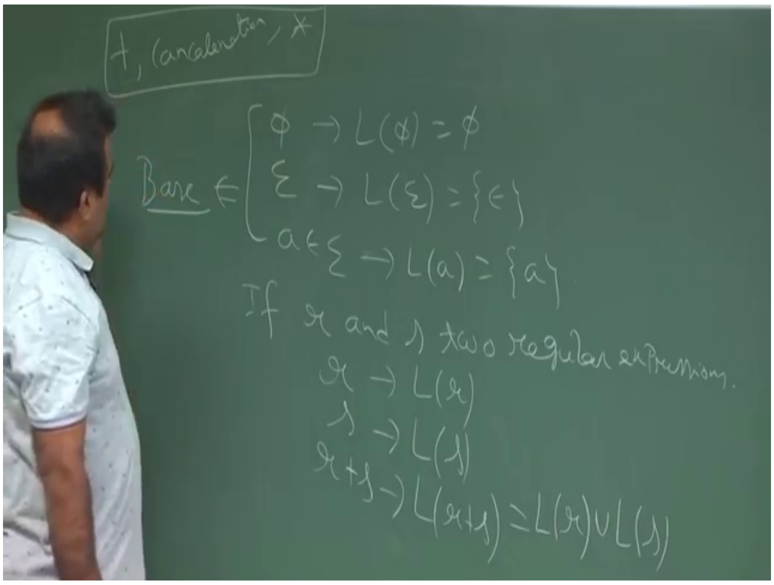
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 17
Regular expression (Contd.)

Ok, so we are talking about regular expression. The last class we have defined the regular expression.

(Refer Slide Time: 00:19)

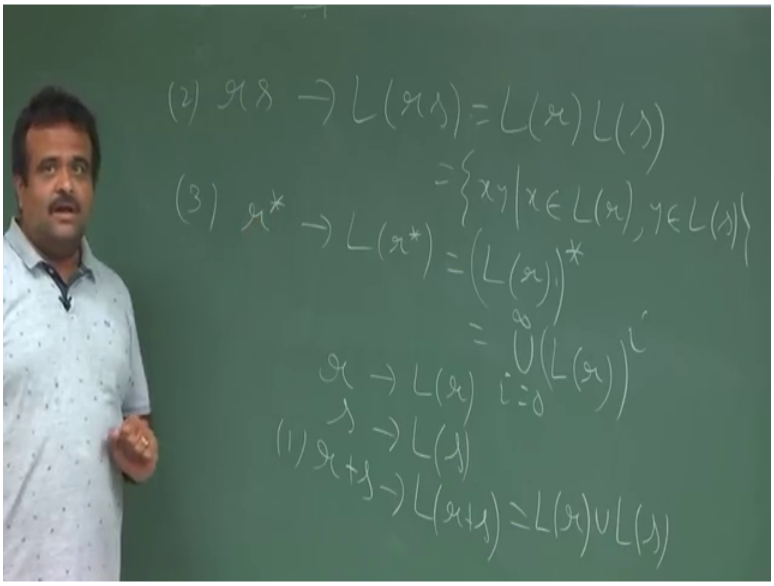


So, just to recap. So, phi is a regular expression which is denoting the language the empty set. Epsilon is a regular expression which is denoting the language the singleton set epsilon. And a which is coming from the alphabet, a is an alphabet is a regular expression for every alphabet is a regular expression which is denoting the language the singleton set a ok. So, this is the way, we define the regular expression for this notations.

And then we are recursively going to define the regular expression like if this is we called as Base. Then we recursively define the regular expression like if r and s two regular expressions like r and s could be any one of this. So far we have only base case, then r plus s will denote the regular expression by three operators plus, concatenation and star. So, by these three operations concatenation and star will define the regular expression ok.

If r is a regular expression, if r is denoting the language L of r, and s is the regular expression which is denoting the language L of s, now r plus s is a regular expression. This is plus. We are using three notations three operator. And the language of these regular expression r plus s is nothing but language of r union of language of s. This we defined in the last class. And then another two is r concatenate s, r s is you see also a regular expression.

