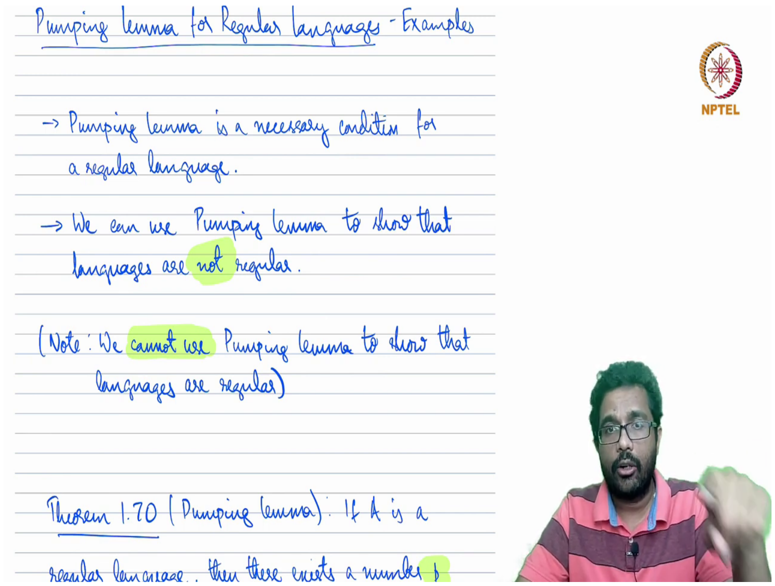
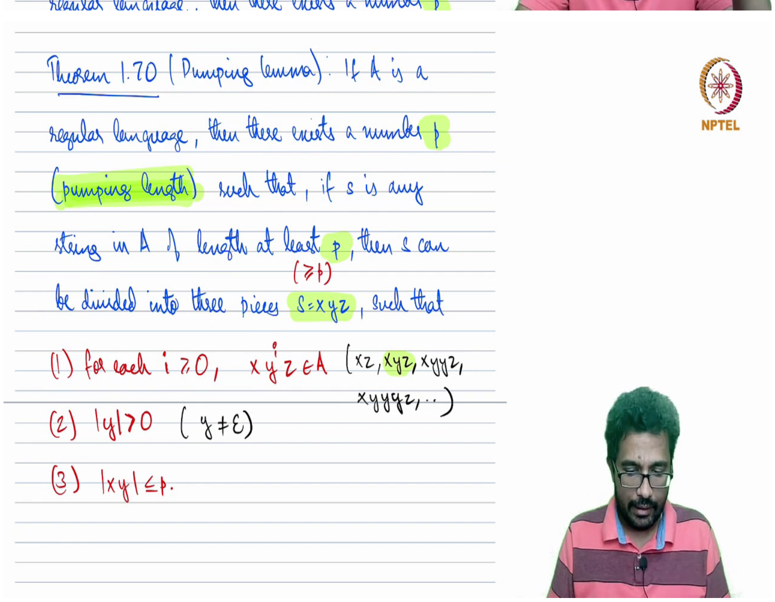
Theory of Computation
Professor Subrahmanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Pumping Lemma for Regular Languages - Part 02

(Refer Slide Time: 00:17)





Hello and welcome to lecture 13 of the course theory of computation. In the previous lecture, we saw pumping lemma for regular languages. So, we saw the, what the pumping lemma statement was and we saw the proof of the pumping lemma. In this lecture, we will see how it can be used to show that some languages are not regular.

So, just to recap, pumping lemma is a necessary condition for a language to be regular and we can use pumping lemma to show that languages are not regular. So, it is a necessary condition. So, meaning if a language is regular it has to necessarily satisfy the conditions of the pumping lemma.

