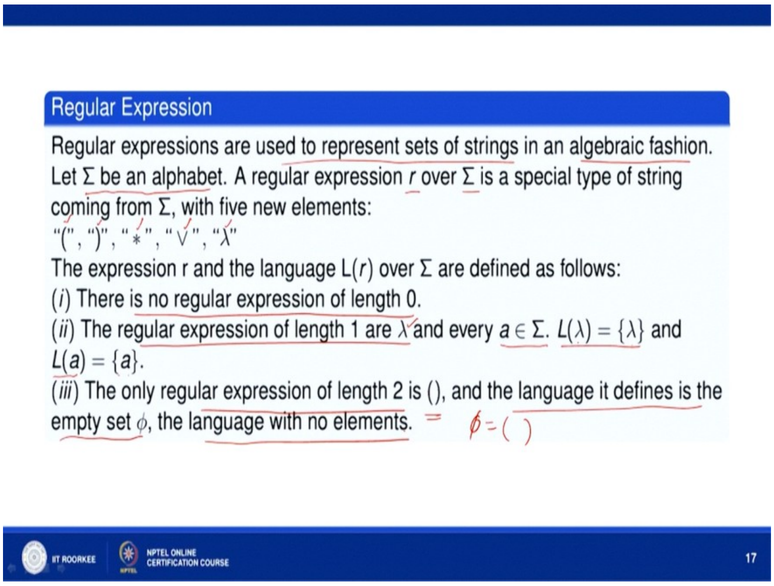
Higher Engineering Mathematics
Prof. P.N. Agrawal
Department of Mathematics,
Indian Institute of Technology Roorkee
Language and Grammers - II
Mod02_Lec08

(Refer Slide Time: 0:26)

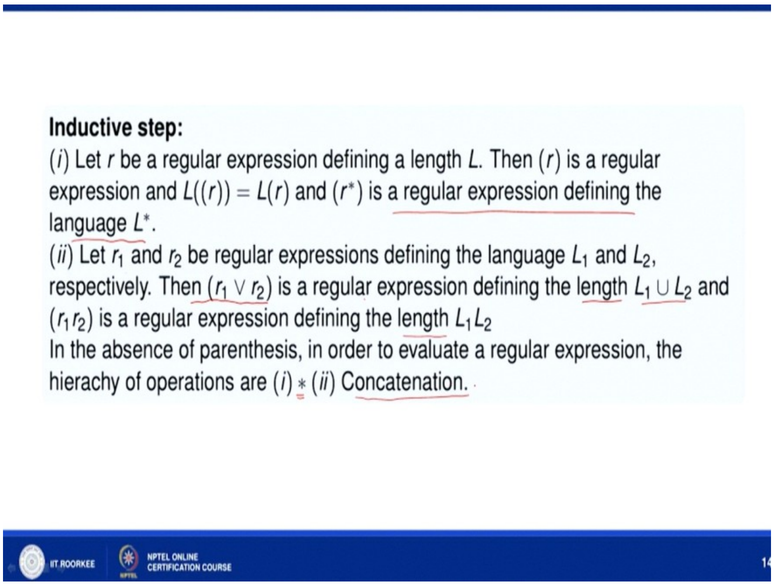


Hello friends welcome to my lecture on Languages and Grammars. This second lecture on this topic. Regular expression let us define, so a regular expression is used to represent sets of strings in an algebraic fashion, let ∑ be an alphabet okay, a regular expression r over ∑ is a special type of string coming from ∑, with five new elements, so the five new elements are this left brackets, this right bracket, ¿ then or, the symbol denotes or and then λ okay.

The expression r and the language L(r) okay, over ∑ are defined as follows, there is no regular expression of length 0 that is first thing. Okay, these are the rules, there is no regular expression of length 0, the regular expression of length 1 okay and 1 are, let us consider the regular expression of length 1, then there are regular expression of length 1 we have as λ and every a ∈∑.

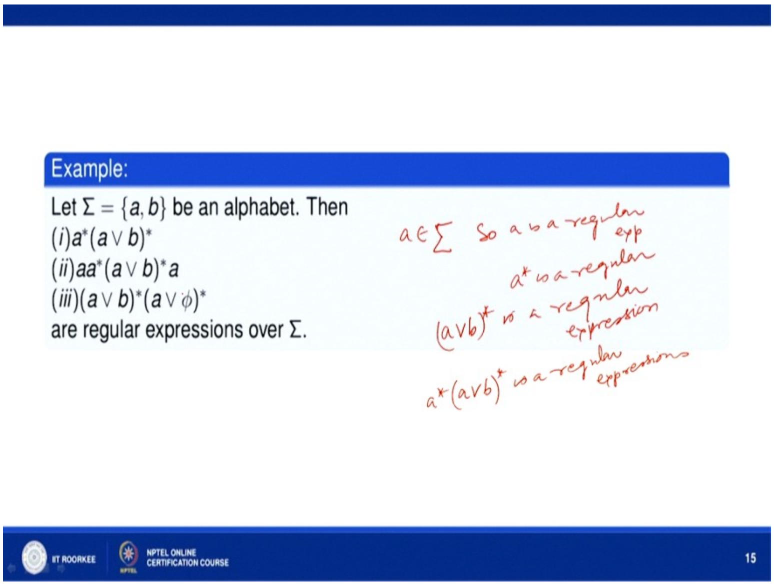
So whatever a ∈∑ you take they are all regular expressions of length 1 apart from λ and L (λ) okay, language over a λ in the set containing λ by language over A is the = set containing A okay, so that is how we get the language L(r), when r= A. The only regular expression of length 2 is this ( ) because you can see, we have ( ), so they are counted 1 each, 1 +1= 2, so regular expression of length 2 is this is ( ) nothing else, so the language it defines is the empty set =∅ okay, so ∅ is denoted, I denotes this regular expression of length 2. Okay, the language with no elements.

