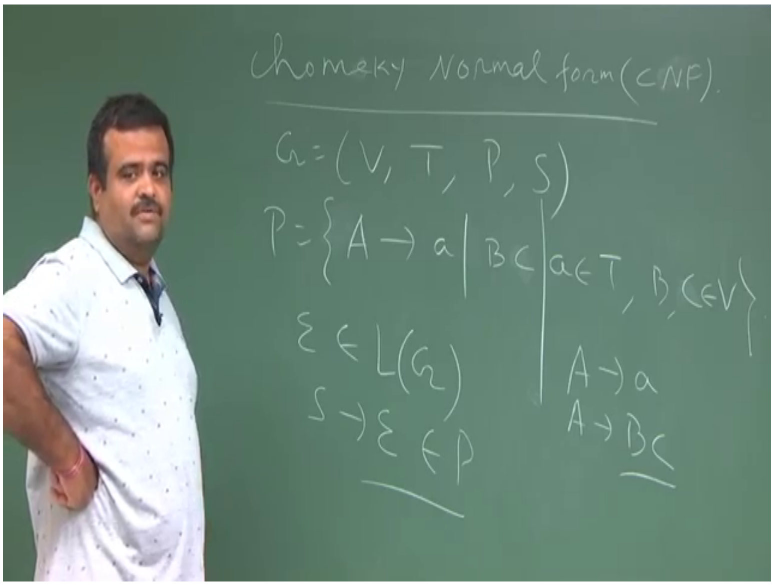


So, suppose you have given this context free grammar G, which we know it is V, T, P, S; where V is the set of all variables, and T is the set of all terminals, and S is the special variable, which is called start variable S belongs to V, S is a starting variable or start variable. And, P are the set of all productions, I mean these are the rules and following these rules, we start from S and we derive the expression. And, the P will be of this form P is consists of all the rules of this from this where, A is a variable and alpha is coming from like this ok.

So, this type of productions we have. So, variable is going to some X 1 X 2 X k say where, x I are either variables or terminals or it could be m it could be epsilon also. So, this type of products cells are there in our P, the P R production set ok.

So, now we will define 2 set of I mean 2 form of normal form, one is CNF, another one is GNF.

(Refer Slide Time: 02:18)



So, this is called Chomsky normal form, Chomsky normal form in sort it is called CNF ok.

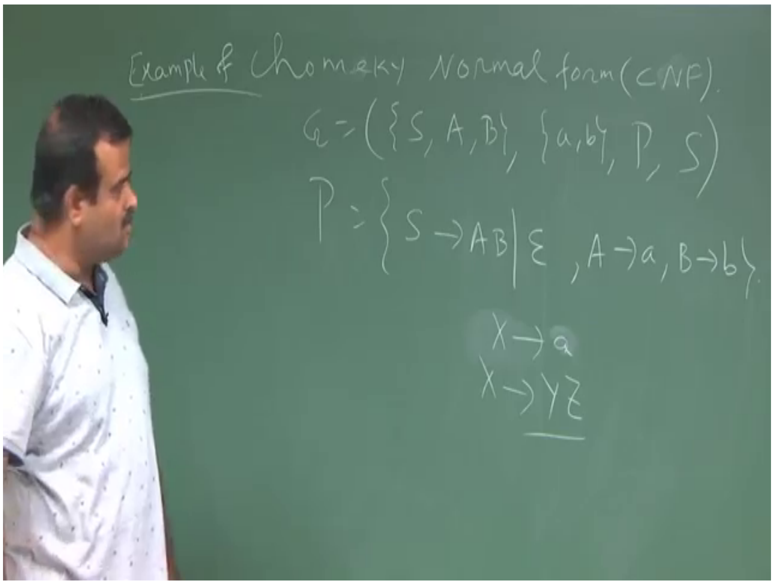
So, the normal form is this is a grammar we have V T P S, normal form means this P the productions will be in some particular form. So, for this case production will be of this form, A is going to alpha or A is going to a or a is going to B c, where B c are 2 variables and a is the terminals.

