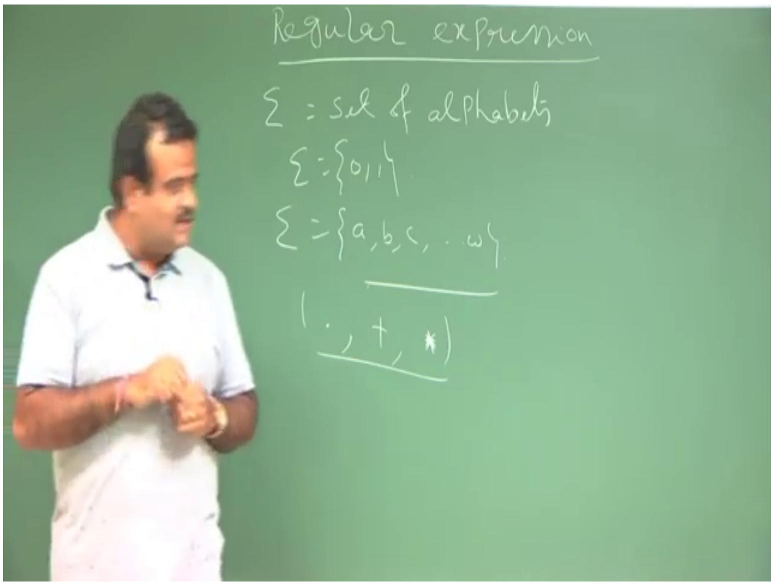
Introduction to Automata, Languages and Computation

Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology Kharagpur

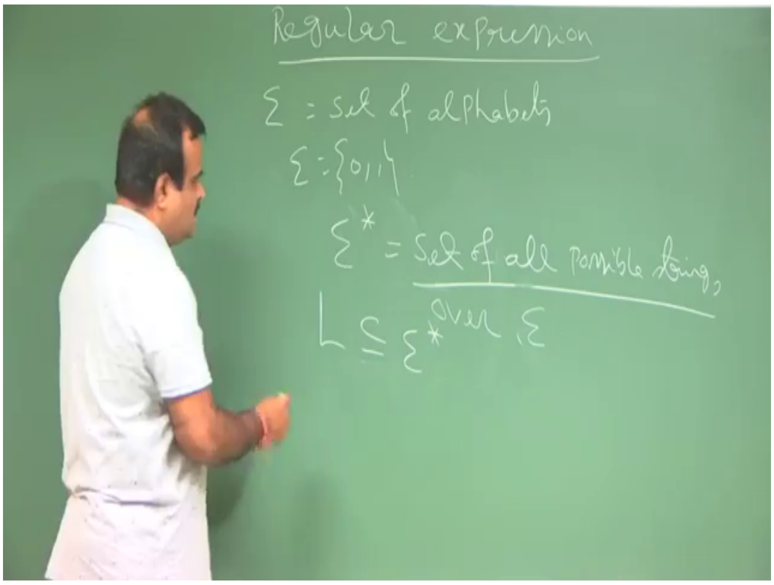
Lecture -16
Regular Expression

(Refer Slide Time: 00:17)