(Refer Slide Time: 2:30)



Now, let r be a regular expression defining a length L. Then r is a regular expression and L(r) = L(r), $r^{¿}$is regular expression defining the language $L^{¿}$, let $r_1 \wedge r_2$be regular expressions defining the language $L_1 \wedge L_2$ respectively. Then $r_1 \vee r_2$ is a regular expression defining the length, defining the language $L_1 \cup L_2$ and r_1, r_2 is a regular expression defining the language L_1, L_2, in the absence of parenthesis, in order to evaluate a regular expression, the hierarchy of operations are, first we take up this ¿ then we take up the concatenation and third we take up R okay, third we take up r.

(Refer Slide Time: 3:24)

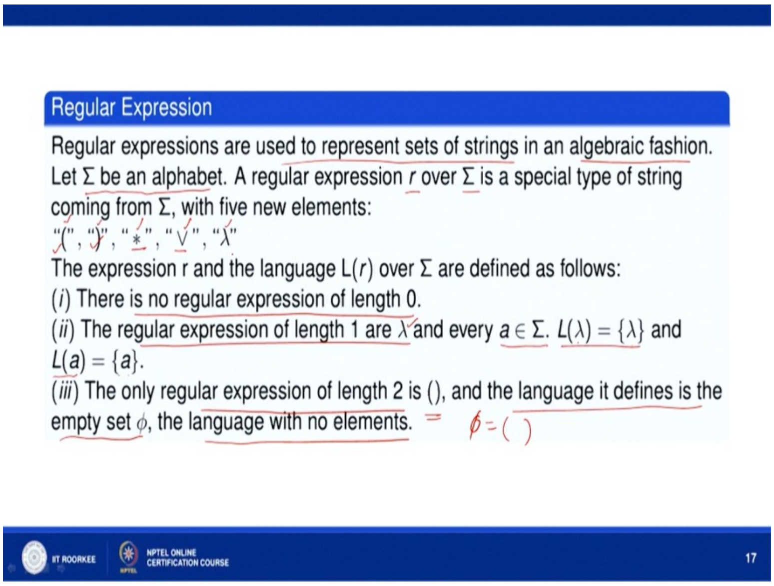


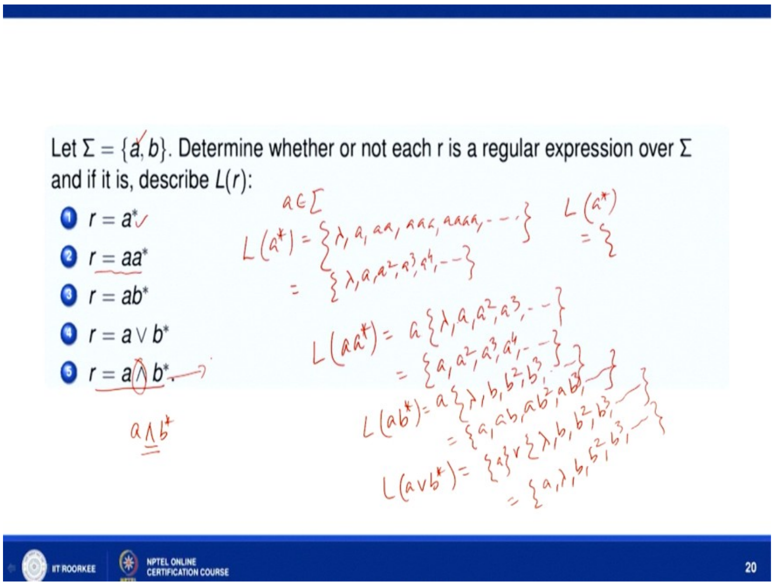
So let a,b and ∑ = a,b be an alphabet, then a ∈ ∑ okay, a ∈ ∑, so a is a regular expression, a is a regular expression and when a is a regular expression $a^{¿}$ is a regular expression. Okay, now a,b ∈ ∑, so a and b are regular expressions, so a ∨ b is a regular expression. Now $a \vee b$ is a regular expression, so $(a \vee b ¿¿^{¿}$ is also a regular expression and this is a regular expression. Okay, where both a and *a are same*.

So, then $a^{¿}$, $(a \vee b ¿¿^{¿}$ this is also a regular expression by the properties of regular expressions. Okay, now a ∈ ∑, so A is a regular expression and therefore $a^{¿}$ is a regular expression and so $a^{¿}$ is a regular expression and here we have $a \vee b$, so a ∧ b are regular expressions, so $a \vee b$ are regular expression and therefore $(a \vee b ¿¿^{¿}$ is a regular expression. Okay, so $a^{¿}$, $(a \vee b ¿¿^{¿}$, a is a regular expression.

Now here (a ∧ b) are regular expressions, so $(a \vee b ¿$ are regular expression and therefore $(a \vee b ¿¿^{¿}$ is a regular expression, now here we have A ∨ ∅ okay A ∧ ∅ are regular expressions, so a ∨ ∅ is a regular expression and therefore $(a \vee \varnothing)^{¿}$ is a regular expression, so $(a \vee b ¿¿^{¿}$ and A ∅, $a \vee b$ ∅ ¿ okay, these are regular expression, so they are regular expressions over ∑.

(Refer Slide Time: 5:19)





Let us say ∑ = {a, b}. Determine whether or not each r is a regular expression over ∑ and if it is described L(r), so r(a) is there in element in a ∈ ∑, so $a^{¿}$, a is a regular expression and therefore $a^{¿}$ is regular expression and L ($a^{¿}$¿ = λ a, aa,aaa,aaaa and so on. Okay, so that is nothing but λ a, a^2, a^3, a^4 and so on, because L ($a^{¿}$¿ is nothing but the ∪L ($a^{¿}$¿ = λ a, aa,aaa, so this is λ a, a^2, a^3, a^4.