So, all the productions so, P consists of this type of productions only, where a belongs to a terminals and BCR from the variables capital C, this is capital C. So, if our all the productions are of this form and if epsilon is belongs to the language, G, then we have a production S is going to epsilon this production will be there in P. Otherwise, if epsilon does not belongs to this then this is because this is a, this is not a T star this is the terminals.

So, if our production all the productions are of this form then we call this is a CNF, Chomsky Normal Form like, and this is either that all the productions either will be in this form a is going to some terminals, or A is going to the product of I mean concatenation of 2 variables ok.

So, we can take an example or if epsilon is a string which is generated by this gamma, then we have a rule S is going to epsilon in P. So, this is called Chomsky normal form CNF, if our all the productions are of this form then the grammar is called in Chomsky CNF ok, we can take an example.

(Refer Slide Time: 05:09)

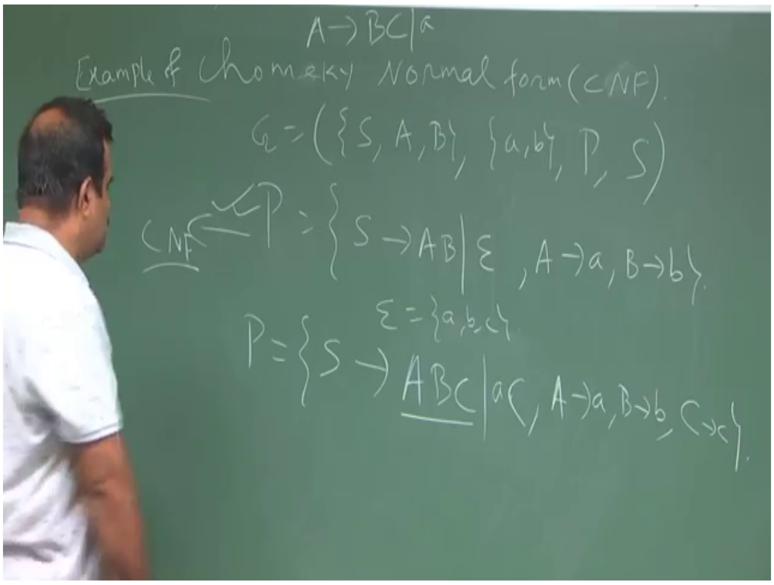


So, example of so, we take a grammar like this, we have 3 variables A, B, C and we have terminal say a, b small a b. P, we have to define and S is the starting variable ok.

Now, let us define the rules. Rules consist of this set say S is going to AB or epsilon. Because, we here our we want to generate epsilon as a string of I mean string language I mean string, which is generated by this gamma. This is 1 2 rules A is going to a and B is going to b that is all.

So, if we check all the gamma, all the rules all the productions in this gamma, all the productions are of this form either a variable is going to a variable say X is going to a terminal yeah, or X is going to Y Z, the 2 variables. And, if epsilon is there; that means, epsilon will be generated by this. So, this is one example of Chomsky normal form, but if we take another language with the same variables and the grammars are like this, rules are like this.

(Refer Slide Time: 06:56)

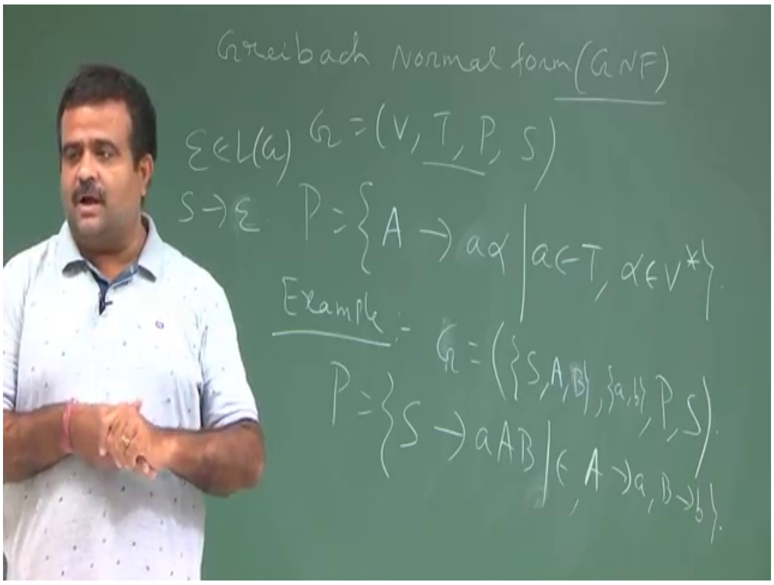


So, this is Chomsky normal form CNF but let us take an example which is not in CNF like this one.