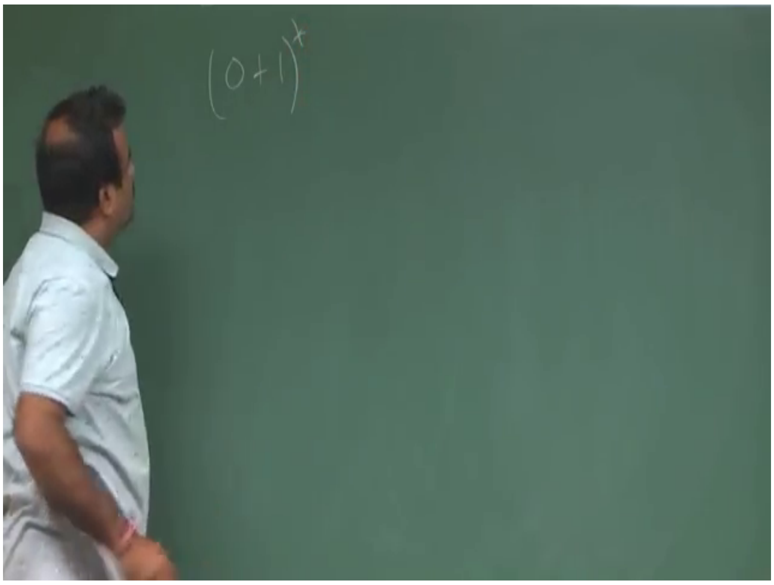
(Refer Slide Time: 02:51)



Now, what is the language of r s? It is language of r concatenate with the language of s. We know the concatenate operation this is a set of all x y such that x string is coming from this, and y string is coming from this ok. So, this is 1, recursive definition 2 and then 3 is r star, r star is also a regular expression. And the language corresponding to this is like this ok. And this is the union of r i, i is 0 to infinity. So, this is the way we have seen we can define the regular expressions.

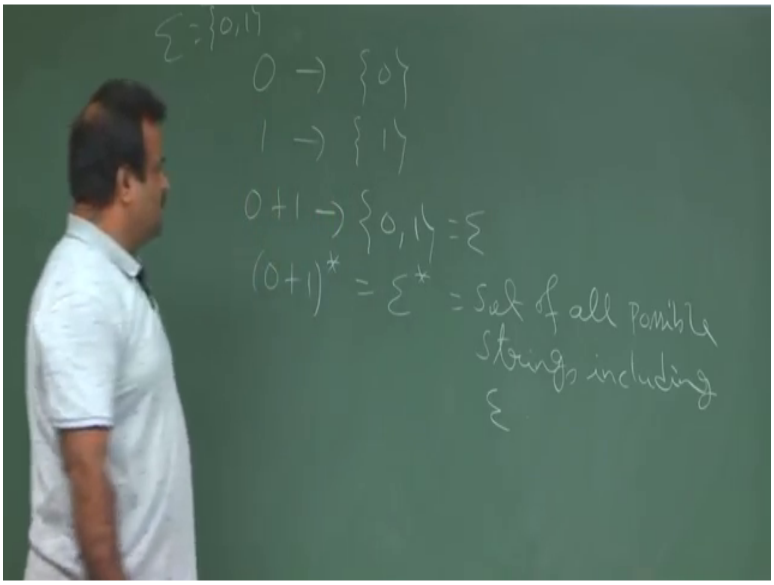
Now, today we will mainly talk about some examples and some properties of this operator. This plus, concatenation and star I mean the algebra of this operators. So, before that let us just quickly talk about some examples. So, regular expression is corresponding to a language.

(Refer Slide Time: 04:27)



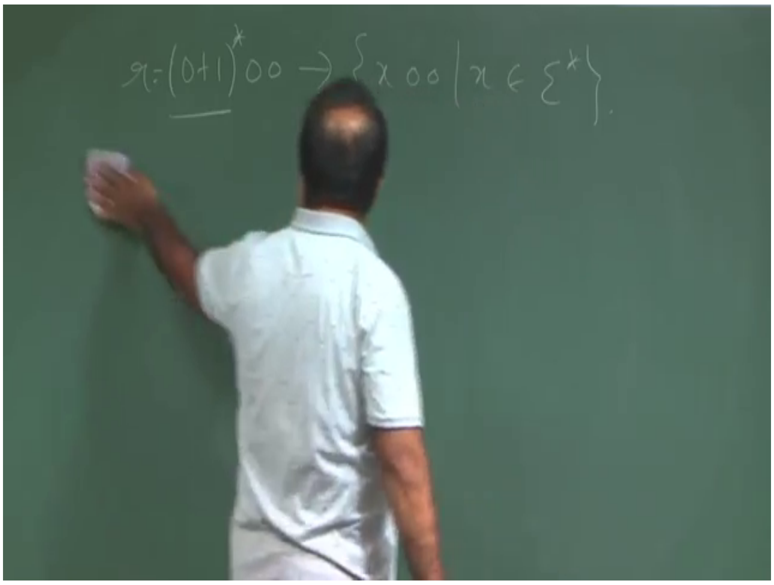
So, 0 plus 1 star ok. What is the 0, 0 is representing the 0 set.