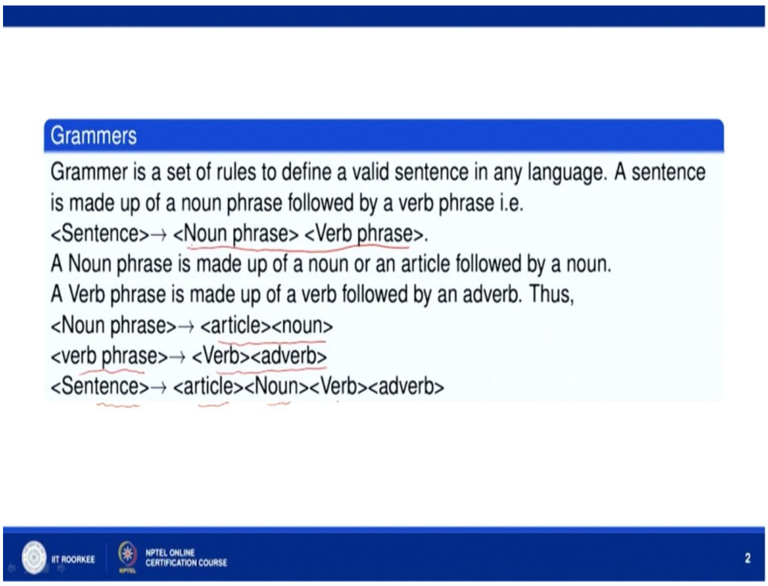
Now, when we consider second example, case r =a $a^{¿}$, since a is a regular expression, $a^{¿}$ is regular expression, a $a^{¿}$ is regular expression and L (a$a^{¿}$¿ =a and then for $a^{¿}$ we have λ a, a^2, a^3 and so on, so we will have, now let us take the concatenation, so a λ, that is a, then a, so then a^2, aaa^2 so a^3, then aa^4, a^3, so $a^4$and so on okay.

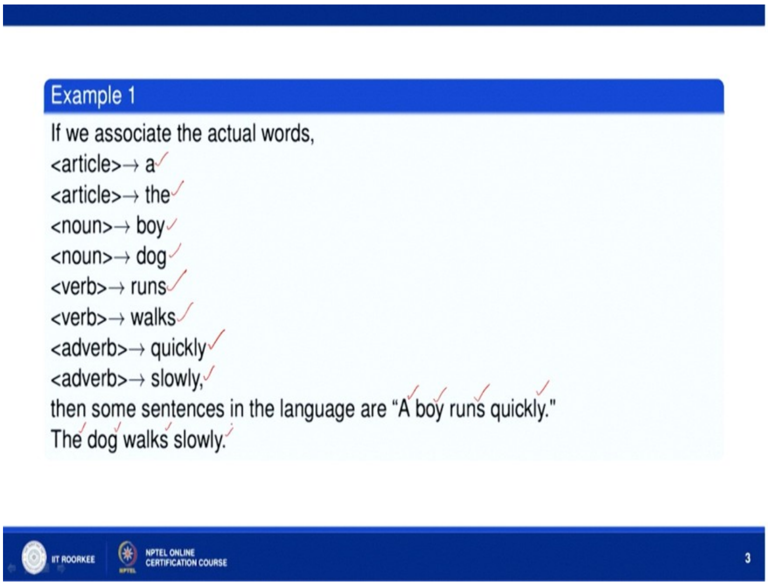
Now, r =ab^2, so L (r) Okay, L (a$b^{¿}$) okay, so a and then $b^{¿}$, $b^{¿}$is λ b, b^2, b^3 and so on, so L(a$b^{¿}$ will be a λ that is a,ab then ab^2, ab^3 and so on, so that is L (a$b^{¿}$¿, then a is a regular expression, b is a regular expression, b* is a regular expression, so (a∨b¿$¿^{¿}$ is a regular expression. Okay, now L(a∨b *), so this will = a∨b okay, so b *, so b * means λ b, b^2, b^3 and so on. Okay, so this is a λ, b, b^2, b^3 and so on, that is L (a∨b*).

Now, in the case fifth, A and then this notation, this notation okay, this notation is not define in the case of regular expressions, we have only four, five special type of strings, so this one, this one, this *, then or and λ, so this is not there. Okay, this symbol is not there, so this is not a regular expression, these are not a regular expression.

Let us define what do mean by a grammer ? A grammer is not set of rules to define a valid sentence in any language, a sentence is made up of a noun phrase, followed by a verb phrase that is sentence is made up of noun phrase and verb phrase, a noun phrase is made up of a noun or an article followed by a noun, a verb phrase is made up of a verbal followed by an adverb, thus a noun phrase is made up of article and noun, verb phrase is made up of verb and adverb.

(Refer Slide Time: 9:26)

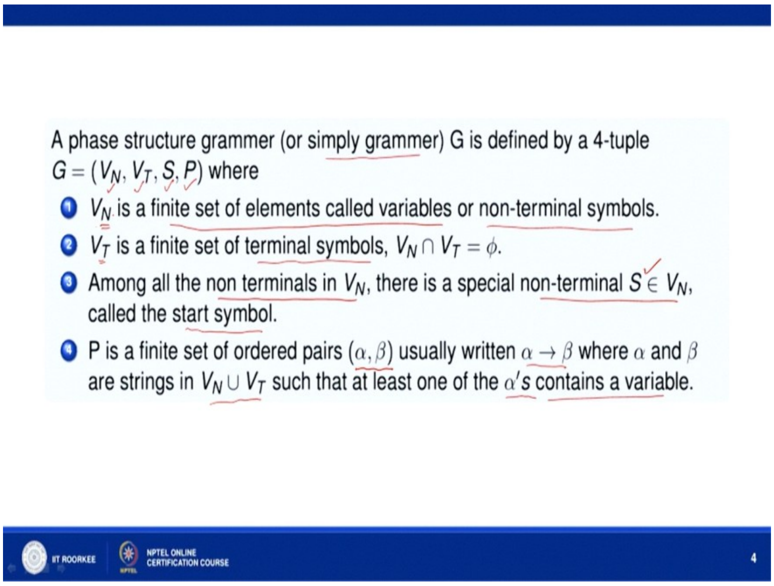




Thus we can see that a sentence is made up of an article, noun, verb, adverb. If we associate the actual words, say with the article we associate A, with the article we can also associate the noun, we can associate with boy and then with this noun we associate dog and the verb we associate runs or walks and with adverb we associate quickly or slowly, then some sentences in the language could be, A boy runs quickly or The dog walks slowly.