Suppose, we have a rule S is going to ABC, ac, A is going to a, B is going to b and capital C is going to small c. Here, sigma is a b c. Now, this is not in CNF why, because this is having 3 variables for CNF we need to have only 2 variables. So, CNF will be of this form A is going to BC type of thing or a is going to some terminals.

So, this is the only form of our CNF, but this is not in this form. Even this 1 C, this is also not in this form. So, this is not a CNF, but we can make this to be CNF given a any gamma, we can convert into a Chomsky Normal Form. So, that we will see before that, let us define the another normal form which is Greibach Normal Form, GNF ok.

(Refer Slide Time: 08:34)



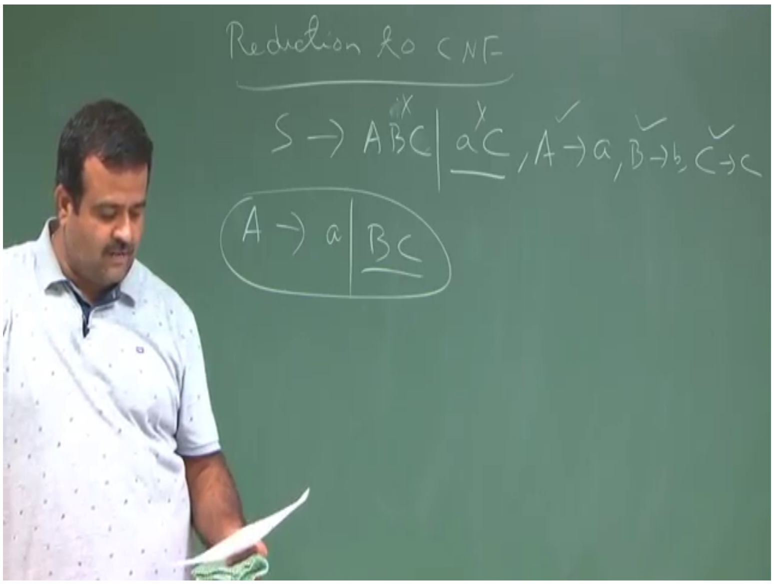
So, this is another normal form Greibach Normal Form; grei GNF ok. So, when we say a grammar is in GNF again these rules are in particular form like, if is like we have a grammar like this. And, suppose the rules are of this form P consists of all the A, variable is going to alpha sorry a alpha, where a is the terminals, but alpha is a string of variables, alpha is a string of variable then it is called in GNF.

This is call this for example, if we take a grammar like this. So, S P on suppose we have a 3 variable say S A B and we have a b this is the T, P and S. And, suppose this P consists of this rules. Again, if epsilon is belongs to L of G, then we have a rule S is going to epsilon that is there ok.

Suppose, we have a rule like this, S is going to say some a AB or epsilon and A is going to a, B going to b, this is in the GNF form. Because, this is this is going to a alpha alpha is just string of variable alpha would be epsilon also, because in this case a is going to a means here alpha is epsilon, but epsilon is allowed. So, that is the valid form. So, this is one example of GNF.

So, we will see more on GNF in the next class, but today now we will discuss how given a given a gamma, how it can convert into a CNF? So, reduction of grammar to the normal form CNF so, that will do now.

