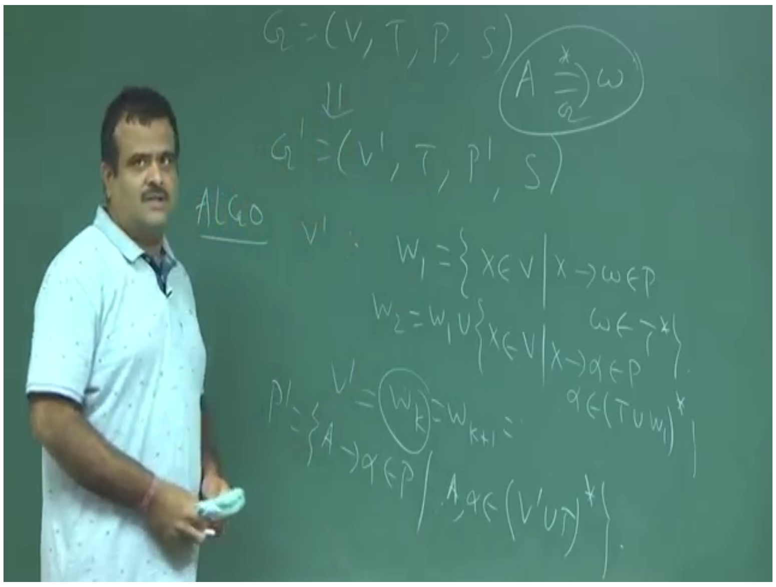
**Introduction to Automata, Languages and Computation**
**Prof. Sourav Mukhopadhyay**
**Department of Mathematics**
**Indian Institute of Technology, Kharagpur**

**Lecture – 44**
**Algorithms to Construct Reduced Grammar**

So, we are talking about the simplification of the context free grammar.

(Refer Slide Time: 00:28)



$G=(V,T,P,S)$
$A \overset{*}{\underset{G}{\Rightarrow}} \omega$
$G'=(V',T,P',S)$
ALGO
$V'$: $W_1=\{X\in V \mid X\to\omega\in P, \omega\in T^*\}$
$W_2=W_1\cup\{X\in V \mid X\to\alpha\in P, \alpha\in(T\cup W_1)^*\}$
$V'=W_k=W_{k+1}$
$P'=\{A\to\alpha\in P \mid A,\alpha\in(V'\cup T)^*\}$

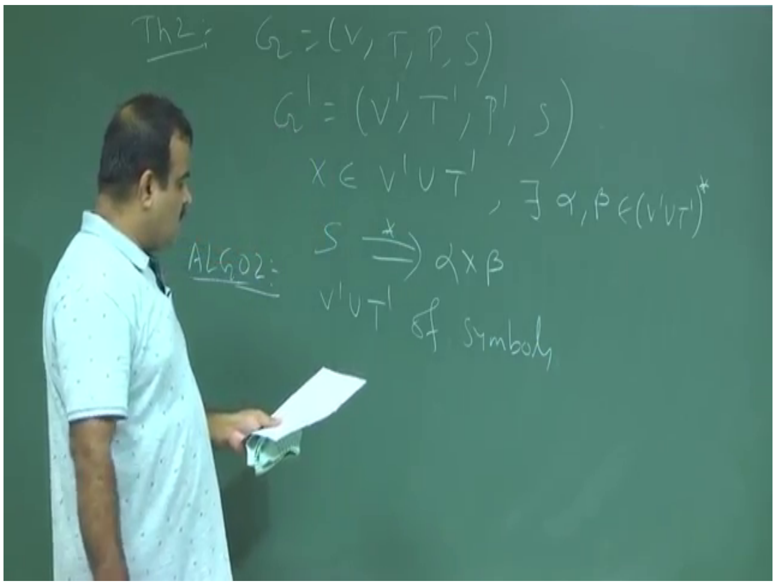
So, we have seen on method ALGO 1 to simplify the context free grammar that is basically the given a grammar. So, we are reducing this to T, we are not changing. So, V prime we are constructing recursively like this. So, we are considering we are first generating the W 1, W 1 is the set of all the variables which are directly going to the sum of the string of terminals. X is going to W where W is a string of terminals; so we are capturing all such X.