(Refer Slide Time: 04:31)



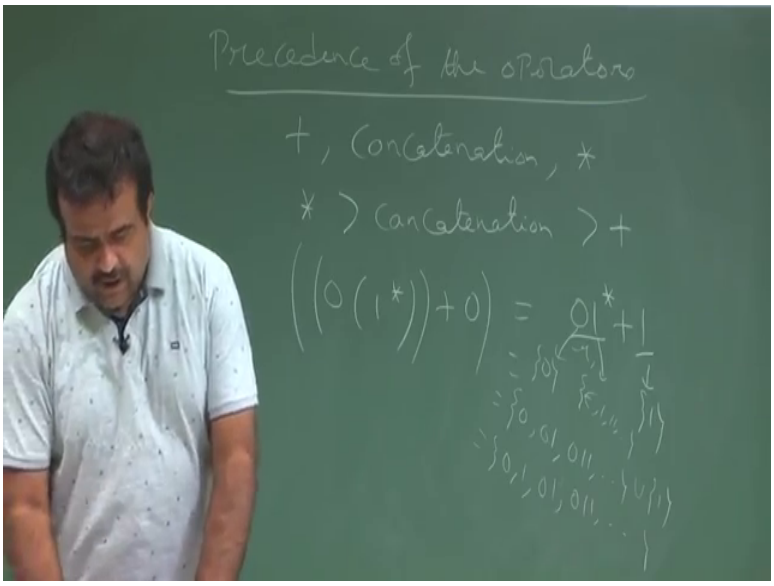
I mean 0 just a singleton set one set. Now, 0 plus 1, if this sigma s is 0 1. This 0 plus 1 will represent the sigma which is nothing but sigma. Now, 0 plus 1 star is basically the sigma star that is the set of all possible string, set of all possible string including one, including the empty string, including epsilon, the empty string that is denoting by this ok.

(Refer Slide Time: 05:35)



Now, if we if we have a say regular expression like this, 0 plus 1 star 0 0. This is also regular expression. Now, what is the language corresponding to this regular expression, the language if this is r the language is nothing but x 0 0, because this could be epsilon also or could be any other string. So, this is the language which is corresponding to the set of all string which is ending with the 0 0 ok.

(Refer Slide Time: 06:23)



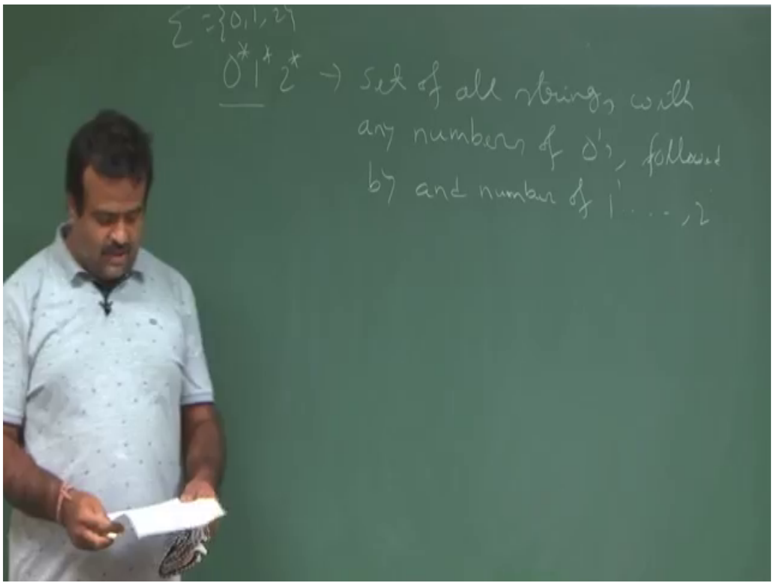
Now, then we have defined the precedence of this operator, precedence of the operators. Operators means you have three operators: plus, concatenation and star. This is the three operator we are using to define the regular expression. Now, among these the star is having higher precedence than the plus, I sorry than the concatenation than the plus.