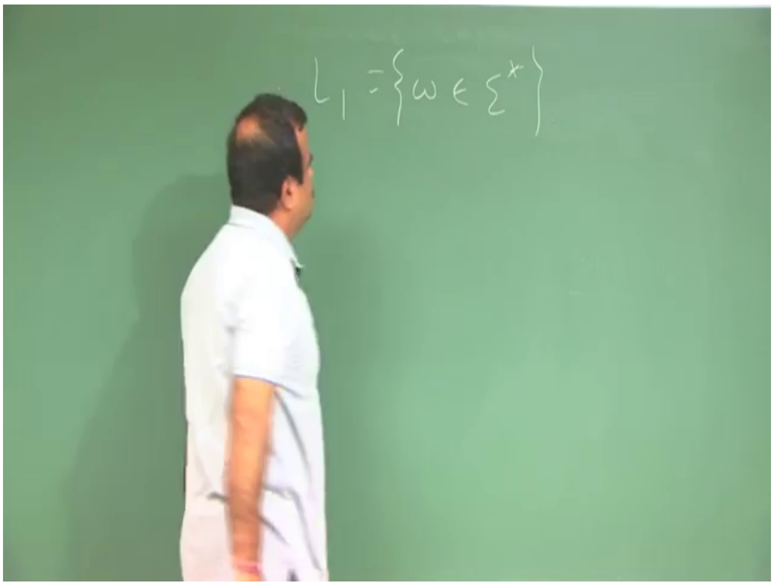
So good morning everybody, so we will start with the Regular Expression and the corresponding language of a regular expression. So, suppose we have a this is the set of alphabet this set is finite as we know this set could be 0 1 if it is a binary alphabet, it could be any anything like a b c d but this set is finite like c d w something like that this set is finite. So, we will talk about the regular expression and the corresponding language of that regular expression and that is defined by the recursive way with the help of 3 operations like product, then plus and star with this 3 operation we will define the regular expression in a recursive way ok. So, for that let us define some operation on the language. So, we know the language is the so we know sigma star.

(Refer Slide Time: 01:37)



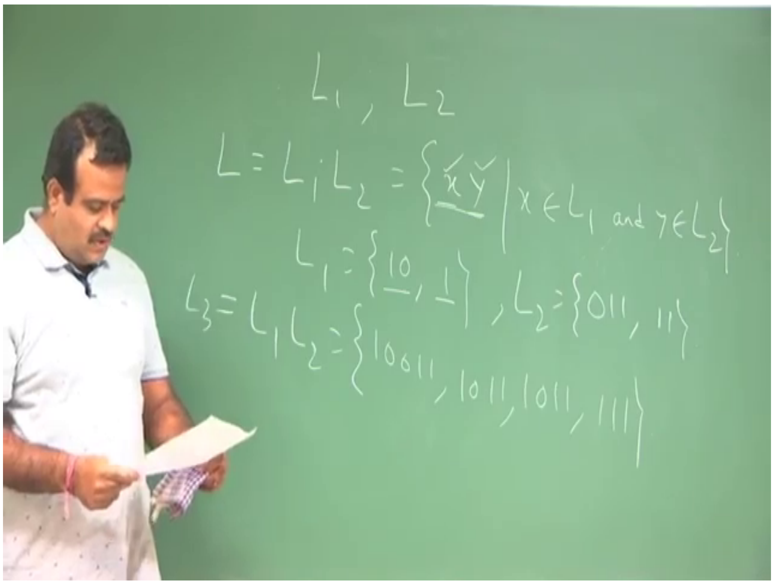
Sigma star is the set of all possible strings possible string over sigma this is our sigma star and any subset of sigma star is called a language any subset of sigma star is called a language. Now we will define product of 2 language. Suppose we have a language L1 and L2 then what do you mean by L1 into L2 that will be again a language. So, how to define that?

(Refer Slide Time: 02:27)



So, suppose we have a language L1 this is a string coming from sigma some sort of string I mean L1 is a language L2 is a language.

(Refer Slide Time: 02:37)

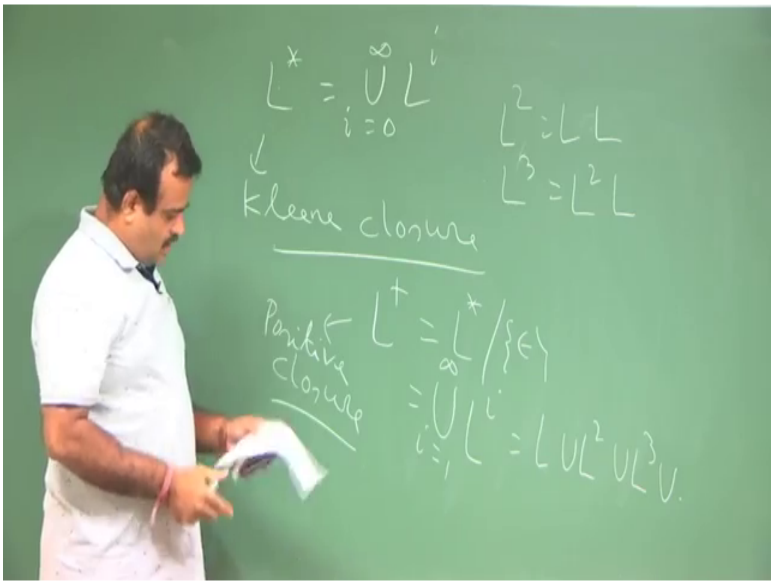


Then we want to define the product L1 L2. So, this is nothing but x y this is string y is a string x is coming from L1 and y is coming from L 2. All possible combination of such x y will be in this language. This is a new language again this is a subset of sigma star. So, this is how we define the L1 L2.