So, here in this sentence A boy runs quickly, A is the article, boy is the noun, runs is the verb and quickly is the adverb, so this sentence is made up of an article noun, verb, adverb. Similarly, the case of the second sentence The dog walks slowly, The is an article, dog is a noun, walks is a verb and slowly is the adverb.

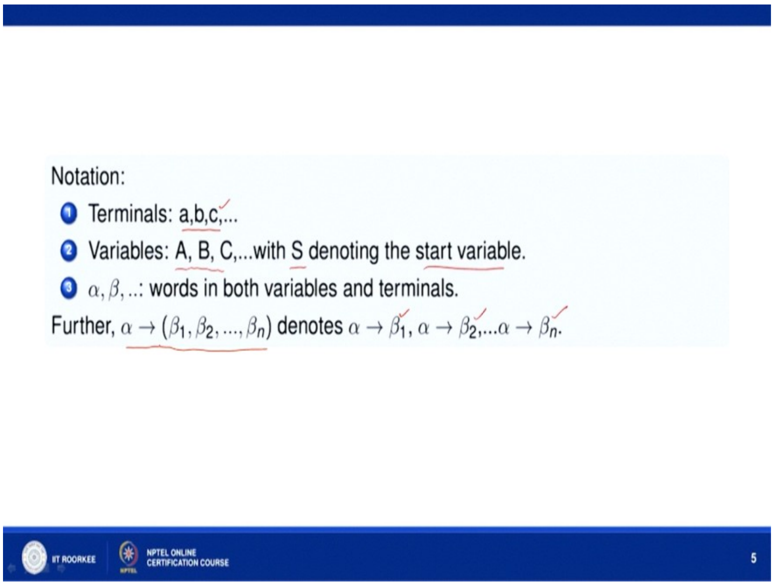
(Refer Slide Time: 10:33)



Now, let us discuss a phase structure grammer, okay phase structure grammer or simply grammer okay, G, G we can denote it by G, it is defined by a 4-tuple G, consisting of V_N, V_T, S, P where V_N stands for a finite set of elements called variables or non-terminal symbols, V_T , this N means non-terminal symbols and this T denotes terminal symbols, so V_T is a finite set of terminal symbols and clearly the intersection of V_N and V_T should be pie, because V_N consist of non-terminal symbols and V_T consist of terminal symbols.

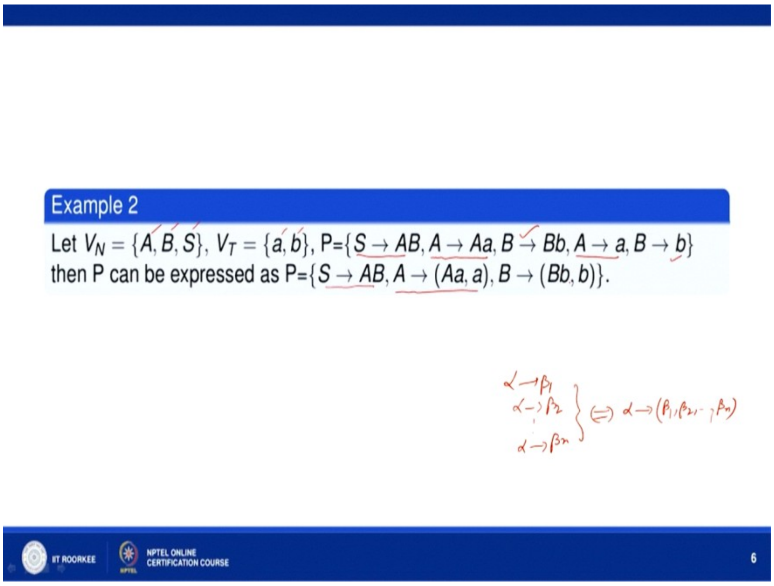
Among all the non-terminals symbols in V_N, in V_N okay, there is a special non-terminal, we denote it by S okay, it is called the *t symbol, now P is a finite set of ordered pairs $\propto$, β, usually written as $\propto \rightarrow \beta$, where $\propto$ and β are strings in V_N and V_T, $V_N \cup V_T$ such that at least one of the $\propto$,s contains a variable, variable means an element of V_N.

(Refer Slide Time: 11:52)



Now, terminals, terminals are denoted by lowercase letters of the English alphabet a, b, c and variables are denoted by the capital letters of the English alphabet A, B, C, S denotes the, as we have already said S denotes the *t variable, $\propto$, β denotes the words in both variables in terminals. Now, there is a notation which we shall be using $\propto \rightarrow \beta_1$ 1, β_2,.... β_n, it means that it consist of, the notation $\propto \longrightarrow \beta_1$ 1, $\propto \longrightarrow \beta_2$, $\propto \longrightarrow \beta_n$, so all these $\propto \longrightarrow \beta_1$ 1, $\propto \longrightarrow \beta_2$, $\propto \longrightarrow \beta_n$ can be represented in terms of a single implication, $\propto \longrightarrow \beta_1$, β_2, β_n, this is just for the sake of convenience.