So, star is having higher precedence than the concatenation and also called than the plus. So, this is the I mean greater that means, this is the precedence wise greater than, it is not the real number greater than ok. This star is having higher precedence than the concatenation then the plus and the concatenation is having higher precedence than the plus. So, plus is having the least precedence.

So, if we follow this rule the convention, then this we can denote by this is 1 star this is the first one, then we want to operate with 0 concatenation then 0. Now, this may be if we know this precedence, this may be written as because this is same as 1 star. We want 0, then 0 then plus 1 simplify we do not need the brackets once we know the precedence.

So, what is the language corresponding to this, the language corresponding to this is this is r 1 this is r 2. So, union of this 2, so this will corresponding to 1 and again r 2, r 1, we can write as product like 0 and this the concatenation. So, this is corresponding to 0 and this is corresponding to epsilon 1 1 1 like this set. So, once we concatenate this so this will be any string with I mean not any string, it is basically 0 then 0 1, then 0 1 1 like this and you are union in with one. So, this is nothing but 0 1, 0 1 1 like this. So, this is the this type of string will appear in this ok.

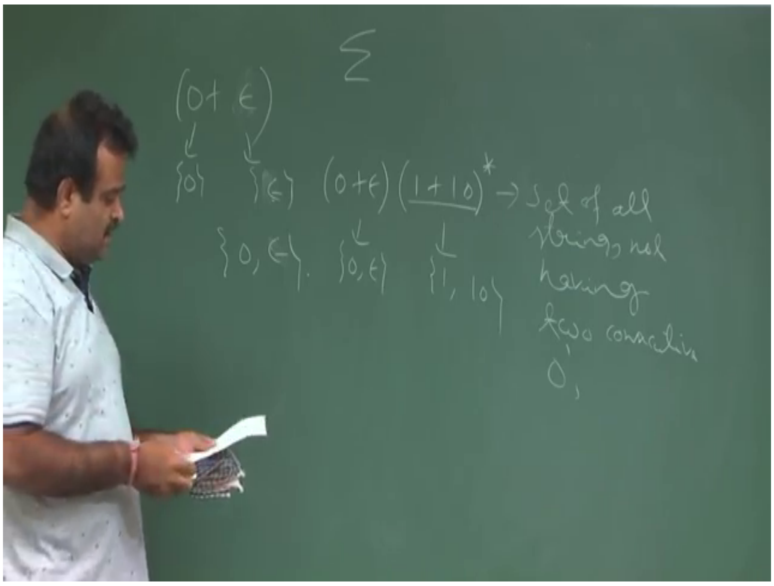
(Refer Slide Time: 09:41)



So, we can have some more example before we go to the algebra of thus of this operator ok. We know this is 1 star 0 0 star 1 star 2 star if sigma is say 0 1 2. What is the this is a regular expression. If we have all these 0 star 1 star say, then what is the regular then what is this is any number of 0's followed by any number of 1s ok. So, this is any number of 0's followed by any number of 1s.

If we take it 2 star also any number of 0's followed by any number 1 this is a set of the language is the set of all string with any number of 0's followed by any number of 1s followed by any number of 2s dot dot dot ok. So, any such string will come here ok.

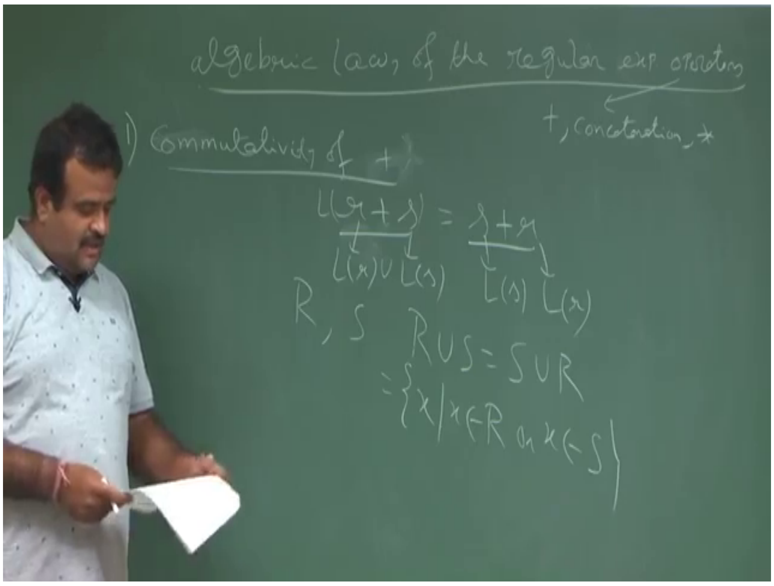
(Refer Slide Time: 10:59)