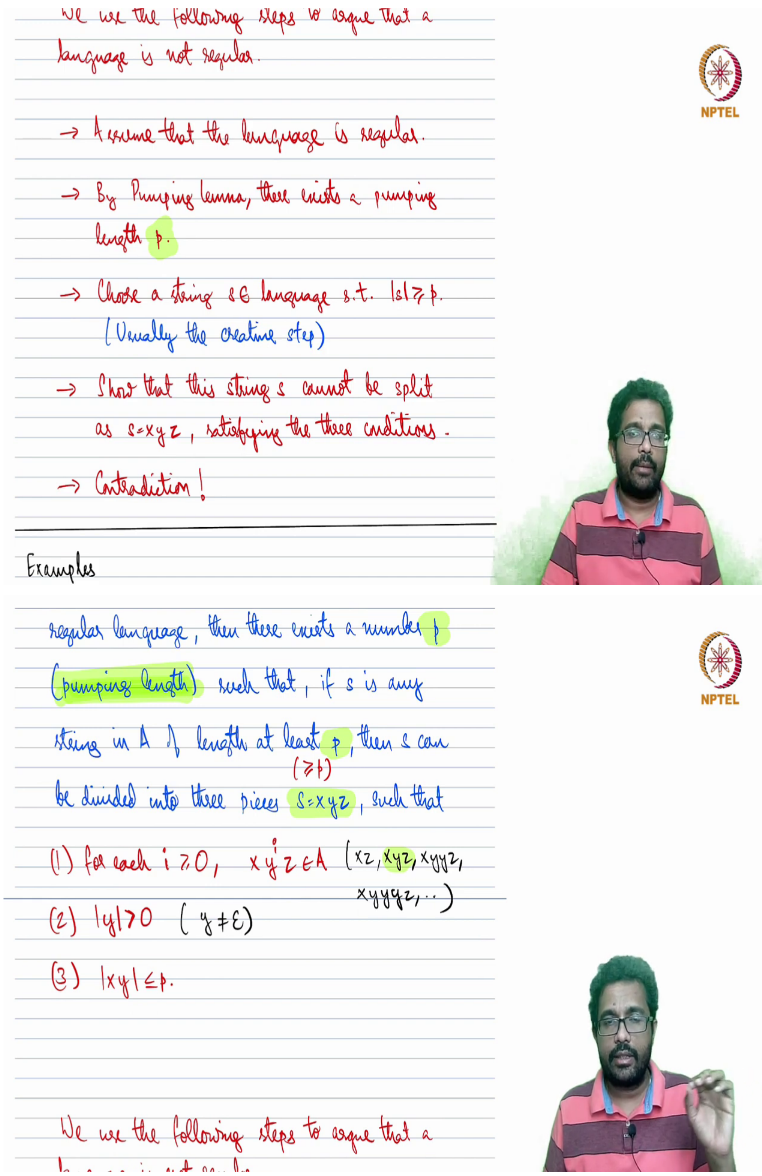
So, you can take a language and show that it does not satisfy the conditions and hence it is not regular, just because it satisfies the conditions does not imply that the language is regular. So, this is very important you can use pumping lemma to only show that languages are not regular, you cannot use pumping lemma to show that languages are regular.

So, just because it satisfies the condition does not imply that, the language is regular what you can infer is that because it does not satisfy the language has to be necessarily not regular. So, this is what pumping lemma is and this is what it can be used to show. So, just to recap pumping lemma states if A is a regular language there is a pumping length such that any string s that is a member of A, and of length at least the pumping length.

So, pumping length here is p then we can split S into three pieces x y z, such that it satisfies the three conditions the three conditions are

one that x y power i z is an A. So, xy^iz could be having 0 repeats of y, 1 repeat of y, 2 repeats of y, 3 repeats of y and so on. It is an infinite set of stirings, second is y is not an empty string and third is that the length of x y is at most p.

(Refer Slide Time: 02:34)



So, usually this is the sequence of things that you do when you want to show that a language is not regular. So, what you say is if you are given a language first, the pumping lemma says that if the language is regular it meets these conditions. So, you assume that the language is regular, suppose you are given a language and you want to show that it is not regular.

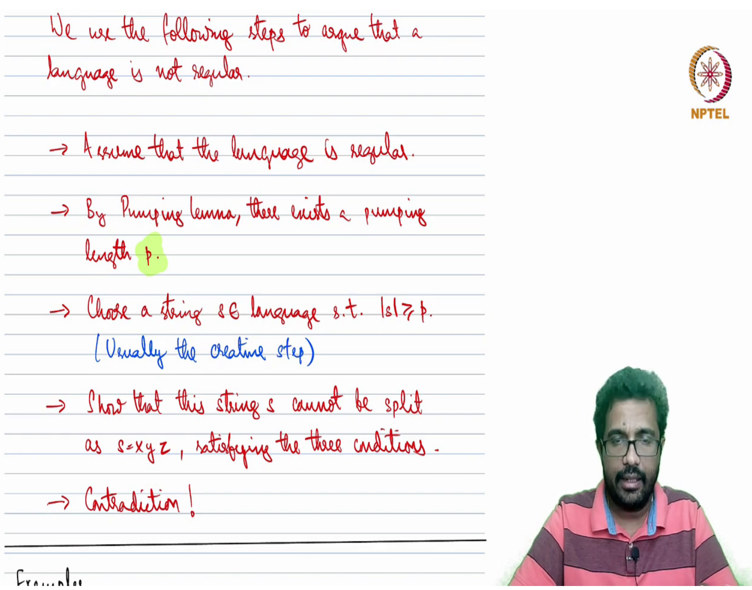
So, assume that the language is regular and because the language is regular, or because you assume that language is a regular pumping lemma implies that there is a pumping length. So let us call that pumping length p, now chooses strings in the language.

And the string has to be of length greater than the pumping length. So, usually, the pumping length, we do not know what it is, because I assume that it is some variable p or some parameter p. So, this string can depend on the p itself? It could say it could have p in its description, even though we do not know what P is.

And you choose a string that is of length at least p, that is in the language as well, and show that this string cannot be split as x y z satisfying the conditions of the pumping lemma. So, the pumping lemma says that any string that is in the language of length at least p can be split satisfying these conditions.

So, to refute that, you want to show that the language is not regular, you want to arrive at a contradiction. So, you take a specific string, this is completely our choice or your choice. And show that there is absolutely no way that it can be split into x y z that satisfies these three conditions. So, that is the rough game plan that we have.