Now, once we generate a W 1, we generate W 2 which is W 1 unions then all such X again from V such that now we have a production like this in P where alpha is now T union W 1 star, like this. So, if you continue like this then we will see that at some point of time, it will stop, I mean W W W 1 like this so, this is our V prime. Now, once we get the V prime will construct the P prime like this. So, you take all the A all the rules from P is such that A alpha all are belongs to V prime T star, ok. So, only those string consists of the variable from this V prime and the terminals, ok.

Now, this was the ALGO 1 this way you are constructing was ALGO 1. And we have seen in the last class that if we construct like this then every terminal will go to sorry, every variable will go to some terminals, the way we constructing this.

Now, will construct another algorithm which is ALGO 2 where we will only take those useful I mean those variables or terminals which are reachable from S, because ultimately you have to start for this is the first algorithm is giving us which are going to W, but ultimately have to start from S, S is the starting variable. So, we will concentrate on those derivation will concentrate any of those steps Y which are reachable from S basically, this is the other way round.

(Refer Slide Time: 03:26)


Th2: $G = (V, T, P, S)$
$G' = (V', T', P', S)$
$X \in V' \cup T'$, $\exists \alpha, \beta \in (V' \cup T')^*$
ALGO2:
$S \xRightarrow{*} \alpha X \beta$
$V' \cup T'$ of symbol

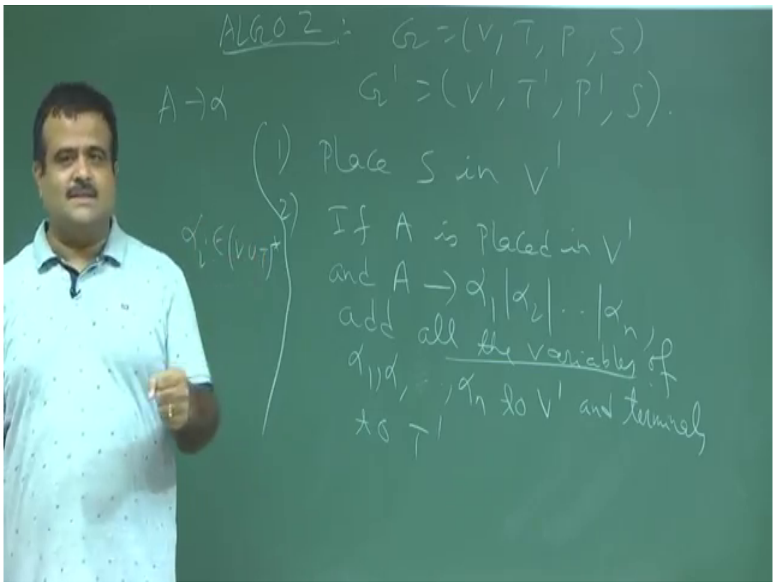
So, that we will see; so that we will write in a theorem form; this is the theorem 2 so, given a grammar sorry, T given a grammar we can efficiently find an equivalent grammar G prime where everything is changing here V prime, T prime also T prime S will be remain same such that if X belongs to V prime, T prime then there exists alpha beta for which X derived this.

So; that means, we look at only those variables or the terminals which are reachable from S, because that is the language I mean we are talking about in terms of language. Ultimately, we want to we are reducing the grammar to get the same language. So, language means which is generating which is yield from S. So, that is the sense; so we are only concerned about those variables or those terminals which is in a sentential form which is in sentential form sentential form like S is S form A.

So, you can reach to that. So, that is the way. So, this is this we are to generate by which is called ALGO 2. So, this will generate by just for ALGO 2. So, what is ALGO 2, algorithm 2?

So, we construct, so, basically the V prime and T prime all the elements this is the, this of the symbols which are coming in sentential form of G. So, let us just write this ALGO, then it will be more clear.

(Refer Slide Time: 05:46)



ALGO 2: $G = (V, T, P, S)$

$G' = (V', T', P', S)$.