(Refer Slide Time: 11:46)

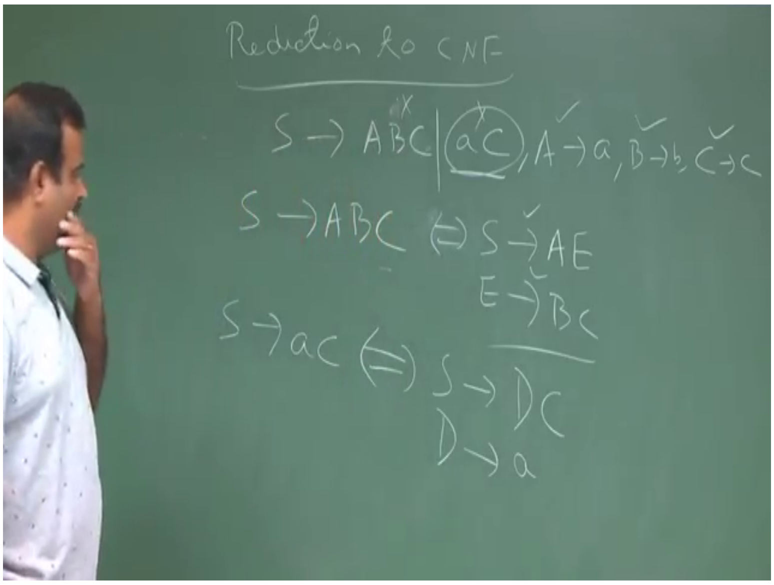


Reduction to CNF; that means, given a grammar how we can convert into a into the normal form, like normal form means there are two normal form, I we will discuss that first one the CNF first then we will discuss that GNF given a any grammar ok.

So, how we can do that? So, for example, in suppose we have taken this example, S is going to ABC or ac or A is going to a, B is going to b, and C is going to c ok. Suppose this is our given grammar and we need to convert into CNF, this is not in CNF, because this CNF means what CNF means the every variables would go to either terminal a or product of two variables. This consists of no terminals. So, this is called this is the CNF normal form, but here this is ok, this is ok, this is, but this is not ok, because it consists of a terminal and a variable. So, that is not allow in CNF. So, this is the CNF and this is also not this is capital C, this is also not ok.

So, these 2 are not ok. So, we need to somehow convert these 2 in the CNF with the help of some other variables like, that we are allowed to we can introduce some other variables to do that. So, how to do that?

(Refer Slide Time: 13:44)

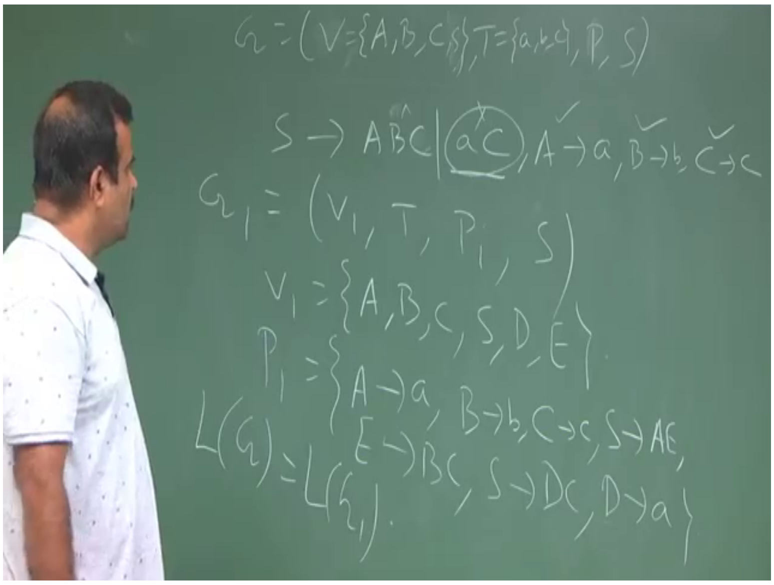


For example how to convert this, S is going to A B C, how we can convert in CNF with the help of some, we can introduce some other variable here.

So, what we can do, we can introduce a second variable E, like E or D anything yeah we can introduce a E here or. So, S is going to A E, like this will be replaced by, this will be replaced by this is same as S is going to A E and E is going to B C. So, this production will be replaced by two more production with the help of another variable, S is going to A E and E is going to B C ok.