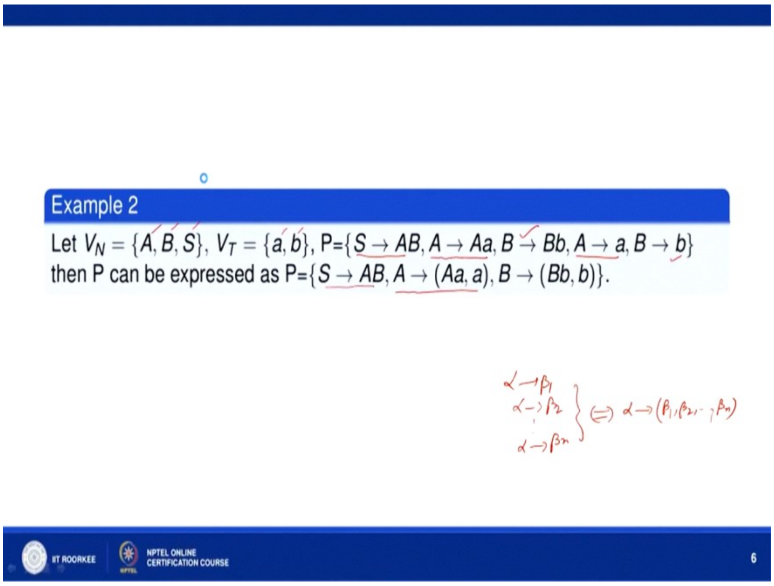
(Refer Slide Time: 12:55)

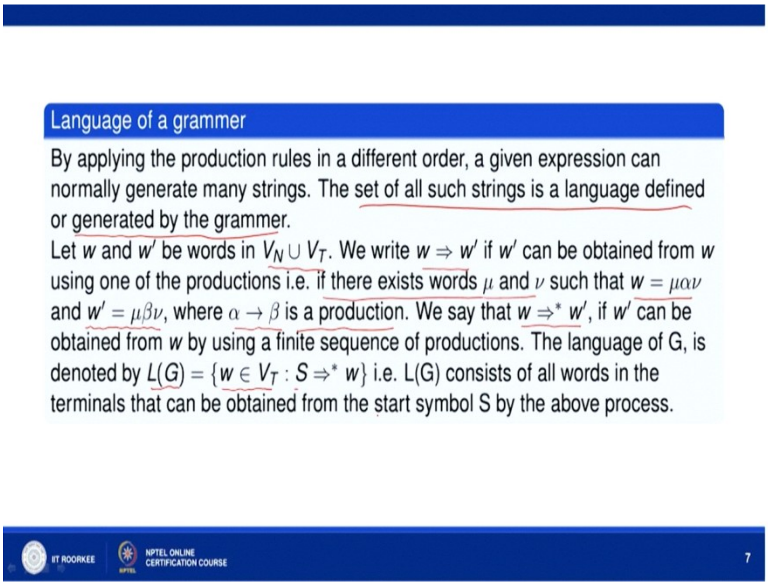


Now, let us say V_N which is the set of non-terminal symbols okay, consist of {A, B, S}, S is the symbol and V_T which is the set of terminal symbols consist of {a, b} and P is the set of productions okay, which consist of S → AB, A → Aa, B → Bb, A → a, B → b, then P can be expressed also as P = {S → AB, so this we can write as such, S → AB}.

Now here we can see that A → Aa, also A → a, so by our previous notation where you said that if $\propto \rightarrow \beta_1$, $\propto \rightarrow \beta_2$ and so on, $\propto \rightarrow \beta_n$ then they can be represented by a, just one notation $\propto \rightarrow \beta_1, \beta_2, \beta_n$. Okay, so using this A → Aa, A → a can be represented as A → Aa, A and similarly B → Bb okay, B → Bb and B → b can be written as B → Bb, B, so this is for the sake of convenience that we write P in this manner.

(Refer Slide Time: 14:21)



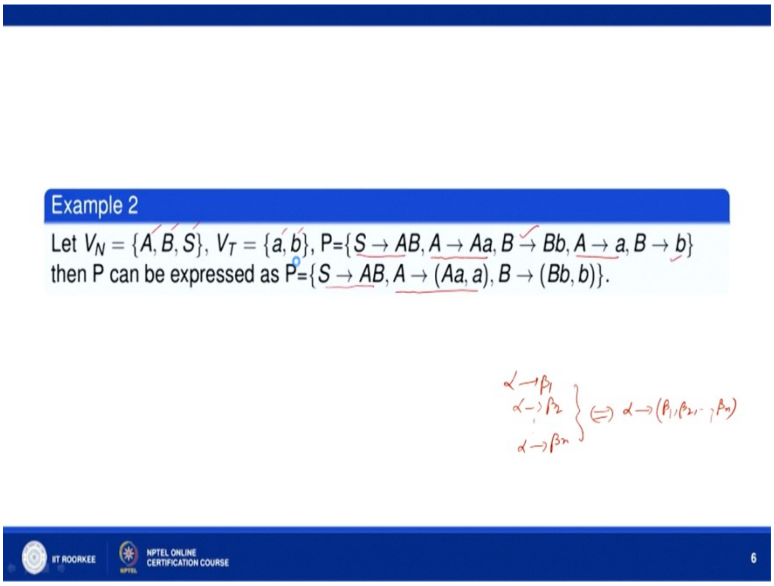


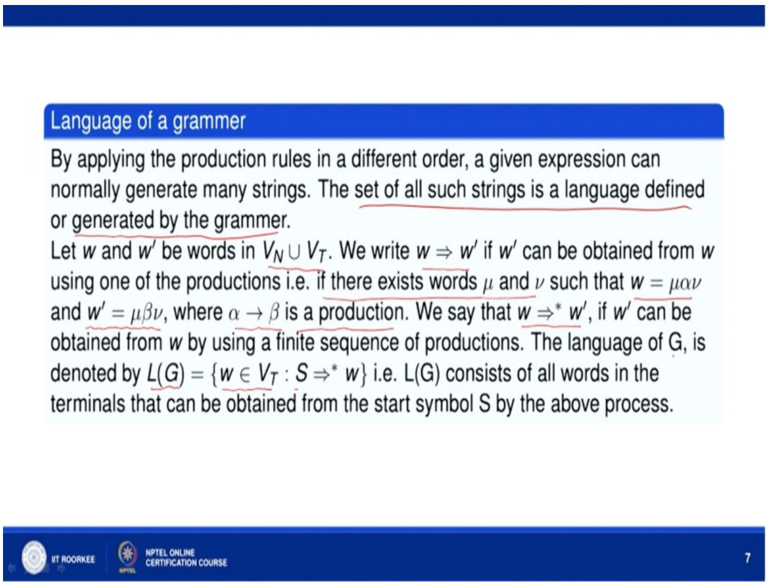
Now, let us consider the language of a grammer, by applying the production rules okay, the production rules are the elements of P okay, the elements here S → AB, A → Aa, A, B → Bb, B these are production rules, so by applying this production rules in a different order. Okay, a given expression can normally generate many strings. The set of all such strings is a language defined or generated by grammer.