So, we can take an example. Suppose L1 is say once this one and say L2 is say 0 1 1 and 1 1. Then what is L1 into L2? So, x y so x will come from here y will come from here, so for if we take this x we can have we have 2 options. So, there are how many so there are 4 elements will be there in this set, so 1 0 with 0 1 1 and then 1 0 with 1 1 this is with this x.

Now, with this x we have 1 0 1 1 1 1 1 ok. So, this is our L1 and L2, all possible x y; x is the string coming from a L1, y is a string coming from L2 ok. Then this is the language L3 then we can define like L3 L2 again by the similar way we can continue with this. So, now we define another operations which is star, if we have a language L then what do we mean by star. So, star is nothing but this we already know this symbol.

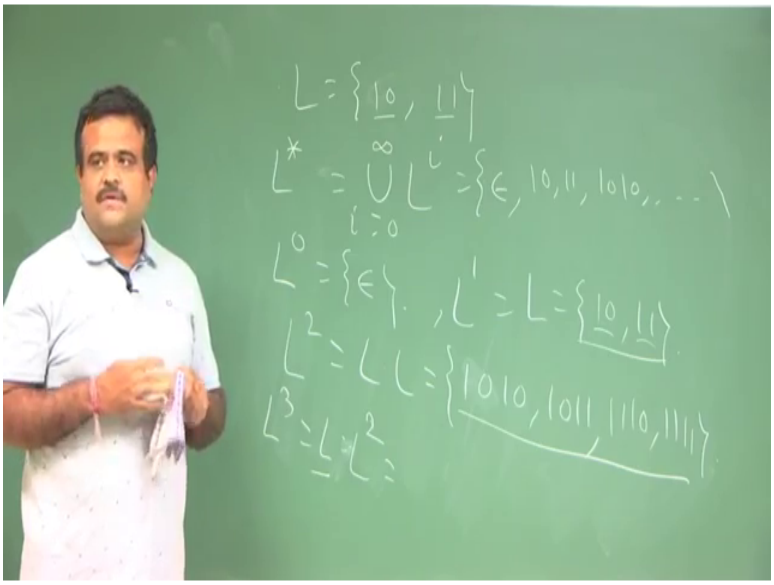
(Refer Slide Time: 04:51)



L star which is nothing but union of Li, i is from 0 to infinity and this is called Kleene product closure. Now if we L 0 means it is the epsilon is included there, if we do not take want to take the epsilon. This is the set of all string coming from L with different length; length 1, length 2 length 3 like this. So, this is if we take a length 0 that will eventually give us the epsilon. Epsilon is the string of length 0 ok.

Now, this L plus is nothing but if we exclude epsilon. So, this is nothing but union of Li, i is equal to 1 to infinity. So, this L is the so this is basically union of L, L square, L cube like this. So, L square means L into L just now we have seen the product L cube is L square into L like this ok. So, this is the way we define. So, this is called positive closure positive closure and this is called ok. So, now we take an example then we will see how we can have this L star ok.