And, this is now this is CNF this is CNF form. Now, also for this one like S is going to a c what we can do we can also take help of some other variable D. So, S is going to D C capital C D C and D is going to a. So, this is the way we can so this is in CNF, this is in CNF so, both are CNF. So, basically this is the grammar we have given so, where G is V; V consists of ok, V which is A, B, C and we have a T, which is also small a, b, c and then we have a sorry A B C and S also, we have S, we have P and we have S ok.

(Refer Slide Time: 16:05)



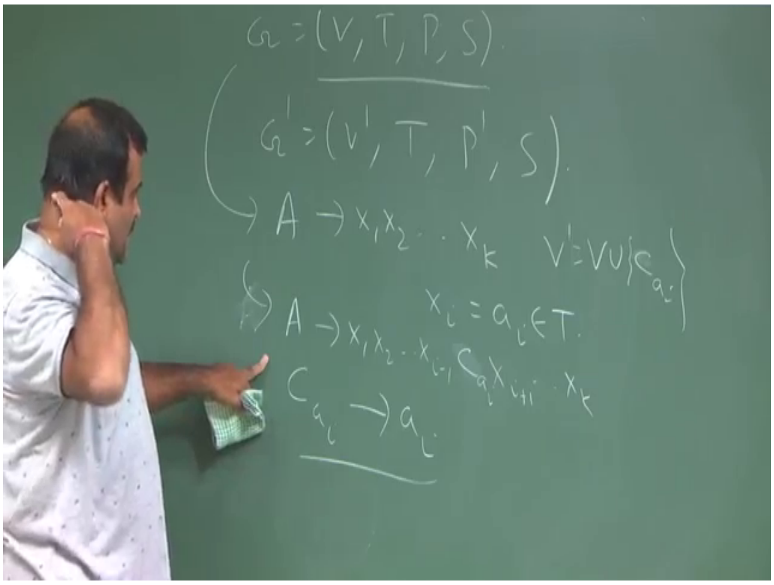
Now, this will be replaced by the grammar G 1 this is G gamma, this will be replaced by the grammar G 1 equivalence grammar G 1, which is in the normal form G 1 which is V 1 T will be same, T 1 S will be same we are not changing the S.

Now, V 1 is having all the new variables along with the old variable A, B, C S and we have introduced D and E. And, T is same S is same only P 1 will be changing, first of all we will put all the gamma, all the productions which are in the already in the normal form that will first put like, A is going to a, B is going to b, C is going to c, these are already in CNF so, will first put those rules.

And, next we are going to replace this by S is going to just now we have seen A, E and E is going to B, C, this one you have replaced S is going to D C and D is going to D is going to a that is all. So, this is the corresponding grammar for this. And, we can see that this language generated by G will be same as language generated by G 1 ok. So, this is two an example.

But, in general we can prove this, we can show this is these are equivalent. So, how to show this? So, that will try. So, given a grammar how we can reduce to a normal forms, particularly CNF ok.

(Refer Slide Time: 18:10)



So, we have given the grammar G. Now, so, P will be of this P this is a any gamma. So, it is not in a normal form. So, we will construct the grammar another grammar G dash, we may introduce some more variable terminal is same, yeah terminal set will be same, S will be same we are not changing the S ok. Now, first suppose we have given a rules like this here A is going to some X 1 X 2 X k say.

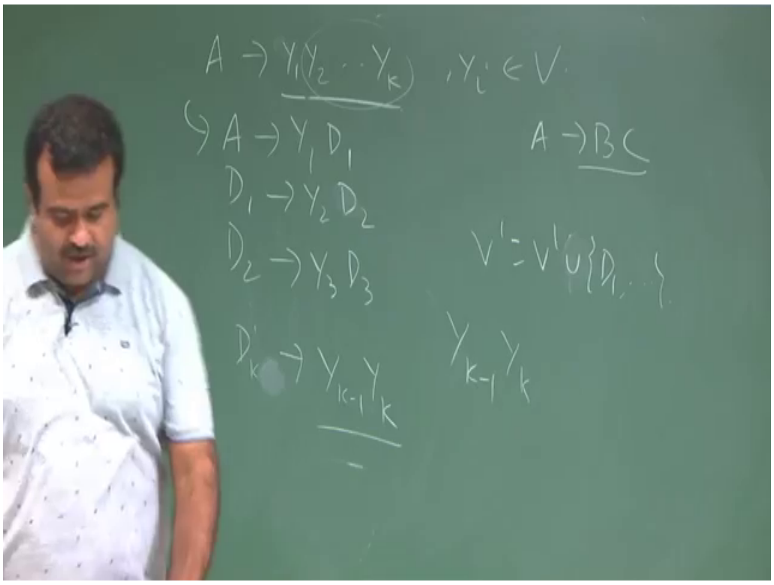
Now, suppose this is a rules. So, some of the X is will be coming from T and some of the X I X I will be coming from V. So, some of the X i's are the variable some of the X is will be at the terminals. Now, we want to if it is terminal then you want to bring it to a variable we want this all X i should be a variable, because we will see that. So, for that suppose this is this X L is a terminal say X L, X i is a terminal X i s is here a i.