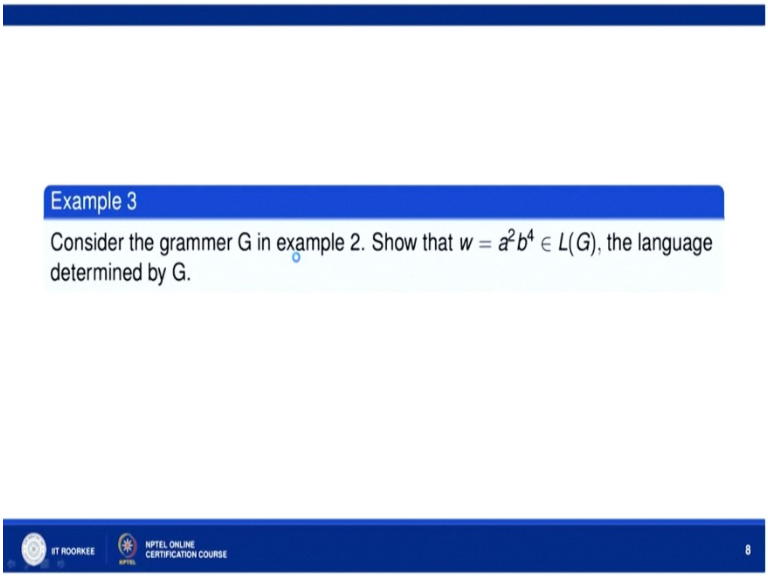
Now if W and W' are any two words in $V_N \cup V_T$, than we see that V →, W ⇒ W', if W' can be obtained from W using one of the production that is if we can find μ and ϑ, such that W = $\mu \propto \vartheta$ and W' is = $\mu\ \beta\ \vartheta$, where, $\propto\ \to \beta$ is a production, so we shall say that W → W' provided we can find μ and ϑ, such that W is = $\mu \propto \vartheta$ and $W'1 = \mu\ \beta\ \vartheta$ and $\propto\ \to \beta$ is a production.

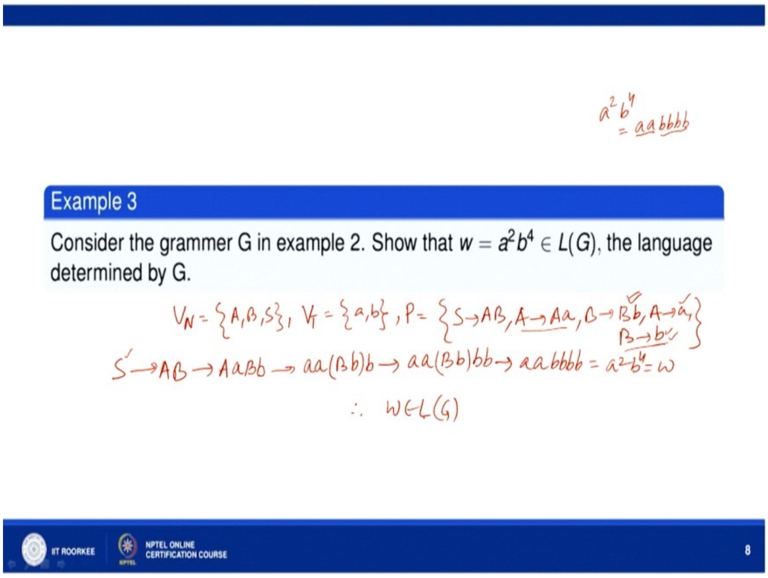
Now, we also say that W $\rightarrow$ * $W^{'}$, if $W^{'}$ can be obtained from W using a finite sequence of productions. So, using a finite sequence of production if a W $\rightarrow$ $W^{'}$, that we write it by the notation W $\rightarrow$ * $W^{'}$. The language of G is denoted by L(G) okay and it is the set of all W $\in V_T$, V_T as we recall, V_T is the set of terminal symbols, so W $\in$ V_T such that S $\rightarrow$ * W, S is the * symbol, that is L(G) consist of all words in the terminals that can be obtained from the * symbol S by the above process.

(Refer Slide Time: 16:20)









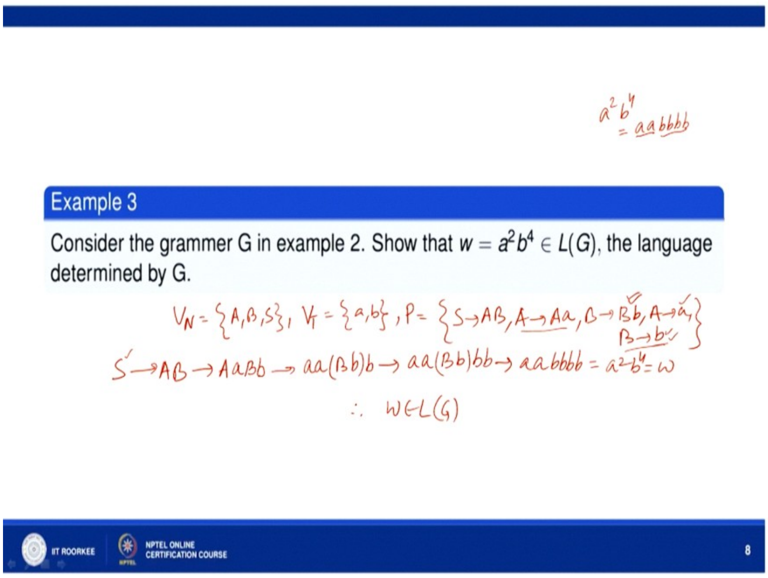
Now, let us see the following example, consider the grammer G in example 2, show that W = $a^2b^4 \in$ L(G), the language determined by G, let us see what is the example 2, this example 2, so V_N ={A, B, S}, B and S, V_T = {a,b} and the production rules are P = S → A, B, A → Aa, B → Bb, A → a, B →b, now we have to say that there double = a^2b^4 ∈ L(G) okay, so we start with S.