(Refer Slide Time: 06:47)

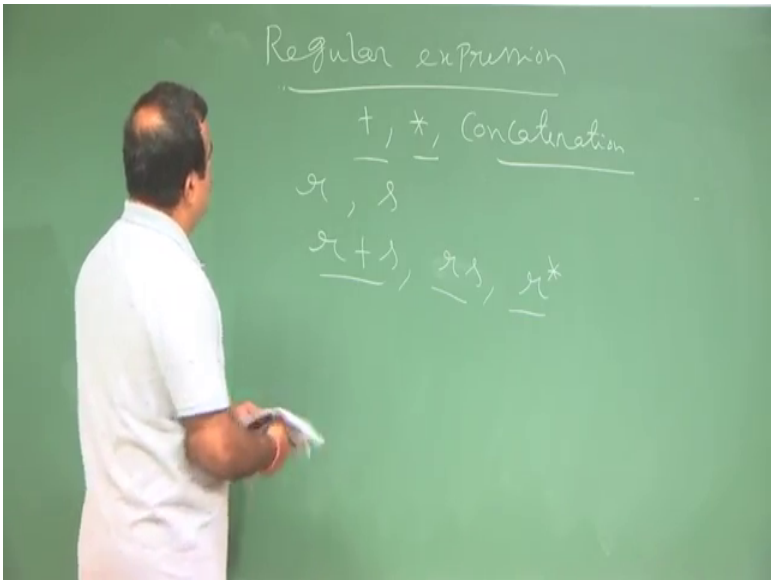


So, let us take a L now, so say a L is the 0 1 1 1 or may be 1 0 this is our L, now we want to find out L star. So, to get L star you have to, so L star is nothing but union of Li, i is equal to 0 to this. So, what is L 0? L 0 is nothing but the string of length 0 which is nothing but epsilon. Now what is L1? L1 is the L which is 1 0 1 1 and L2, L2 is the L into L. So, this is therefore this is the collection of x y this is all x y; x is coming from L, y is coming from L. So, this will be nothing but so we take this one zero 1 0 1 0 1 1 then we take this as x 1 1 1 0 1 1 1 1; this is our L 2.

Now, how to get L 3 L 3 is nothing but L into L into L2, now L2 we know and L we know L is nothing but this. So, again this is a collection of x y x is coming from this set y is coming from this set, so this will be how many elements for this x we have 4 options for this x we have 4 options, so there are 8 elements over there ok. So, this way we can just take the union of this all and that will be the L star. So, L star is basically this epsilon and L1, then L2, L 3 so on this collection is called L star ok.

So, this way we can have L star. So now we will define how we can I mean the we will define the regular expression and it is corresponding language with the help of 3 operations plus product and star. So, let us do that.

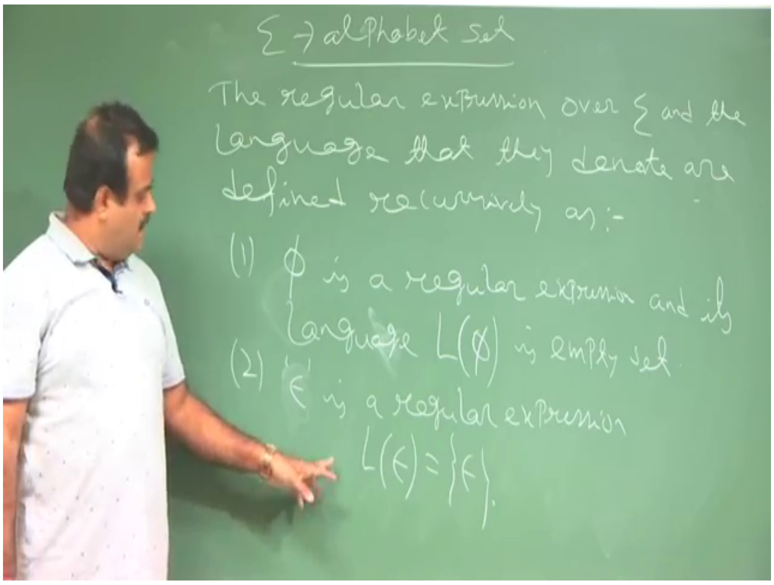
(Refer Slide Time: 09:29)



So, Regular expression- so this will be defined based on these 3 operation concatenation; concatenation means which is basically product ok. So, this will defined recursively like and if we if r is irregular these are the possible operation on the regular expression, r is a regular expression and s is a regular expression, then we can r plus s r into s or r s like yeah r into s or r star. These are these 2 are binary operator on this and this is the unary operator on this, so these are all regular expression.