So, then we can have something with epsilon. 0 plus epsilon, what is this language, this language is this is 0, this is epsilon. So, this is nothing but 0 and epsilon, yeah 0 yeah epsilon we can write in this way depending I mean yeah. So, that we should not mix epsilon with this sigma, sigma is the set of alphabet, and epsilon is the null string, null string is denote by epsilon we know. So, anyway I mean so this is epsilon this is this set ok. So, this is this language.

Now, if you take another language say 1 plus 1 0 star along with this. So, this is we can take 0 or epsilon from here, 0 or epsilon. So, this is nothing but 0 or epsilon. And here what it will come, now inside it, it is 1 and 0 1. So, then the star means we have all combination of these two. So, this is the set of all string that that must have two consecutive 0's. So, this is the set of all string not having two consecutive 0's, not having two consecutive 0's ok. So, this is the way we can have some more example on this. Now, we will talk about that the algebra of this operator like how to simplify this like algebraic law.

(Refer Slide Time: 13:05)

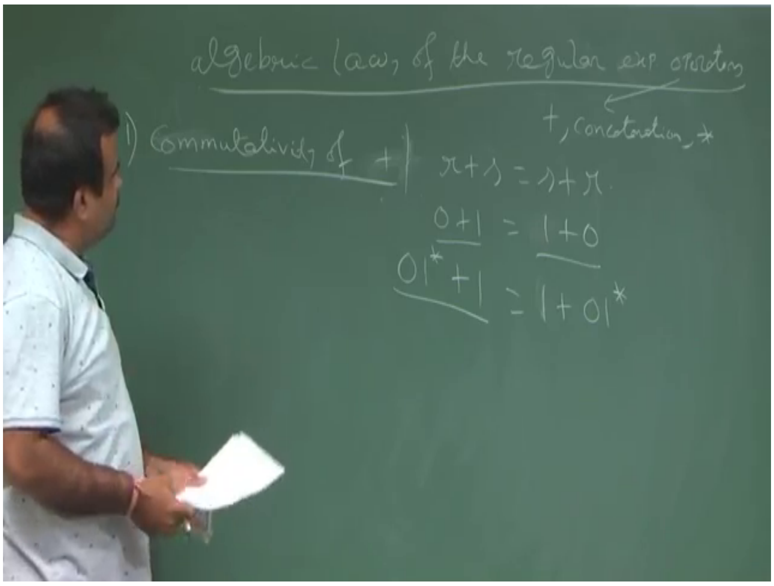


Algebraic laws of the regular expressions operators ok. So, what type of algebra we can I mean what type of algebraic law we can have. So, first one is the commutative I mean the union is commutative. We have three operator, this operator means we have three operator plus, concatenation and star ok. Plus is the union operator. So, now plus is commutative of plus or union.

So, what is that that is basically if we have two regular language r, the r plus s is also regular language which is same as s plus r. So, they are (Refer Time: 14:29). What is the proof? Proof is straightforward because this is so they are same means when you say two regular expression are same, if they are giving the same language because everything in terms of language. We are only concerned about the language which it is accepting I mean which it is representing because regular expression is a form to represent a set of language I mean easier way to represent a regular language. And these tools we are just algebraic law we are just learning now to have the more simplification. If we have this law, then we can use this law to simplify a bigger regular expression try to simplify in a smaller way, so that is the idea.