Then what we do, we just replace this, this is equivalent to 2 rules X 1 X 2 X i minus 1 and then C of a i capital C C of a i or we can say, yeah in the node it will be this notation C of a i then X i plus 1 dot dot dot X k.

So, C of a I we have introduced a new variable here. So, our V will be now V prime will be V union of C of capital C of I some notation some new variable we are going to introduce. And, C of a i is going to yeah ok. This is ok, because this is allowed in the CNF. Only thing in CNF we should not have this combination of the combination of the terminals and the variables is not allowed.

So, that is why if there is a variable there is a terminal, we have to replace that by variable and that variable will go to that particular terminal. So, this way we do. So, if we are more such terminals, then we will do this replacement like this ok.

(Refer Slide Time: 20:58)



Now, finally we have the form like this A is going to some Y 1 Y 2 Y k and where each of these Y i S are variables, after doing this job. And, say remaining are in of this form CNF form remaining rules including this newly added rules ok.

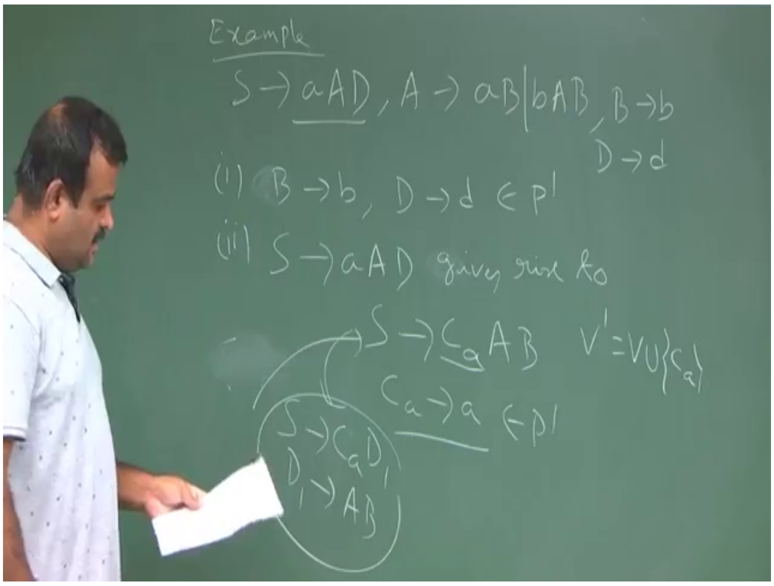
Now, these are all variables. Now, how to convert because we know we have to convert these also, because we know that our CNF is of this form a is going to B C only 2 variables here you have many variables. So, what about to transfer? So, this will be again transfer as this I will follow the notation like this. So, A is going to yes so, A is going to say some Y 1 D 1. This whole thing now D 1 is going to Y 2 up to Y k, but then again will take help of some D 2.

So, we are keep on adding this. So, now, this V prime is V prime union, D 1 D 2 how many are there? So, D 2 then D 2 will be going to D 2 D 2 is nothing, but the D 2 is the D 2 (Refer Time: 22:30) D 3 like this. So, dot dot dot then so, lastly we have Y k minus 1 and Y k.