So, we will define this in a recursive way, I mean like inductive way will define this. So, let us define that. So, for that let us take sigma, so this will be regular expression over a alphabet set.

(Refer Slide Time: 10:49)

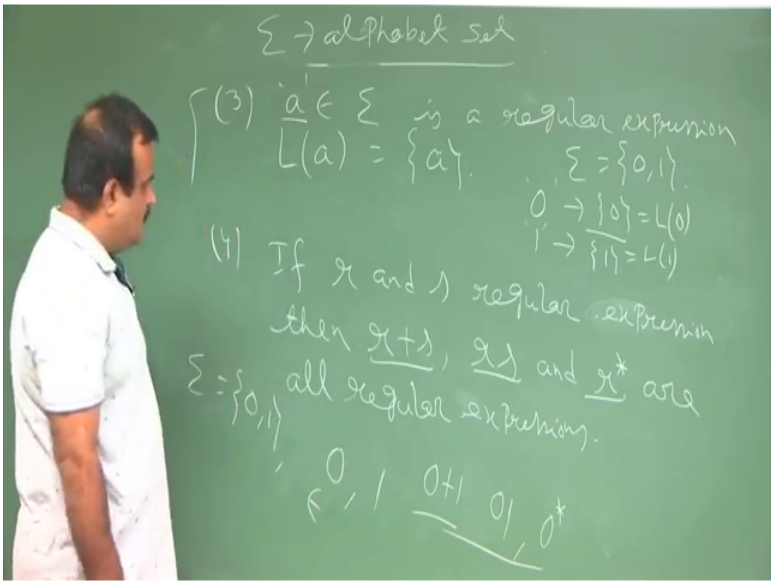


So, let that this is the alphabet set which is finite, then we are going to define the regular expression over this and the corresponding language and the language. They denote language that they denote are defined recursively as recursively as this.

So, first one is phi denote a regular expression phi and this is corresponding to phi is a regular expression. This is the recursive definition. This is the base case you can say phi is a regular expression. This is the recursive definition; this is the base case we can say phi is the regular expression expressions and it is language is the empty set it is language. Language we denote by L of phi set because, phi is a regular expression L of phi is empty set.

This is one condition this is on the this is on base case and sigma I mean epsilon is also a regular expression, epsilon is a regular expression regular expression and what it is language denotes. So, it is languages denote by L of epsilon is nothing but the singleton set epsilon. So, epsilon is corresponding to the language singleton set epsilon and then any other alphabet from sigma that is also a regular expression.

(Refer Slide Time: 13:35)

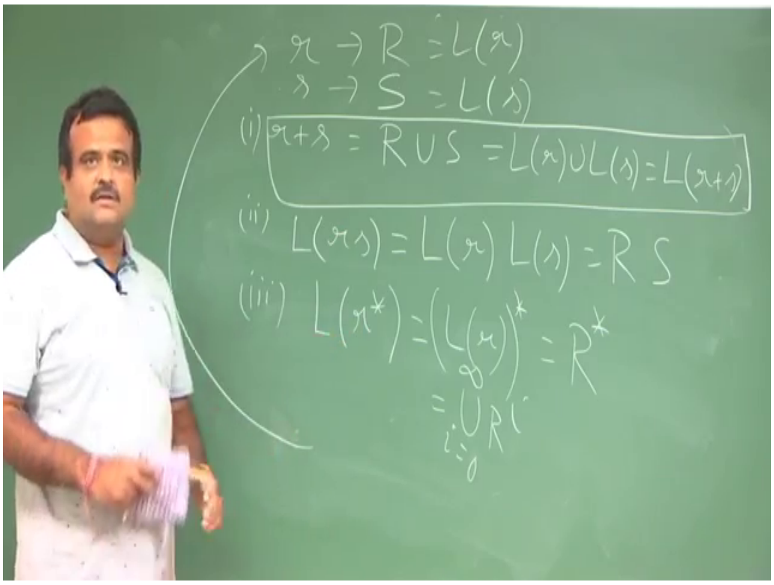


Any a belong to sigma is a regular expression and which language it will correspond, so that will denote by L of a is a regular expression language corresponding to this is the singleton set a, where a is a where a is a alphabet a is an alphabet ok. Like if sigma s is 0 1 then 0 is a regular expression which is corresponding to the set 0, I mean the singleton set this is a language which is corresponding to the language this language and one is also a regular expression which is corresponding to the this.