$A \to \alpha$

(1) Place S in $V'$

2) If A is placed in $V'$ and $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$, add all the variables of $\alpha_1, \alpha_2, \ldots, \alpha_n$ to $V'$ and terminals to $T'$

$\alpha_i \in (V \cup T)^*$

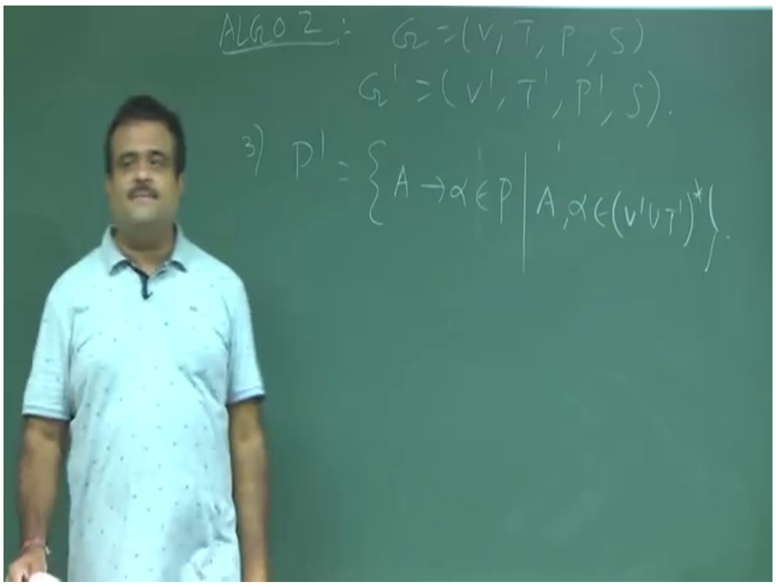
So, this is the ALGO 2; so we have given this. Now we have to constrict this, ok. So, this is the ALGO the first step is we first place S in V prime because S is the I mean we are looking for all the variables and terminals which is a reachable from S, I mean which are in the a part of the sense a sentential form of sentential form which are reachable from S basically, ok.

Now in the second step what we are doing if A is placed in V prime and if we have a production like this alpha 1 alpha 2 alpha n then we are going to add all the variables, all the variables, all the variables of this alpha is alpha 1 alpha 2 this will consist of some variables and the terminals say alpha each of these alpha is, is a string of variables and terminals.

Now, if we know a is already in V prime and if there is a rule A is going to say alpha. Now alpha consists of some variables, some terminals. So, we are going to put each of that variables and terminals into we are going to put each of those variables in V prime and each of the terminal in T prime.

So, that is we are going to write, the each of the all the variables of this in V V prime and terminals terminals to T prime, because those are basically reachable from those are part of the sentential form, sentential form. So, those are basically we can derive those from S; so that is all. So, this way we construct V prime and T prime and the production is basically which is all involves this V primes and T primes that is the step 3.

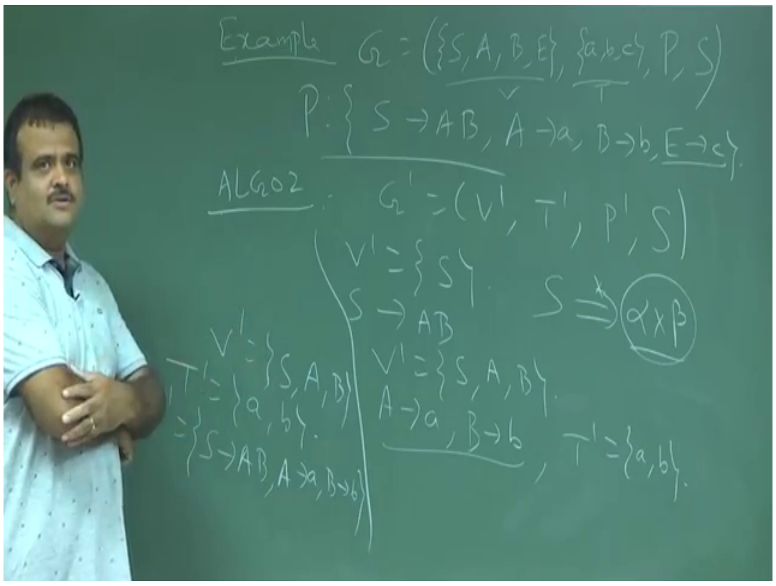
(Refer Slide Time: 08:37)


ALGO 2: $G = (V, T, P, S)$
$G' = (V', T', P', S)$.
3) $P' = \{A \to \alpha \in P \mid A, \alpha \in (V' \cup T')^*\}$.

So, this is to construct the P prime. So, P prime we take like this is the set of I mean A is going to alpha which is a production in P such that A and alpha coming from V prime is the string of new or we only consider the productions which are involved this new terminals and the new variables.