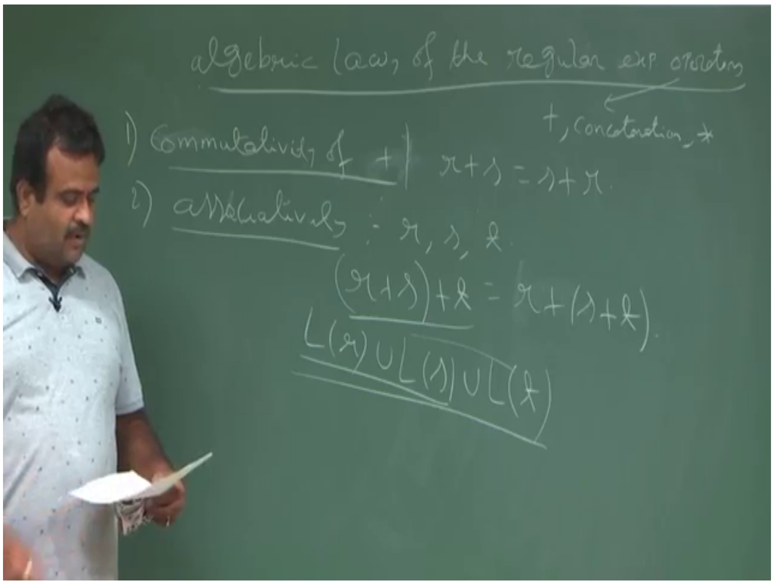
So, this is because this will corresponding to this is L of r the language this is L of s this is L of s, this is L of r now this two are union. So, the language L of this is union of this. Now, union is commutative because if we know if we have to set r and s r union s is s union r. So, union is commutative, because the union is the just a set I mean this is just a set, so that we should cover all I mean this is set of all x such that x is in R or x is in S. So, it does not matter which order they are coming I mean first we are taking from R or first we are taking for because that is the union. So, this is the commutative under union. This is the first law.

(Refer Slide Time: 16:23)



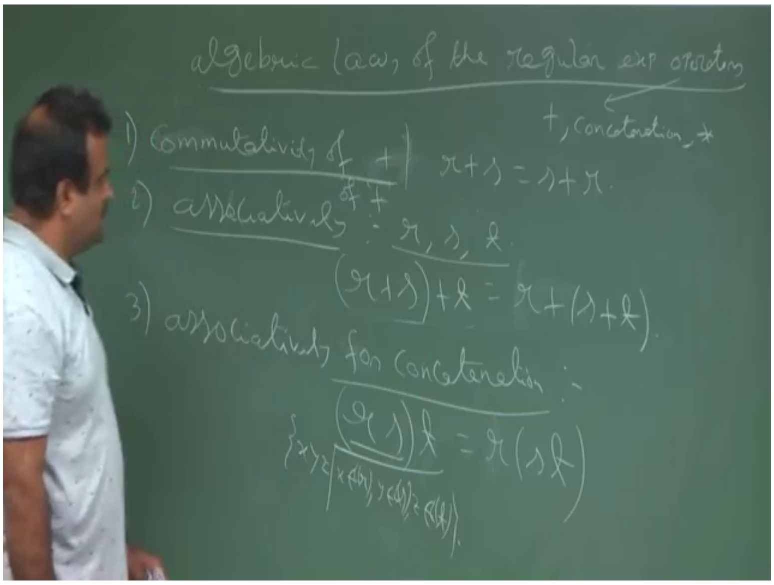
So, r plus s is equal to s plus r yeah. For example, if we have say 0 plus 1 which is a regular expression you can write as 1 plus 0 both are same. Or if we have 0 1 star plus 1 this is regular expression, we can write as 1 plus 0 1 star either all of this way ok. Now, we will check the associativity of this plus.

(Refer Slide Time: 17:01)



Associativity, associativity means so if we have three regular expression r, s, t, then say r plus s plus t is it same as r plus s plus t ok. So, this is the meaning of associativity. This is also true, because this is nothings but any one of this nothing but the language corresponding languages is L of r union L of s union L of t. So, it does not matter which way we should take. We should first take the union of this two, then union of this three or you should take union of this two then union of these three ok, it does not really matter because this is a union itself is a associativity property is there ok. So, it is associativity is there.

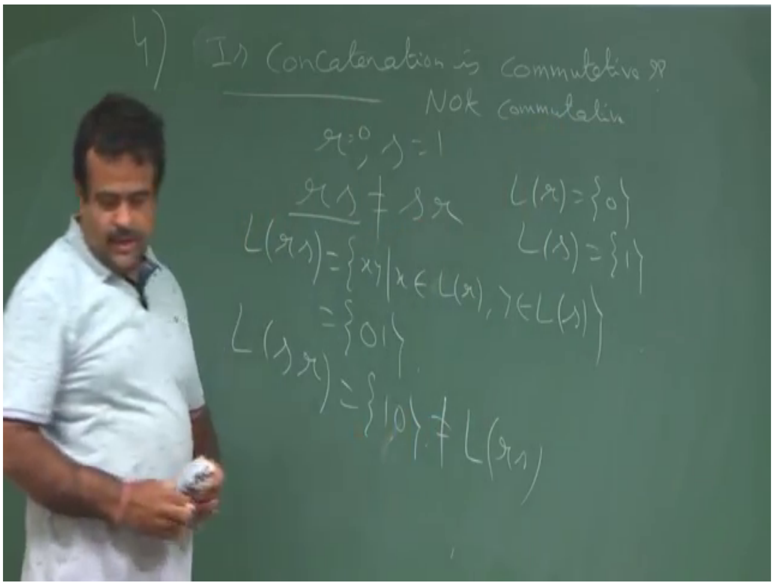
(Refer Slide Time: 18:17)