So, we have D k. So, D 1 is going to this. So, D k minus 1 is going to sorry D k plus 1 now D k D k D k is going to Y k minus 1 and Y k. So, like this we will do ok. So, if this is not even then we can have Y k then that is going to epsilon.

So, this is the way we first convert everything into the variable. And, then once we have a string of variable that will replace by this is the way. So, we will take a quick example we will take an quick example.

(Refer Slide Time: 23:47)



Suppose, we have the grammar like this a is going to AD sorry S is going to AD, A is going to a B or b A B, and B is going to b and D is going to d. So, this is the grammar.

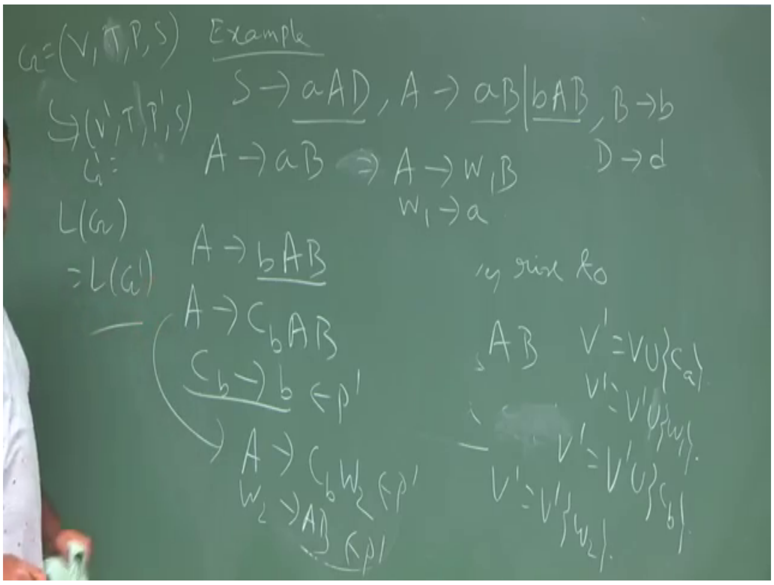
So, we have corresponding variables S a B C a B D ok. So, now, first we have to remove the null and unit productions, because that way that is the redundancy we have. So, we can remove the nulls and unit production, but here there is no null and unit production there ok. Now, what we do? So, we can check which are in the already in the CNF.

So, first of all B is going to b, and D is going to d, these are already in this form. So, we will add this into this new rules and then we take this one. So, this one is mixture of variable and terminals. So, what we do? We just replace this by this gives rise, S is going to C a A B, where C a we are going to introduce there. So, B prime is B union C a C a is the new variable.

So, now this is all variables, now we introduce S is going. So, this is we can do it in second phase where we have all variables then we can do it, but here also we can do, but anyway we can try this. So, yeah so, anyway this we can finish it off. So, this is the rule and C a is going to a. So, this we can add in P prime and for this we have to take help of another variable, S is going to some this C a and some D 1 and D 1 is going to A B.

So, these 2 combined will give us this ok. This is done, now we take these two; these two are not in normal form.

(Refer Slide Time: 26:29)



So, A is going to a B. So, this we can do one thing we can change this to a variable another variable. So, this will replace as this will replace as A is going to some W 1 B and W 1 is going to a like this. And, similarly we can do it for this also. And, every time once we are introduced a new variable here we need to add this in this variable set.

So, you have to add D also D 1 D 2 like this. So, again A is going to b A B. So, this will be replaced by like, we have to change this to variable. So, this is going to be C b A B and C b is going to B. So, these we have to add to the rules and here we have to add this C b here ok.

Now, this is ok, but this is not ok. So, for this we have to change this to C b some if we have not used W 2 we can be use here and W 2 is going to A B. So, these are we are going to add a P prime and V prime will be V prime W 2 like this. So, we will keep on doing this and we get this thing yeah, ultimately we will get the everything in the normal form ok.

So, and we can prove that this reduced grammar like if this is the grammar V P V T P S. Now, this is reduced to the grammar V prime T P prime T P prime S. And, we can show that this is G and this is a G prime, we can show that grammar generated by G is same as grammar generated by G prime. So, the proof will be in the lecture note.

Thank you.