Not new I mean that we are removed basically we are deleting some of the variables and some of the terminals. So, that way we, we construct this; so this is called ALGO 2. Let us take an example, then it really more clear.

(Refer Slide Time: 09:42)


Example $G = (\{S, A, B, E\}, \{a, b, c\}, P, S)$
$P: \{S \to AB, A \to a, B \to b, E \to c\}$
ALGO2: $G' = (V', T', P', S)$
$V' = \{S\}$
$S \to AB$
$S \overset{*}{\Rightarrow} \alpha X \beta$
$V' = \{S, A, B\}$
$A \to a, B \to b$, $T' = \{a, b\}$
$V' = \{S, A, B\}$
$T' = \{a, b\}$
$P' = \{S \to AB, A \to a, B \to b\}$

And we can prove that this, this will generate the same grammar I mean so, yeah; so let us take an example.

So, suppose G is like this, we have the rules and the variables these are the variables A, B, D, E and the terminals are say A B C small c and we have a productions. So, what are the productions we have, let me write the productions S is going to A B, A is going to b sorry, A is going to a, B going to b and the E is going to small c. This is our G and S is the starting variable. So, now, we want to apply the ALGO 2 on this.

So, you want to apply the ALGO 2; that means, we want to construct a equivalence grammar V G prime which is V prime. So, this is V, this is T, this is P, this is S.

So, V prime, T prime, P prime, S is same. So, now, how to construct V prime? To construct V prime, we just follow the ALGO like initially we first put S in this. Then we look at all the rules which are coming from I mean if A is already in V prime, we look at all the rules, all the production A is going to alpha then the variables and the terminals alpha is a stream. Then the variables and terminals which are in alpha the part of the sentential form those we are going to add in variables, we are going to add in V prime and the terminals we are going to add in T prime. So, that is the way. So, let us just do that. So, now, S is the variable.

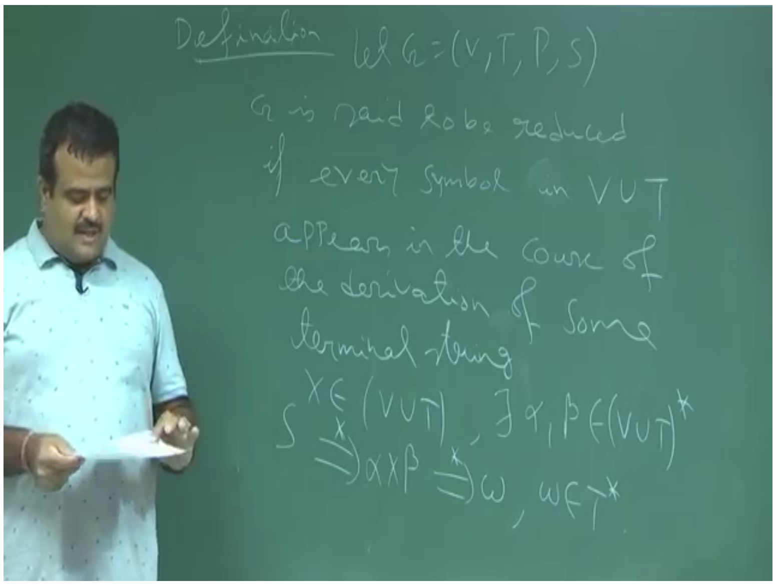
Now, we can have from S what are the rules. So, S is going to A B so; that means, so, this is alpha, alpha consists of A and B so; that means, we are going to add A and B in V prime. Now once, we add A and means V prime, we look at all the other rules which are coming from A and B. So, from A and B these are the rule, A is going to a once we get that and this B is going to b, and S is going to A B which is already there. So, if this will give us what this will give us this, this A is going to a. So, this A is a terminal.

So, we are going to add this terminal in T prime similarly this. So, T prime will be now initially it was an T A B. So, now, no further modification is possible. So, now, what is our V prime. So, V prime is yeah, so, V prime is just S, A, B and what is our T prime, T prime is A B. And then what are the rules? The P prime, P prime will be that the rules of subset of P where it is only involved this string, string coming from this. So,; that means, only the a S is going to A B and A is going to a, B is going to b. So, this will not come into the picture, because neither E or c small c belongs neither E belongs to B prime or small c belongs to T prime. So, that is why this will not come into the picture. So, this is our reduce, reduced grammar.