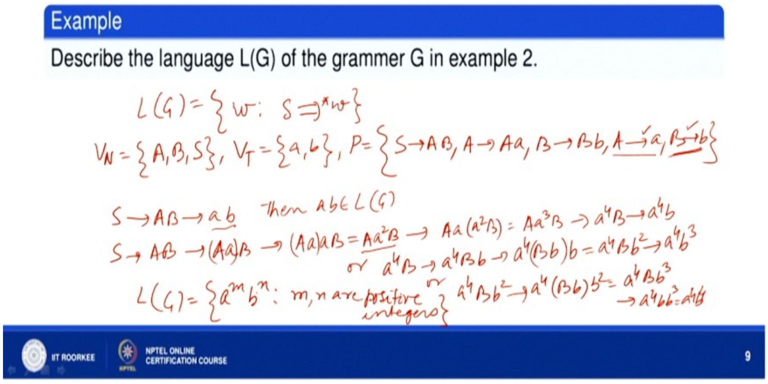
Now, when we will say that W ∈ L(G), let us go to the definition the language of G is denoted by L(G), and L(G), consist of all W ∈V_T such that S as → * W, S → * W means we can reach W by using a finite sequence of productions *ting with S, so let us see, we *ts with S. Okay, now S →, you can see here. S → AB, so be write AB here. Okay, now what we want? We want ultimately W, W means a^2 b^4, a^2 b^4 means AA, BBBB okay, so that is AA means A^2, B^4 means B^4 okay, now let us how we will reach there to AA, BBBB okay.

If you simply write A → A, B → B, then AB will → AB and we will not reach here, so we cannot directly use A → A, B → B, what we should do? We should use A → AA okay, A → AA, then we will write A → AA and B → BB, B → BB, now what we can do, because we just want 2 As here, so we can use the, this production rules A → A, then AA will go to AA and BB, B can now be replaced with again BBBB because we want 4 Bs okay, so B can replace with Bb, so BB B okay, now here, now if you replace this B using this production rule B → b, we will have only 3 Bs okay, so we again use this production rule B → Bb and we will have Aa, Bb bb.

Now we can use this production rule B → b and we shall have AaBbbb which is = a^2b^4 okay, so this is W, so you can see we have reach W using a finite sequence of production rules starting with the start symbol S and therefore W ∈ L(G), okay, so W ∈ L(G),.

(Refer Slide Time: 20:21)





Now, describe the language L(G), of the grammer G in example 2. Okay, let us see now what kind of elements could be there in L(G). L(G) = set of all W such that S → * W. Okay, so let us see what we have here V_N = again we will have to right the values of V_N, V_T and P okay, so V_N = {A, B, S}, V_T = {a,b} and production rules are P → P, production rule P is set of productions, P = S to A, B, then we have A → Aa, we have B → Bb and then we have A → a, B → b okay.

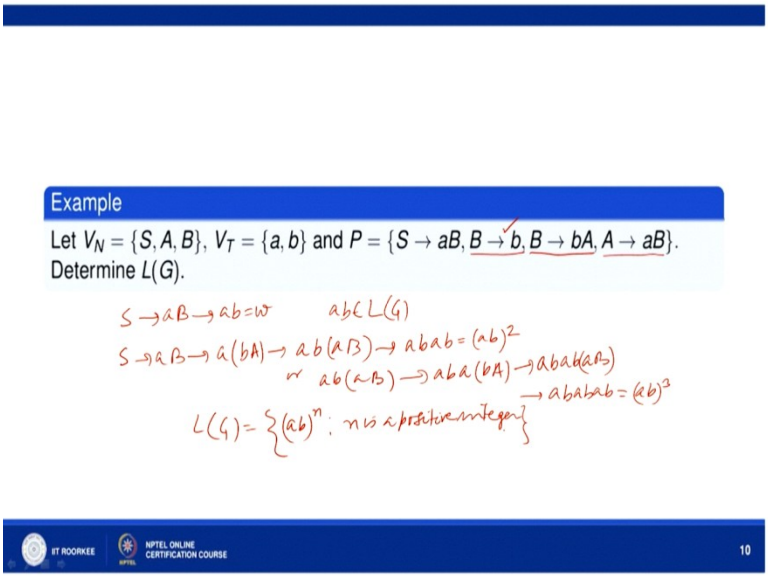
So, we start with the symbol, start symbol S, now S → AB okay, one thing could be that we straightaway replace A this A with A and this B with B, so we will have AB, if I replace A with A, B with B be using this two production rules, I will get AB, so AB is an element of L(G) okay, else what we can do S → AB, now A → Aa, A → AA B okay, again A → Aa we can write okay, so we will have A A^2 A, A A^2 B okay, again we can say this → AA A^2 B, so we will have A A^3 B okay.

Now, if I replace A with a, I will have A to the power 4 B okay, then B → BB, I can straight away write, use this production rule B → B, then I will have A^4 B or I can also do A^4 B → A^4 BB and then I can see, if I replace B with b. I will have A^4 B square okay or what I can do A^4 BB B okay, which will give me A^4 B B^2, if I replace B by using this production rule B, I will get $A^4 B^3$, I can also continue further and from here $A^4 BB^2$ which will take me to $A^4 BBB^2$ and I will get $A^4 BB^3$.