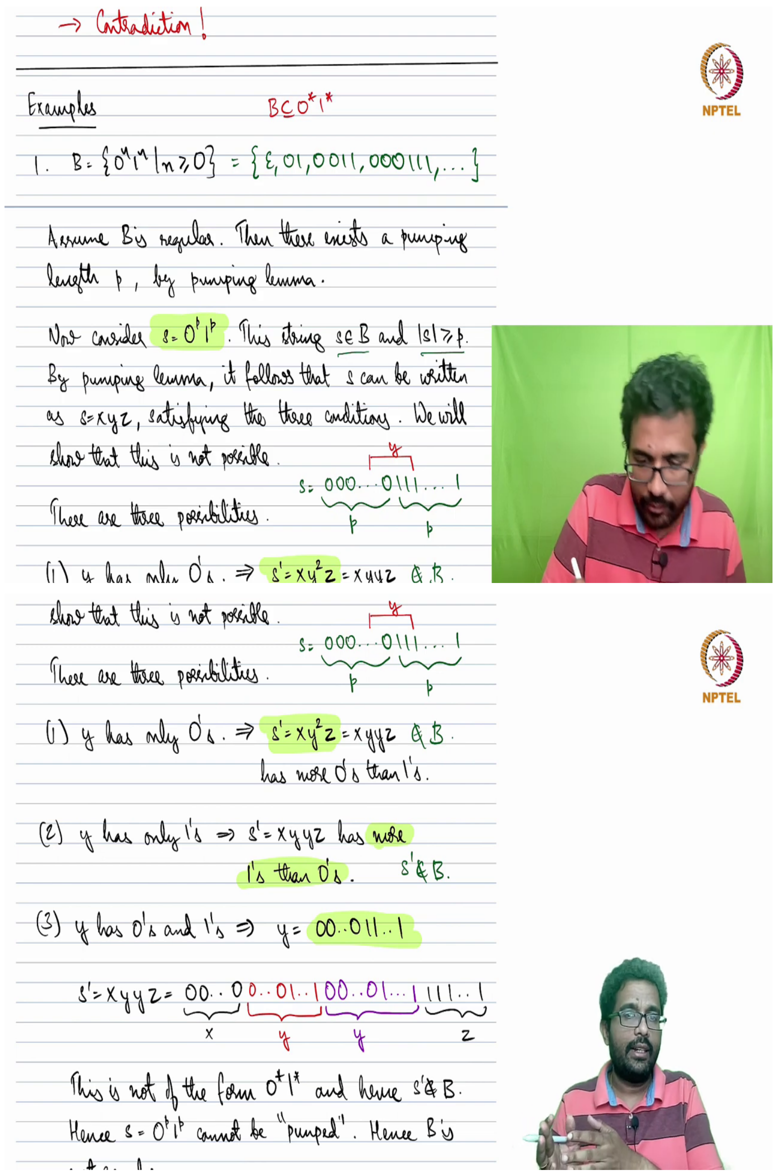
(Refer Slide Time: 04:29)

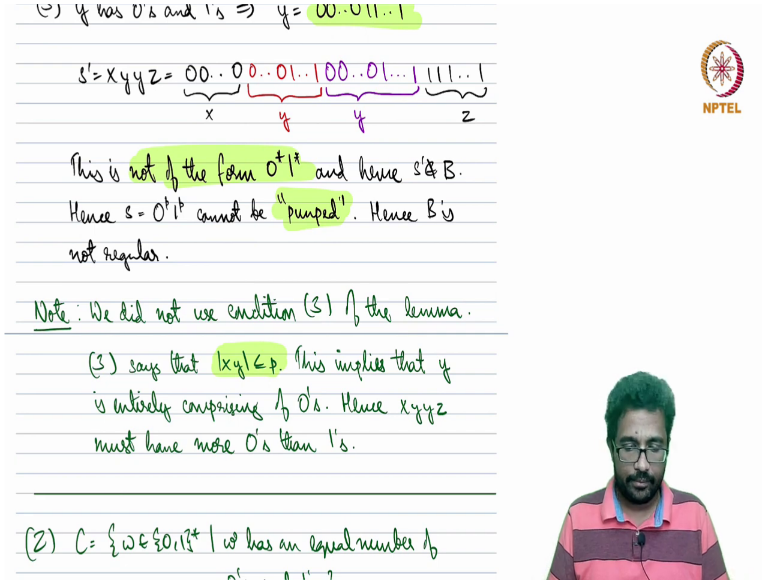


So, again, I want to reiterate two things. One is that the choosing of the string is usually the most important step. And it is also the most creative step involved. the rest follows fairly systematically. This is where one has to probably get a bit creative. So, the string could be anything, it can be any string that we like, which is in the language and is of length at least p. But then the choice of a string is entirely our choice.

But once having chosen the string, having chosen the string, we have to show that there is absolutely no possibility of breaking this string down into x y z which meets these conditions, so we have to rule out any possible split of x,y,z saying that if you split it this way, some condition gets violated if you split it that way some other condition gets violated and so on. So, you have to only pick one string, and for that string, we have to show that none of the splits satisfy the conditions and that is how you get the contradiction.

(Refer Slide Time: 05:51)





So, maybe, let us see an example and that will make things clearer. So, the first example which is the most common language that is not regular is B which is 0 power n, 1 power n, this is the same as the language. So, when n is 0, it is the empty string that n is 1 it is 01, then 0011, 000111 and so on. So, this is the language.