So, these are the base case like based on this then we will use that operator to have the regular expression in the upper case, so these are the this is the definition. So now, the 4 number 4 number 4 will define recursively ok. So number 4 this is the recursive definition if r and s are regular expression regular expression, then r plus s is also a regular expression, r into s is also regular expression and r star or s star is the binary these are all are all regular expression.

So, now the question is which language it will correspond that will come regular expression, these are all regular expression yeah like if we if sigma is 0 1 then we know 0 is a regular expression 1 is a regular expression. So that means, 0 plus 1 is a regular expression 0 1 is a regular expression and 0 star or 1 star is a regular expression and this is a regular expression. So, we can have combined with this also empty set is a regular expression we can have combined with this. Now we have to know what are the language corresponding to this regular expression, so let us just define the.

(Refer Slide Time: 16:49)



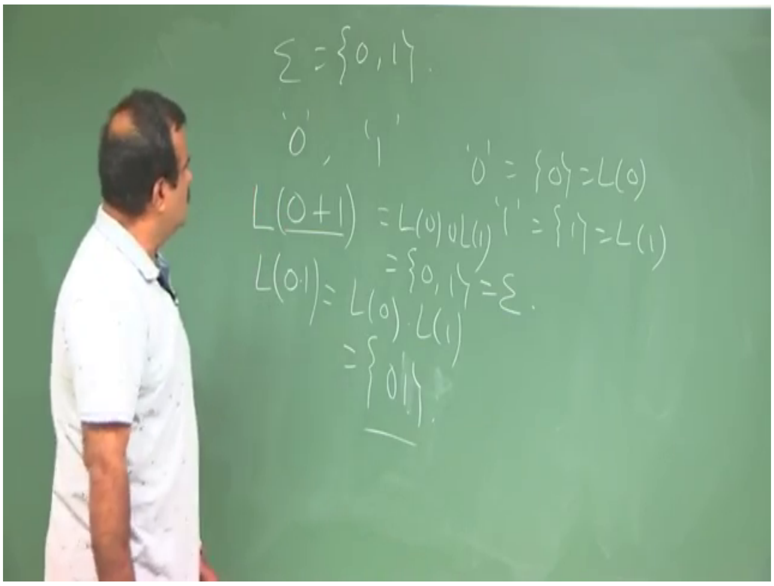
So, suppose yeah we write here no ok. So, suppose r is corresponding to the language capital R which is nothing but u of r and s is corresponding to the language capital S which is nothing but L of s we know the base case ok. If it is singleton then we know they are the singleton, I mean if it is just a input symbol or epsilon we know they are the singleton language ok.

Now, what is the regular expression corresponding to r plus s. So, r plus s is nothing but R union S. So, this is nothing but L of r union L of s, so this is our L of r plus s this is our L of r plus s ok. Now the product now the product is so what is L of, so we know r into s is a regular expression L of r into is nothing but L of r this is L1 into L of s, we know product like this is x y. So, x plus string x will come from this set string y will come from this set. So this is nothing but RS and then the third one is these are the language corresponding to denotes this by regular expression.

Then r star so L log r star is nothing but L of r star (Refer Time: 18:50) star. So, this is basically r star we know R star. If we know the language we know the star, so this is nothing but union of R to the power i is equal to 1 to infinity sorry 0 to infinity; this star ok. So, this is the recursive definition of regular expression, so if we know r is a smaller is a regular expression small s is a regular expression then we can recursively define the r plus s r into s r star or s star. They are all the all will corresponding the regular expression and the language of that regular those regular expression is defined like this, so this is the definition recursive definition.

Now we will take some example then will move to the some precedence of this operation this plus product this is the concatenation and this term. So, based on this definition let us take some examples on regular expressions and their language.