Now, if I use B → B I will get $A^4 BB^3$ that will give you $A^4 B^4$okay, so what we will get? We can get AB, we can also get like here, we had AA^2B okay, I could have use A here, I could have use the production rule A → A, then this would have given A^3B, so this L(G) $a^m b^n$, where m, n are positive integers, so all such elements are there.

Okay, minimum value of m and n could be one each okay, so we will have this, we have seen AB ∈ LB, so m, n both of them are = 1 each, then we will have L(G), AB ∈ L(G), other will be can also have A^2 B or A, B^2 or we can have A^3 B, $A^3 B^2$ like that, so $a^m b^n$are elements of L(G), where m and n are positive integers, so this language of the grammer G is given by this L(G).

(Refer Slide Time: 25:27)

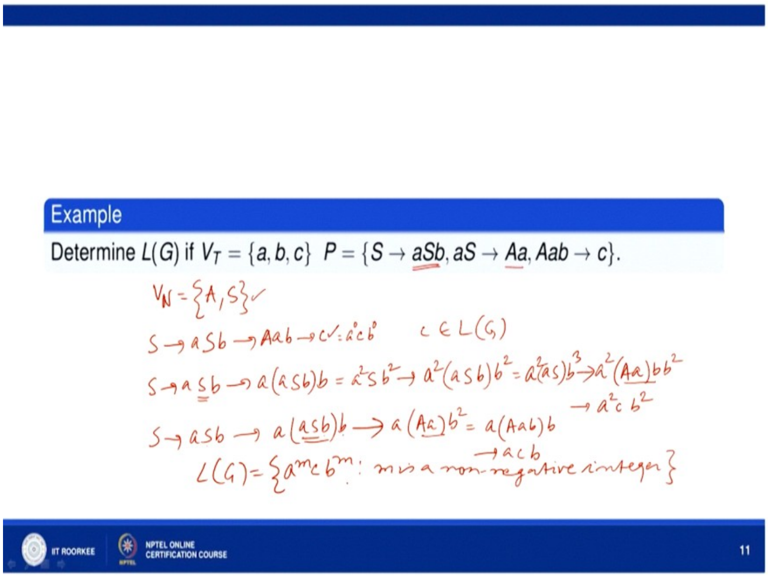


Now, let see another example, let V_N = {S, A, B}, V_T = {a, b}, P is the set of productions, A → aB, B → b, B → bA, A → aB, then let us see what is L(G) okay, so S → AB okay, now if I replace B with B directly okay, I will have AB, so AB ∈ W, AB ∈ L(G), okay, because AB is obtained from S by a finite sequence of production rules.

Now, another thing could be S → AB and then B → BA, we can use this, B → BA okay, now if I replace, now A can be replaced with this AB by using this production rule, so this → AB AB okay, now if I use this production rule B → B, then what I will get? abab okay, right, so by taking is B → B, I will end up with abab, if I do not do that. Okay, so from here, I can also do abaB, if I do not use this production rule, I use this one, then what I will get? ABA, B → BA okay.

Now, A → AB, so let us use that, so AB, AB A → AB okay, now B →, you can replace B →, you can replace B with this B, so then that is will give you AB AB AB, if use this one, if you do not use this one, you use B → BA, then here in place of B which will have BA, A will be replace with AB and then B can be replace with B, so we will have, this is AB^3 and this is AB^2okay and if you use instead of B → b, you use B → bA and then A → aB and then can B → b, we shall have ¿, so here L(G) =¿, n is a positive integer, so this is the language determined by G in this case.

(Refer Slide Time: 28:27)



Now, let us look at this determine L(G) if V_T = {a,b,c}, P = so here V_N, here V_N is A, S okay, the non-terminal symbols are A and S, S is the start symbol, A is another non-terminal symbol and V_T={a,b,c} and P is the set of productions like this, so S goes to here, you can see S → aSb, now aS → Aa okay, aS → Aa okay, so we have Aab, now what we can do? So one thing could be that Aab directly, yes Aab now we have to go to C, so C is an element of L(G), now here we can also do one thing is instead of replacing aS with Aa, I could have done S → aSb and then again I use production rule for S, S → aSb okay.

So, then what I will get A square SB square, A square SB square, now if I, now I can use S → again, S → ASB, so then this will be A cube or I can write it as, because I need to have one AS, so that yes, okay, so AS, I can now write A^2ASB^3. Okay, aS now → Aa okay, so a^2 AA, Bb^2 okay, now Aab → C, so this → A^2 C B^2okay, so you can see here we have on left side of C we have A^2, on right side of C we have B^2, so powers of A and B are same. Okay and here also if you can, if you use this ASB, ASB you replace AS by AA, we got this C, so here power of A is 0. Okay, this is nothing but A to the power 0 C B to the power 0.

We can also get ACB, how will get ACB? Let us see, S → ASB okay, if I replace S by ASB, I will have the powers of okay, let me see ASB okay and then what I can do AS I replace with AA okay, so we will have A AA B^2 but then AAB, AAB will be going to C okay, this is nothing but A AAB B okay, so this → ACB okay, so S → ASB okay, S now can be replaced with ASB, so we get A ASBB okay, now we can use this AS → AA okay, AS → AA means this →, I should write, this → A, ASB → yes, AS → AA okay.

So AAB^2, now one V, we have to take from here, so A AAB B okay, AAB → C, so we get ACB, so this, we can see that by using this production rules we will get on either side of C, the same power of A and B that is here A^2, here B^2, so we will have L(G) = $a^m c b^m$, where m is a nonnegative integer, so this is the problem where we have taken V_N to be {A,S}, V_T to be {a,b,c} and the set of production rules P and we found L(G), L(G) is the set of all elements $a^m c b^m$, where m is a nonnegative integer, so like this, we can find the language determined by the grammer G. With this, I would like to end my lecture. Thank you very much for your attention.