So, we go by what we said already, we first assume that this language is regular. Now there is a pumping length p. So, now let us choose a string that is in the language and is of length at least the pumping length. So, here I am going to choose 0 power p, 1 power p, which is basically p 0's followed by p 1's.

So, this is clearly in the language. So, the language is any number of 0's followed by the same number of 1s, so the language is an infinite language. So, you could have two 0's followed by two 1's, three 0's, followed by three 1's, four 0's followed by four 1's, five 0's and, and so on.

So, it is an infinite language so like that, whatever value p is, we have the string p 0's followed by p 1's. So, this string is clearly in the language. And it has a length of 2p. So, there is p 0's followed by p 1's. So, it is of length of 2p, it satisfies both the conditions. Now, the pumping lemma says that since the string is in the language, and since we assume that the language is regular, we can split this into xyz satisfying the three conditions.

Now, we will show that it is not possible so, let us see y. So, what is S, S is all 0's, p 0's followed by p 1's. This is S, suppose y, so now, if we split into x,y,z, so y could that is x followed by y followed by z. So, y could be entirely in the zero part. If y is entirely in the zero part, let us say y is somewhere here.

Then the problem is that condition two says that y is not an empty string, so which means y has at least one zero. So, if you consider x y square z, let us say I call S’ as x y^2 z. If you consider x y^2 z or in other words xyyz. So, this has one more y than what has had which means S’ has more 0’s than what is?

Because y is entirely 0’s. So, when you add one more copy of y, the resulting string has more 0’s than 1’s, but we know all the strings in the language B have the equal number of 0’s and 1’s. So, this S’ is not in the language. So, this is not in the language. The same thing holds when y is just entirely in the 1’s, so suppose y was entirely in once somewhere over here, then not the same thing.

But similar thing because we again consider S’ x yy z again because of condition 2y cannot be empty. So, y has to have at least one symbol and by assumption y is only by the assumption of this case y is only once. So, xyyz, we will have more 1’s than 0’s because xyz, xyz had the equal number of 0’s and 1’s when you add one more copy of y you will end up having more 1’s than 0’s.

So, xyz has the same number of 0’s and 1’s and y has the same number once and when you add one more copy of y you end up having more 1’s than 0’s. So, this is what happens when y has only 1’s, so, even this string is not in the language. Because y has only 1’s also is not in the language.

Now, consider the third case. So, the three possibilities are y only 0’s, y is only 1’s and the third one is that y contains both 0’s and 1’s. So, just to denote in the figure above so maybe y transcends the boundary of 0’s and 1’s.

So, y , it comes something like this. So, y is some 0’s followed by some 1’s. So, something like this that I have written here. So, now consider S’ which is again xyyz so, if y is transcending the boundary x has to be entirely 0’s and z also has to be entirely 1’s, y is crossing the boundary. So, x has to be entirely 0’s, and z has to be entirely 1’s. So, now, let us see what shape xyyz has?

So, it will have x which is entirely 0's, y which is 0's and 1's some 0's followed by some 1's. Then again another copy of y which is some 0's, followed by some 1's followed by z, which is entirely 1's. So, basically, you start with 0's x and the initial part of the first y then you move to 1's then again you move to 0 this initial part of the second y then you have entirely 1's.

So, it is like some 0's, some 1's some 0's again followed by 1 still there. So, this is some 0's, some 1's, some 0's, some 1's. So, this is not so if you look at the language B it is all some 0's followed by same number of 1's, this so, in other words, B is a subset of 0 star 1 star, but this string here xyyz this string over here that I have this is not of the form $0^* 1^*$. Hence, this S prime is also not a part of B.

So, this means we took a string S which is 0^p, 1 power p we wanted to show that it cannot be split in the as xyz meeting the three conditions. So, there are three possibilities for y, y is entirely 0's entirely 1's and then a mix of 0's and 1's. What we showed is that in each of these three cases there is some contradiction that we get and xyyz is not in the language B.

This means that there is no way to split xyz sorry, there is no way to split S xyz meeting the three conditions or in sometimes informally we referred to as there is no way to pump the string. So, pumping means we cannot split it in as xyz meeting these three conditions. So, this string there is no way to split and hence B is not regular, if B was regular all strings that are in B and that are of length greater than p can be split, however, this string cannot B, hence B is not regular.

So, we assumed that B is regular then we assumed the pumping length then we chose again this is entirely our choice 1's strings 0^p 1^p and showed that there is no possibility of splitting it as xyz meets the conditions. So, this is the usual template of what we follow when we want to show that a language is not regular.