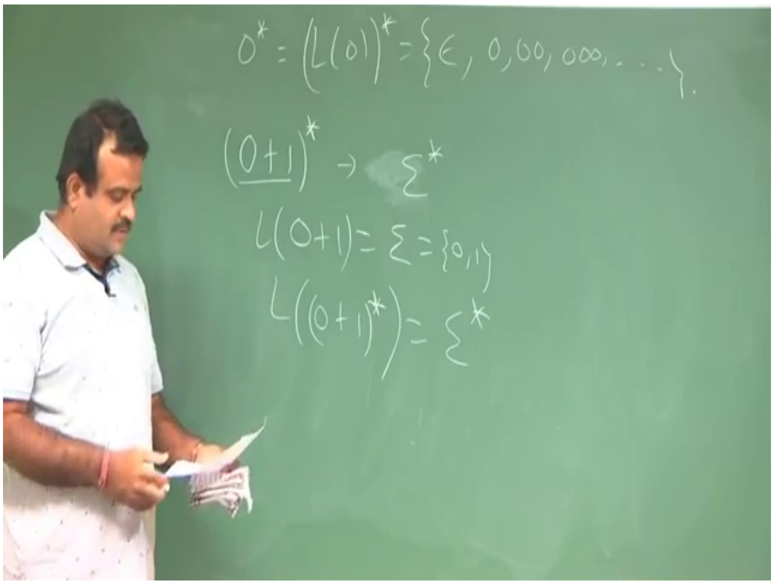
(Refer Slide Time: 20:01)



So, let us take simple one like if we have sigma s is 0 1 ok, then we know 0 is a regular expression and 1 is a regular expression then 0 plus 1 this will be a regular expression. So, what is the language corresponding to this so yeah, so 0 corresponding to 0 this is L of 0 and 1 corresponding to 1 this is our 1 L of 1. So now, this will correspond to L of 0 union L of 1, so this is nothing but 0 1 so sigma itself ok. Now 0 1 0 this concatenation, so this will corresponding to like this will corresponding to the this is the expression and this is the language you can say L of that ok.

Now, you can say a L of this is nothing but L of 0 product of L of one. So, L of 0 is L of 0 is 0 product of 1, so this is nothing but 0 1 this 1 ok.

(Refer Slide Time: 21:29)

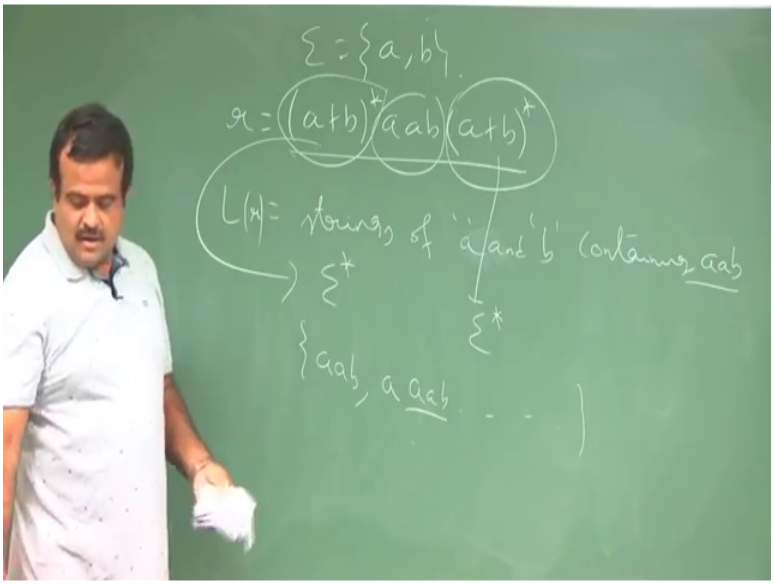


Now, what is 0 star or 1 star, so 0 star 0 star means so this is the language so this is L of 0 L of 0 star. So, this is nothing but epsilon 0 0 0 0 0 0 all possible all every length of this expression. Now we can take some more examples like if we have say, yeah so like 0 plus 1 star which language it will denote this is a regular expression this is a regular expression, because we know this is a regular expression this star is a regular expression.

Now this is nothing but sigma this will corresponding to sigma star because, this is 0 plus 1 is nothing but sigma which is corresponding to language of this we have to be careful when you talk about expression and the language, expression is just a the expression of using the alphabets and the language is the corresponding language of that expression so that is there.