So, for this grammar we can for this grammar we can see that S is going to alpha X beta where alpha X beta are all coming from alpha beta are coming from the string which are V prime and T prime and X is also A. So, alpha beta X, X is a variables or a terminals. So, this X is going to this sentential form, ok. So, maybe we, we are not now we will use the ALGO 1 here to reach to that terminals only. So, this ALGO 2 is the reaching to a sentential form which involves the variables and the terminals now for language, you have to look for only the terminals. So, that will give us from ALGO 1. Now, we are going to combine this ALGO 1 and ALGO 2 to get a equivalence grammar, ok.

So, let us do that. So, this is ALGO 2, ALGO 2 is just reducing all the all the variables all the terminals which are not in the sentential form of G so; that means, which cannot be derived which is not in the part of the string, which is not in the part of the string which can be derived from S, so which is not in the parse tree of this. So, that is ALGO 2. So, now, when we combine these two, it will give us the reduced grammar. So, let us try that. So, let us try to combine these two; so that is basically our ALGO 3, ok.

(Refer Slide Time: 16:15)



Definition let $G = (V, T, P, S)$

G is said to be reduced if every symbol in $V \cup T$ appears in the course of the derivation of some terminal string

$X \in (V \cup T)$, $\exists\, \alpha, \beta \in (V \cup T)^*$

$S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$, $w \in T^*$.

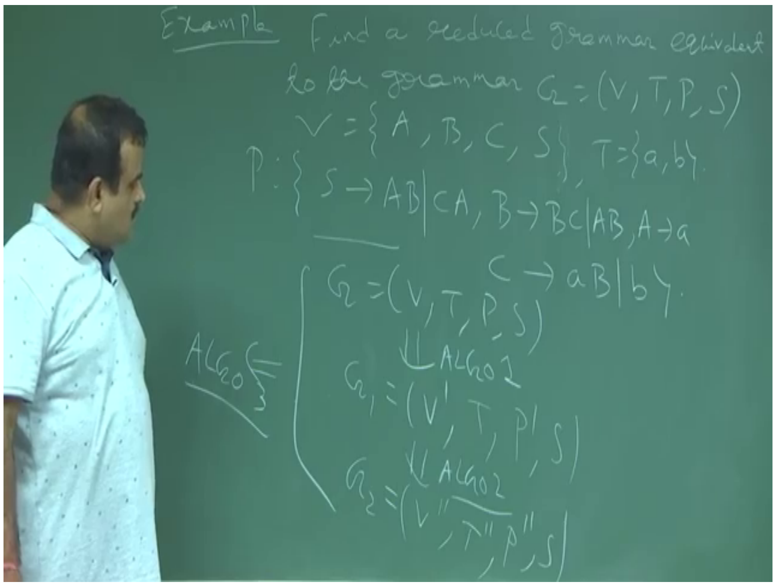
So, yeah, so, before that let me define this reduce and non, non reduced. So, suppose let G V a grammar G is said to be reduced, reduce. If, if every symbol, symbol in V P appear, appears in the course of derivations of the derivation of some terminal strings, some terminal strings.

So, that is for every; that means, for every X which is coming from which is either a variable or a terminal then there exists A alpha beta which is a string of variables and terminals such that S is I mean this can be derived alpha X beta and which can make clear W that W is a string. Then we say this is already reduced grammar or which cannot be further reduced.

So, that we are going to achieve, from a given grammar you want to achieve this type of grammar where these all the variables are terminal will be useful in the will be in the same sentential form of G and which will leads us to a string of the terminals. So, that is basically the, we are looking for this language, ok. So, that will basically come from which is called combination of this ALGO, ALGO 1 and ALGO 2; so that we are going to do now.

So, let us that is called ALGO 3 which is basically the combination of ALGO 1 and ALGO 2. So, given in a grammar first we apply the ALGO 1 which will only consider all the variables and the terminals which are in the sentential which are in which I belongs to the sentential form of this. And then once is no for ALGO 1 which are just go to the string of the terminals and the ALGO 2 will reduce the all the unnecessary the variables and the terminals.

(Refer Slide Time: 19:55)



Example Find a reduced grammar equivalent to the grammar $G = (V, T, P, S)$

$V = \{A, B, C, S\}$, $T = \{a, b\}$.