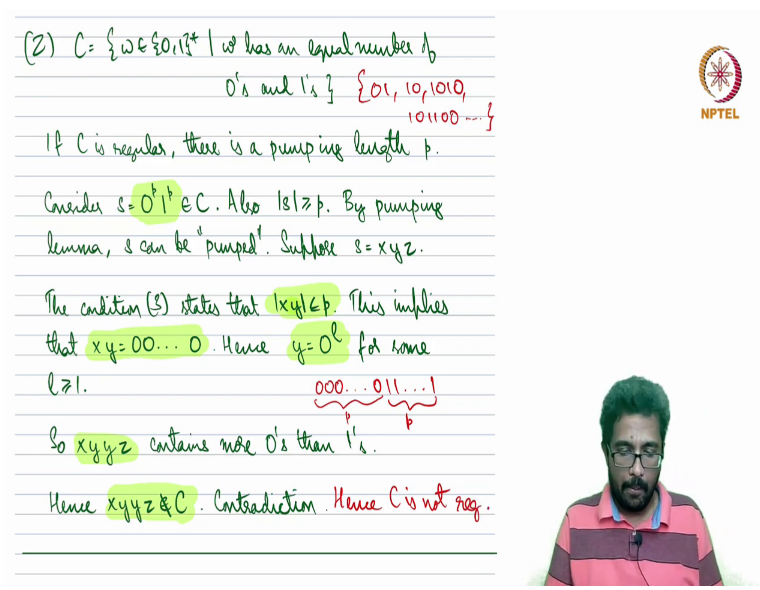
So, notice one thing here we use condition 1, like because condition 1 says that x y^i z, it has to be in the language. So, x y^i z, so we use that in all the three cases. We also use even though I did not make it very explicit, we also use condition 2, because condition 2 says that y is not empty. If y were like we are using that to say in this case, in the first case, we are using that to reason that y has at least one 0.

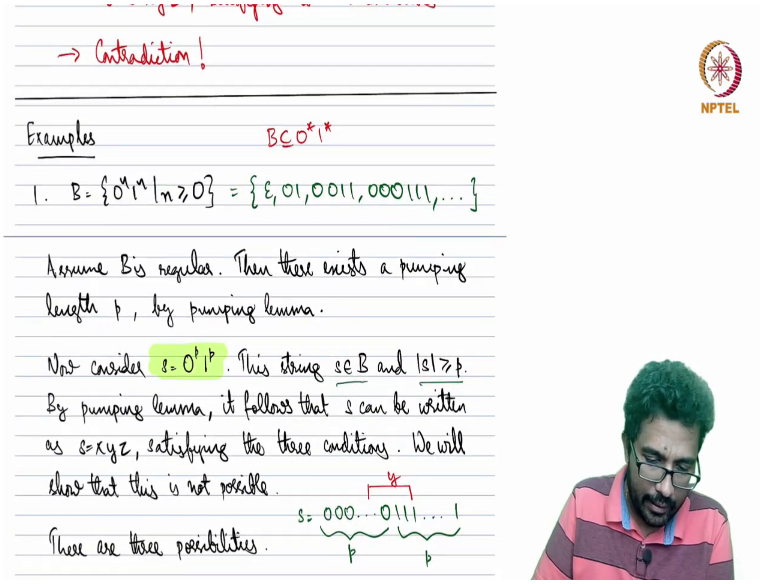
Because if y is empty string that does not make sense. So, we are using condition 2 that y is non empty in all these three reasoning. However, we are not using the third condition, the third condition says that the length of xy is at most p, if we had used the third condition, so, if you had used the third condition, the length of xy is entirely p and we know that x y is the initial part of the three splits.

So, xyz, so z is the last part x and then y and then z. So, if x y is entirely p, the first p symbols. So, in fact, x y is at most the length of x y is at most p. So, it not even B p's symbols, it could be less than p symbols. This means that xy is entirely contained in the 0 part. This means that y is also entirely contained in the 0 part.

And this means that we necessarily have to be in case 1, the case 2 or 3, do not even are not possible because of condition 3, but then if you come to case 1, then again, like we reason above xyyz has more 0's than 1's and sends, it is not the language. So, this is another way to like if we use condition 3, we get a different chain of reasoning, but still we end up with the same inference that we cannot split y as sorry we cannot split x S S xyz satisfying the three conditions.

(Refer Slide Time: 17:08)





Now, let us consider another language C, C is a set of all strings, binary strings, that has an equal number of 0's and 1's. So, B also had a set of binary strings that have equal number of 0's and 1's, but B required the structure we have all the 0's and then all the 1's C does not require all the structures, so it could be any string. So, for instance, C could contain 01, 10, 1010 like 1011001 like anything that contains sorry, not the last one anything that contains equal number of 0's and 1's.

It could be mixed in any order, any sequence. It also the empty string has 0 0's and 0 1's. Now, let us try to show that this language is not regular. So, it is the usual thing that we already saw, we assume it is regular then that is a pumping length, and then we have to choose a string. Here again the same string is good enough 0 power p, 1 power p. So, pump that p is a pumping length.

Clearly this has equal number of 0's and 1's p 0's is followed by p 1's. So, hence, it is in the language C also it is of length 2p which is at least p. Hence this has to be this can be pumped meaning this can be split as xyz meeting the three conditions. Now, let us look at the string. So, this string is p 0's followed by p 1's. Condition 3 of the pumping lemma says that x y is x y is of length at most p.

So, this is what condition 3 says x y is of length at most p this implies that since xy is of length at most p this implies that x and y are entirely in the initial p symbols which are entirely 0's. Hence, it follows that x y is entirely 0's.