Now if we have say yeah, so this is basically this language. So, language of this is so language of this is nothing but this is the set of all possible strings ok. Now if we take some more example on this so.

(Refer Slide Time: 23:35)

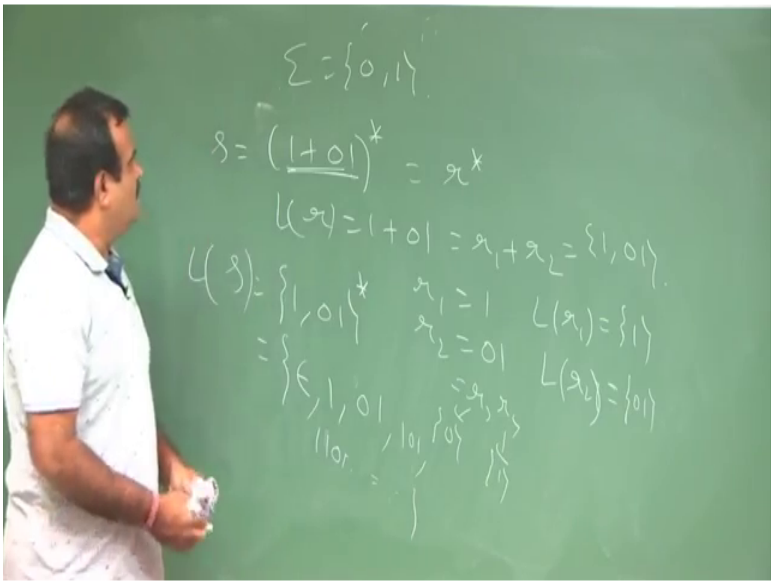


So, if our sigma is a b, now if you take a plus b star a b then a plus b star. Now this is a regular expression star, now we want to know what is that L of I the language corresponding to this regular expression because, eventually we will talk about the languages the language is accepted by the automata. So, we will come to know in the next class that how this regular expression is corresponding to a automata, given the regular expression we have a automata which is accepting that we have a automata epsilon and f a which is accepting the language of that regular expression. We will discuss slowly ok.

So, yeah so what is the language corresponding to this. So, language corresponding so this is one part this is other part this is other part. So, this is the language so L of r is nothing but this is the strings of a b strings of a and b which containing a a b which containing a a b as a substring.

So, that is the that is the expression ok, because this is this includes what this includes sigma star this is also sigma star. Now it would be epsilon also any combination of this, so this is basically if it is epsilon this is also epsilon this is a b. Now if it is a, then a a b and this is epsilon like this. So, all possible combinations of this is this.

(Refer Slide Time: 25:57)



We take another example like if we take sigma s 0 1, then what is we know L 0 0 L 0 0 is just a singleton set 0 0 and then how to write say sorry not L 0 0 then we want to write this like 1 plus 0 1 star we want to find it. We know this is a regular expression because 1 is a regular expression these 2 is a regular expression then their star is also a regular expression. Now you want to find the language corresponding to this regular expression. So, how to find this? So, we know this is a regular, so we have to just this is we can treat this r star. So, what is r r is 1 plus this ok.

So, this is this is say s now r is nothing but is equal to r 1 plus r 2 what is r 1 r 1 is one and r 2 is 0 1. So now, r 1 corresponding to the language L of r 1 is nothing but 1 and L of r 2 is nothing but again r 2 is some r c r 4 this is 0 corresponding to the language. So, this will corresponding to the language singleton 0 1. So, this will corresponding to the language one 0 1.

So, this is nothing but 1 0 1 star, so the language so this is this is r r is 1 0 1 language corresponding language is this. So, this s is nothing but 1 0 1 star language of s. So, this now we can have so star means we have epsilon, then all possible combination length 1 then length 2 means we take this length 2 then length 3 we take we may take 1 1 then we may take I mean this with L. So, like this 1 1 0 1 or 1 0 1 sorry 1 1 0 1 like this, so this is the language corresponding to this. So, we will continue this in the next class.

Thank you.