$P: \{S \to AB \mid CA, B \to BC \mid AB, A \to a$

$C \to aB \mid b\}$.

ALGO 3

$G = (V, T, P, S)$

$\Downarrow$ ALGO 1

$G_1 = (V', T, P', S)$

$\Downarrow$ ALGO 2

$G_2 = (V'', T'', P'', S)$

So, let us try that. So, that is that combination of ALGO 1 and ALGO 2 is called ALGO 3 which will give us the reduced grammar. So, find a reduced grammar which is equivalent to, to the following grammar G, grammar G whose production is grammar G which is V,T, P, S where V is the variable A B, C, S and the terminals are A B only and the productions are this. So, S is going to AB or C A B is going to B C or A B and A is going to a and C is going to c is going to a B or b. So, these are this is the, this is the; this is our production rules, so, set P. now this, so, we want to get a reduced grammar from this.

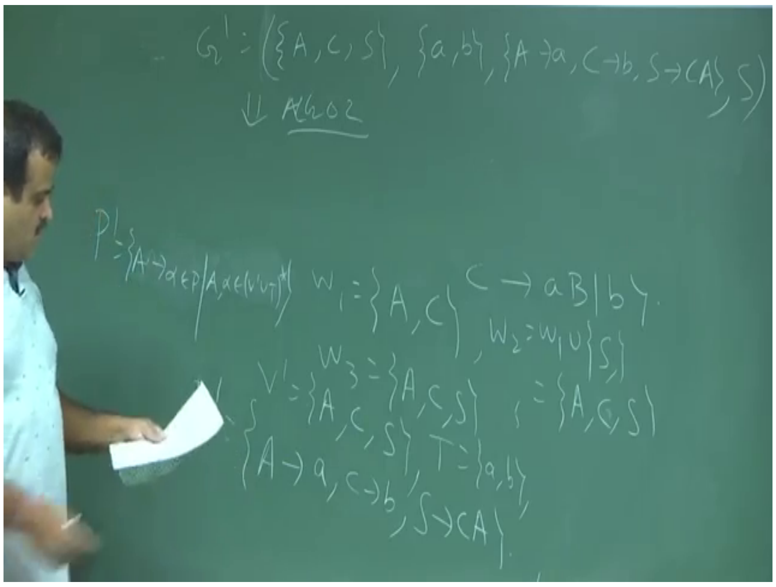
So, basically you have to apply the ALGO 3 which is basically ALGO 1 followed by ALGO 2. So, given G what we do? We just first apply ALGO 1. So, ALGO 1 will give us G prime or say G 1 which is basically V prime T is remain same. If you remember P prime and S and from here after that we will apply ALGO 2, ALGO 2 will change everything.

This is G 2 which is basically the V is a double prime T double prime P double prime S, ok.

So, this is the ALGO 2. So, if we apply this ALGO 2 so, this combination of this is called ALGO 3 which will give us the reduced grammar; that means, we, which cannot be further reduce like which cannot be further reduce in the sense of cannot eliminate the some of the we can do by null elimination, I mean epsilon string elimination that we will do, but this will the this is called reduced grammar.

So, yeah, so, just try to complete this. So, first we will apply ALGO 1 on this.

(Refer Slide Time: 23:35)



$G_2' = (\{A, C, S\}, \{a, b\}, \{A \to a, C \to b, S \to CA\}, S)$

$\Downarrow$ ALGO 2

$P' = \{A \to \alpha \in P \mid A, \alpha \in (V' \cup T)^*\}$

$W_1 = \{A, C\}$ $C \to aB \mid b\}$.

$W_2 = W_1 \cup \{S\}$ $= \{A, C, S\}$

$W_3 = \{A, C, S\}$

$V' = \{A, C, S\}$, $T = \{a, b\}$,

$\{A \to a, C \to b, S \to CA\}$.

So, if you do that what we will get. So, we will be getting the ALGO 1, if we apply. So, G 1, G 1 is nothing, but so, for G 1, you have to calculate say G 1 sorry, W 1. So, W 1 means we are just consider all the all the variables which are directly going to this terminals. So, this is basically A and C; so we put AC first there.