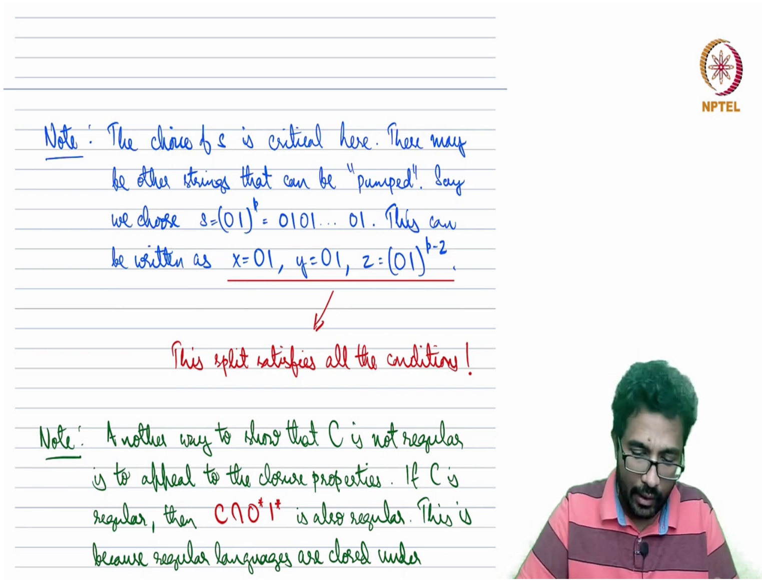
So, this the number of 0's, is could be p it could be less than p. Since x y is entirely 0's y is also entirely 0's. Let y be like some number of 0's let us says l 0's. And by condition 2 we know that l is at least 1, y cannot be an empty string. So, condition 2 implies that l is at least 1.

Hence we know that y is some string of 0's of at least one 0's. So, now consider the string be used condition 1 which is x y power i z. So, now consider x y square z. So, xyyz this contains, so, xyz contains equal number of 0's and 1's we add another y which is entirely 0's. Now, which means this will contain l more 0's than 1's like how many ever 0's y contain that many more 0's than 1's.

So, this is not part of the language, but condition 1 says that xyyz should be part of the language. So, this is a contradiction. Once again we choose a string 0 power p, 1 power p which is in the language and is of length greater than the pumping length. Condition 3 states that y condition 3 implies that y is entirely 0's condition 2 implies that y is not empty.

So, y has at least one 0, could have more 0's and does not have any 1's. So, now xyyz as per condition 1, should be in the language, but xyyz contains l more 0's than 1's, which is a string not in the language. So, we picked one string and showed that it cannot be split. This is the only way to split it meeting these conditions, but then again it fails in condition 1, if you try to forcefully satisfy condition 3 and 2. Hence, this string is not in the language which is a contradiction to the assumption that C was regular. Hence, C is not regular.

(Refer Slide Time: 22:04)



Couple of points here the choice of the string is very important. So, as I said earlier, this is usually the most creative or when I say creative, it is not like an extreme level of creativity or anything it is usually the more the thing that you need to like. Maybe think about a couple of options and finally, we will figure out something. But this is very important as well, because if S is regular, all the strings can be pumped; all the strings that meet the requirement that SS in the language and SS of length at least p can be pumped.

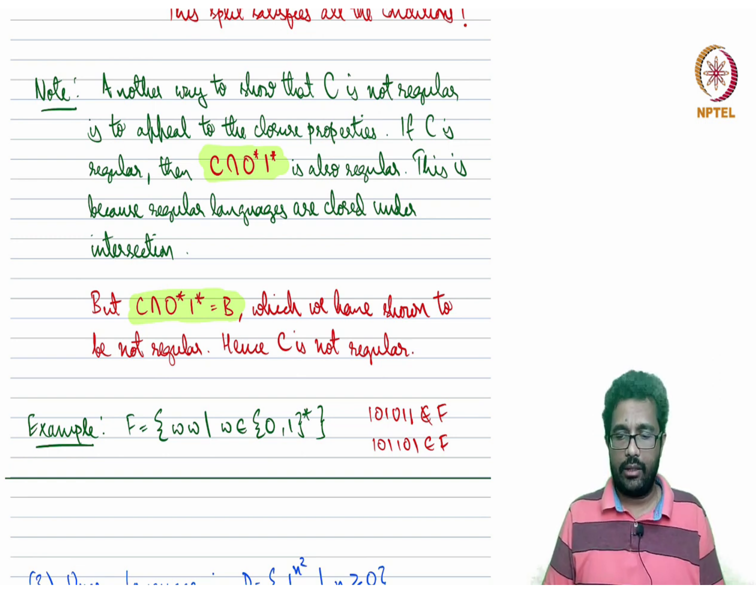
Even in this case, there could be other strings that could be pumped. So, for instance, if you consider 01 power p, that 01 repeats, p times 01 01 01 01 like that p times. However, if you take x as 01 y as 01 and z as the rest of the string, this split satisfies all the conditions. Assuming p is like something like 10 or more than 10 or something anything more than 5 will work I think, this satisfies all the conditions if you see y is not empty, if you see x y is of length 4 which is of length less than assuming p is at least 10 it is fine.