Then the third property is associativity in plus or in union. Now, we will check whether this is associativity they are for concatenation is also because plus. And associativity for concatenation so that means, so if you have three regular expression, we want to check r is then t whether this is same as r, s t ok. This is the meaning of if this is true for all such regular expression. So, when you say two regular expression is same, if they are representing the same language because everything in terms of the language.

Now, what is the language corresponding to this? So, language corresponding to this is set of all x y and we have a t. So, set of x y z where x is coming from r, y is coming from s, and z is coming from t. I mean L of that sorry L of that ok, this is the concatenation operation. So, now, it does it does not matter whether you read this x y, then z or x or y z then x. So, they are same, because they are basically the x, y, z. And here also we have x, y, z. So, these are same. So, this is associativity is there for concatenation. But the commutativity is not there for concatenation that we will check.

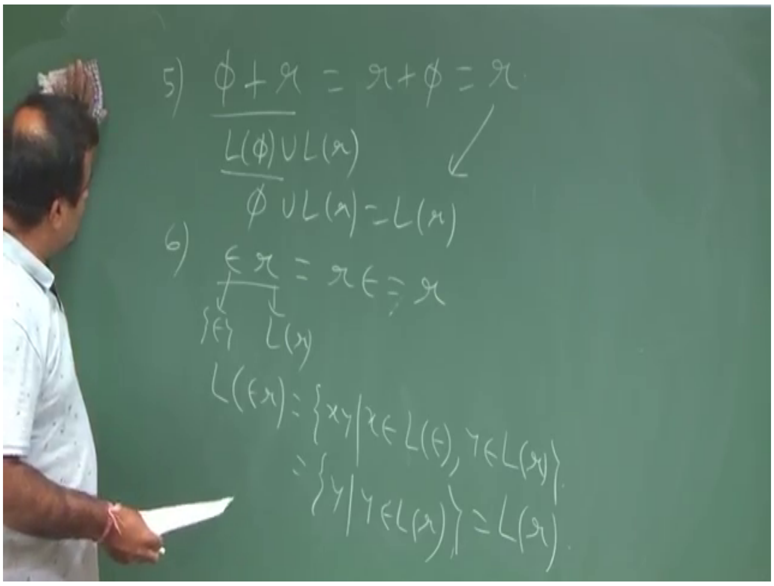
(Refer Slide Time: 20:27)



So, we will check whether this is the fourth property is concatenation, concatenation is commutative; that means, if you have two regular expression r and s, then is r s equal to s r, this is a concatenation operation. That means, the if the language corresponding to r s is same as language corresponding to s r. So, what do you think? No, why because we can take very small example. Suppose r is 0 s is one. So, what is L of r L of r is just 0, and L of s is the singleton set 1.

Now, what is the language corresponding to r s, this is the x y the definition is x y, x is coming from r L of r, and y is coming from L of s. So, this is nothing but all the one set 0 and 1 because there is only one element and what is language of s of s r, this is 1 0. So, these two are not same, it is not same as. So, it is not same. So, it is not same in general ok. So, this is not commutative. So, this is like this concatenation is not commutative ok. Now, we will take some more example ok.

(Refer Slide Time: 22:39)