Then once we calculate W 2 which is basically W 1 in union of we consider all the variables which are going to a string combining of W 1 and the terminals. So, which are going so, yeah? So, S is going because S is going to C A which is already W 1. So, this is S anything else B is going to B C B is noted there A B is noted there A is C is going to no more.

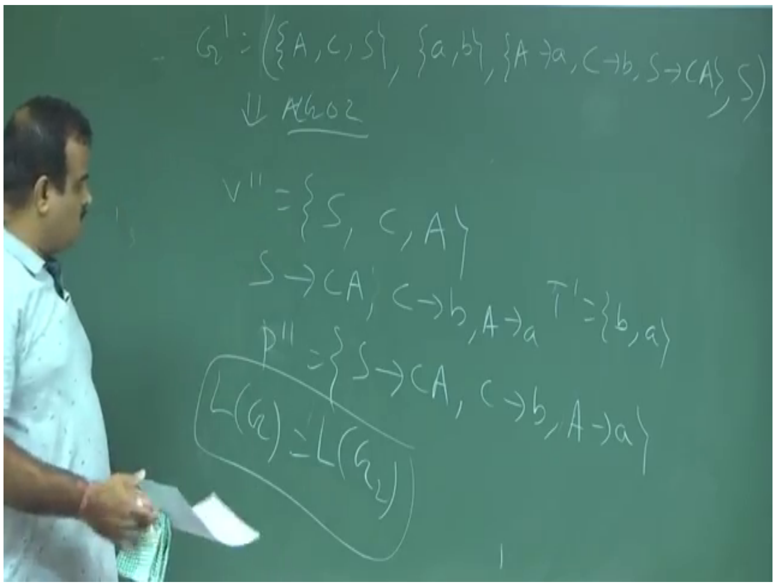
So, this is the, this is our W 2, W 2 is AC S this is capital C, ok. So, now, W 3, W 3 will be this W 2 along with W 4 which will be empty, you can easily check. So, this is this here is stop. So, this is our V prime. So, our V prime is this W 3 which is nothing, but.

So, V prime is A, C, S and T is remain same. We are not doing anything on the T and the production will change production will involve only this. So, production P prime will involve only this. So, if you do that we will get only so, we will get only these are the production like A is going to S is going to A B, but B is not there in this. So, S this rule will not come.

So, S A is going to a is going to A and C is going to b and S is going to C. So, this will be P prime, how to get P prime? P prime is set of all production of P such that, such that A and alpha both are coming from V prime P star. So, this way we construct this. So, we got this. Now once, we got these then you have to apply G 2 on this. So, how let us quickly do that. Let me remove this. So, now, we got a new grammar which is G 1. So, G 1 is nothing, but A C S terminals are same A B. And we have productions these are the productions A is going to a C is going to B.

And S is going to C A and we have P prime. So, this P prime is there. So, T and we have a start variable S, ok. Now we want to apply ALGO 2 on this, ALGO 2 means we just consider all the variables which are all the variables or terminals in the string which are in sentential form. So, further if you do that then we can get the reduced grammar.

(Refer Slide Time: 27:48)


G₁: ({A, C, S}, {a, b}, {A → a, C → b, S → CA}, S)
⇓ ALGO2
V'' = {S, C, A}
S → CA, C → b, A → a
T' = {b, a}
P'' = {S → CA, C → b, A → a}
L(G) = L(G₂)

So, if we apply ALGO 2. So, V prime, we first add S. Once, we add S then we see S is going to were C A. So, we had both the CA. Now S is going to C A and C is going to b.

So, this will add now B in the terminal strings and A is going to a. So, we have to add A also in the terminal strings, ok. And now what is P prime?

So, this is our, what is our V prime? This is T and what is our P, P is nothing, but the same way S is going to C A and C is going to B and A is going to a. So, no, no, no further changes is there, but there will be some example where it will further reduce. So, this will give us the same grammar.

So, L of G is basically L of G 2. So, that is the way we are going to; that means, they are coming in same sentential form and there were reducing the variables which are redundant I mean which are not used and the terminals also we are reducing. So, we will talk about this on more. So, this is the ALGO 3 which is the combination of ALGO 1 and ALGO 2. Now next, next class will discuss the null elimination we will remove the all the epsilon move there epsilon transmission there. So, that will be ALGO 3 and then we will, will, will do ALGO 4 of no ALGO 4 and ALGO 5.

Thank you.