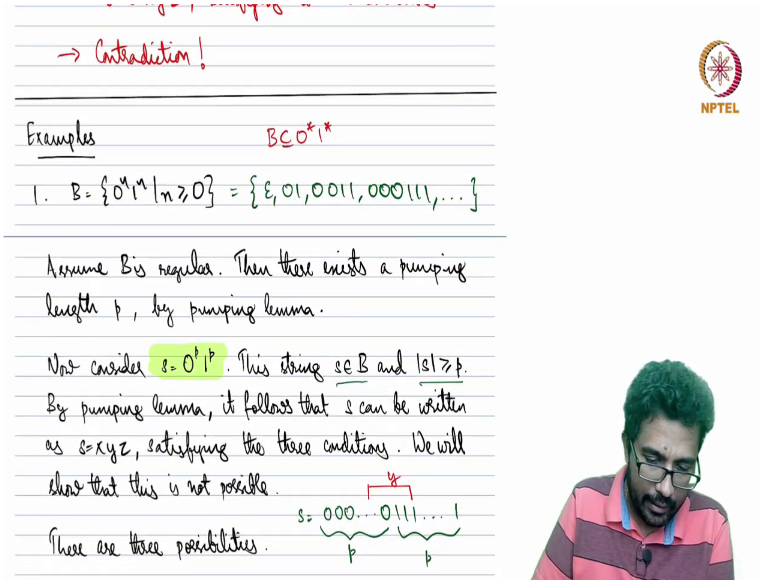
And if you take x y^i z for any number of i's, all of these have equal number of 0's and 1's because x has equal number of 0's and 1's y has equal number of 0's and z also has. So, even if you repeat y whatever many times the number of 0's and 1's equally get repeated. So, this meets all the conditions, but we cannot use this to infer that the language is regular because if the language is regular all strings can be pumped all the strings that are in the language and that are of length at least p.

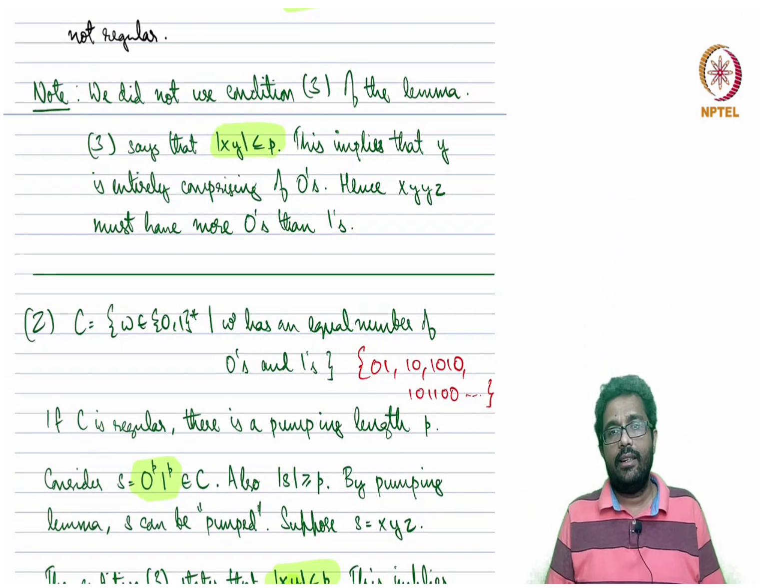
Just by showing one string that can be pumped it is not good enough. To show that language is so, in fact, we cannot show that language is regular using pumping lemma to show that the language is not regular . We just need to demonstrate one string that cannot be pumped. However, to show that the language is regular it is not enough to show that some string can be pumped.

So, to show that the language is not regular you have to carefully choose a string. It is not like you choose whatever string that meets the conditions and then you can get a contradiction. So, here is an example: you have to carefully choose a string because if you choose a wrong string that perhaps it can be pumped or split in the manner satisfying the three conditions. So, this is very important and I hope you remember it when you are trying to work out exercises.

(Refer Slide Time: 24:56)







Another point is we first showed that B which is 0^p ,1^p is not regular. And I observed that B is a subset of a sub language of secret C has all the strings that have an equal number of 0's and 1's but B has all of them that take a particular structure 0 power p, 0 power n, 1 power n.

So, B is a subset of C. Here we can show that C is not regular. The other way is the following. If C was regular, then C intersection 0 star 1 star is also regular. This is because regular languages are closed or intersection. So, if C is regular, we know by assumption C is a regular 0 star 1 star is clearly regular because it is a regular expression, $0^* 1^*$ means the set of all strings that start with 0's, some 0's.

And then followed by some 1's, some 0's followed by some 1's. So, this including 00 are possible, some zero 1's are possible, it could be some 0's alone, it could be some alone, it could be some 0's followed by someone, but it will not have some 1's fall for some 1's and then some 0's. This is a regular expression, and hence it is regular. So, the intersection is also regular, because regular languages are closed under intersection. But what is C intersection this? In other words, the set of C is a set of all strings that have equal number of 0's and 1's.

And out of these strings, which are the 1's that take the 0 star 1 star structure, if you think about it, you will see that the only way this can happen is if you have some number of 0's followed by the same number of 1's, which is exactly what we define B to B. So, $C \cap 0^* 1^*$, this is a regular language. And this happens to be B. So, we have said that it is regular, and now we are observing that it is equal to B, but we have already shown that B is not regular. So, if C was regular, this intersection is regular. But this intersection is equal to B, which we have shown to be not regular.