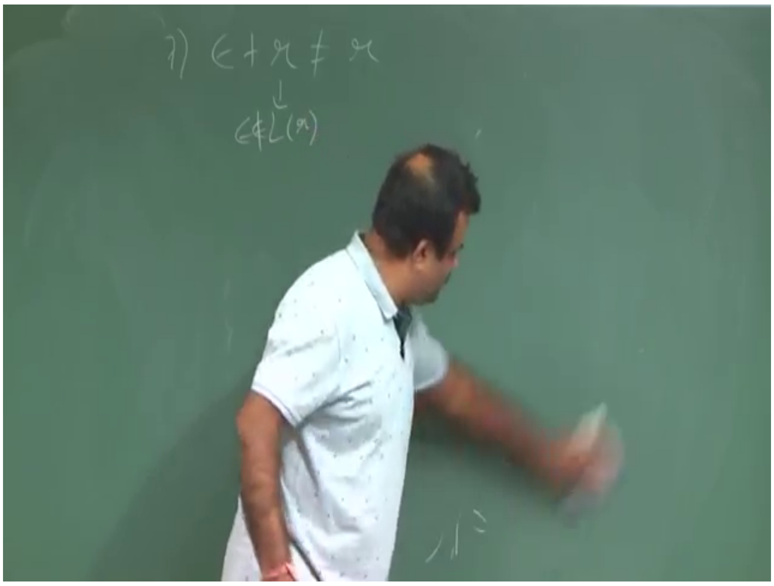


So, phi plus r is same as r plus phi anyway plus is commutative we know, but only thing you need to show phi plus r is r. How to show this? This is also easy because what is the language corresponding to this it is this union this. Now, what is this is phi union L of r this is basically L of r which is same as the language corresponding to r. So, this is a another property which is straightforward.

Now, 6, epsilon r is same as yeah here it is we can write r epsilon is equal to r. How to check this? This is also easy to check because this is nothing but this is corresponding to epsilon, and this is corresponding to L of r. Now, what is the product the concatenation, concatenation is basically so L of epsilon r is nothing but L of epsilon r is nothing but so set of all string x y where x is coming from this set L of epsilon and y is coming from L of r.

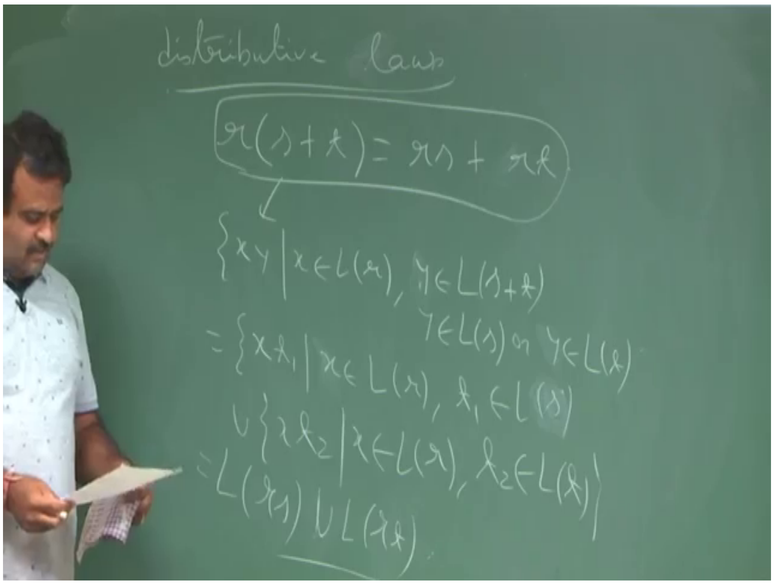
Now, L of epsilon is only epsilon. So, if you add with epsilon, epsilon is nothing, it is just a symbol it is a string of length 0, it is that it does not exist. I mean say so this is same as y basically y where y belongs to L of r. So, this is nothing but L of r ok. So, epsilon r is basically r over here.

(Refer Slide Time: 24:53)



But in the plus for concatenation is true, but for plus it is not true, why, for the plus. So, this is 7. Epsilon plus r is not same as r unless L of r contain epsilon, if L of r if epsilon does not belongs to L of r then this true set are not same ok. So, but for concatenation it is ok.

(Refer Slide Time: 25:23)



Ok, now we talk about distributive law, distributive law ok. So, this is telling r this is the left distributed law r plus s into t is equal to r s plus r t ok. So, this is also we can easily verify because what is the language corresponding to this, the language corresponding to this is x y where x is coming from L of r and y is coming from L of s plus this ok.

Now, this is same as y is coming from either this or y is coming from L of t. So, that, that means, this set we can write as x t 1 where x is coming from L of r, and t 1 is coming from t 1 is coming from L of t L of s sorry L of s L of s. Union of x t 2 where x is coming from L of r and t 2 is coming from L of t, so that means, this is same as language by r s union language of r t. So, then these two are same ok. These are these are the properties we require for the simplification purpose ok. Then we have the more property which we will discuss in the next class.

Thank you.