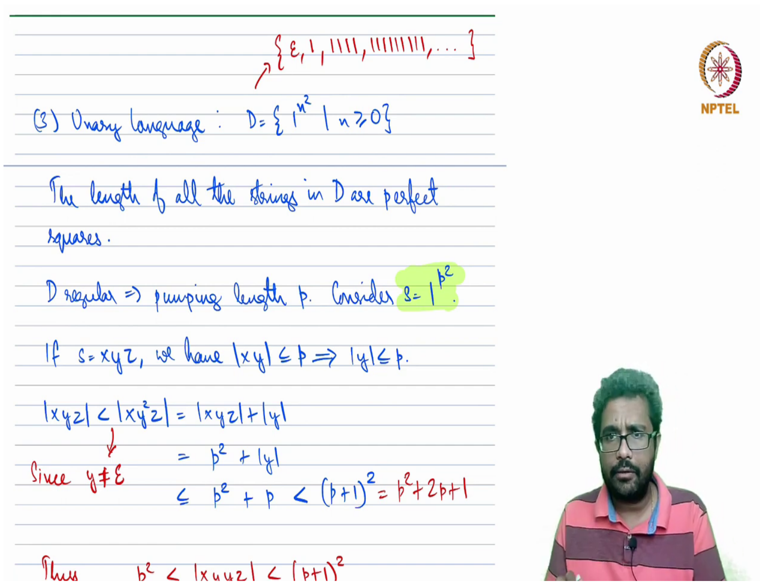
So, this implies that so we started with the assumption that C is regular, and we ended up with the inference that B is regular, which we know is a contradiction, because we have already shown B to B not regular. So, this implies that C is not regular. So, this is a way to show that C is not regular, without directly going through the pumping lemma. So, here we are just using closure properties, we are just using the 0 star 1 star to say I said is a regular expression, regular and 2 we are using the regular languages are closed under intersection.

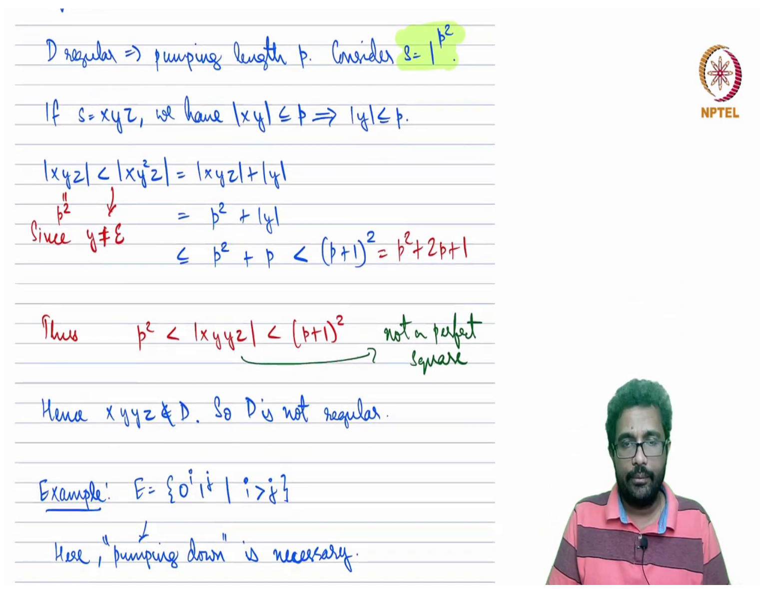
And 3, we are using the B is not regular, so these are the three facts that we are using. So, maybe I will just list them down here so three facts that we are using, maybe I like it here are 1 maybe a lot list them down because space it is constrained. The first is that 0 star 1 star is regular second is that regular languages are closed under intersection and third is that B is not regular. So, this three together imply that C is not regular without directly applying pumping lemma on C.

So, this is just one example, that is there in the book as well, but you can try it out. So, set of all strings, which are of the form w w. So, basically you should be able to write it as a repeat of two strings. So, 10101 this is not of that form. First of all, if it is repeated, we will have even even this is not of that form, but 101101.

So, this is not in F whereas, this will be in F. So, set of all strings that are of the form w w, where w is some string, so, basically some string repeated two times. So, this language is not regular. So, this is just I am just leaving it as an exercise, I think it is there in the book, otherwise you can try to work it out it is pretty much similar to what we have been doing before.

(Refer Slide Time: 29:56)





So, we took C and now we are see one more language which is a unary language. So, unary language means all the strings in the language usually one symbol in this case it is the symbol 1. So, what is a language D is 1 power n squared for all n.

So, let us try to write down what this languages where n is 0, this is an empty string, when n is 1 this is just 1, n is 2, it is like 4, where n is 3 it is 9 and so on. So, basically it is set of all strings of 1's whose length so entirely of 1's whose length is a perfect square, so, 1 4 9 16 and so on. So, it is again an infinite set of strings set of all strings whose length is a perfect square.

Now, the standard template applies we assume D is regular which implies there is a pumping length. And now we consider the following string S which is of length p square, so the length all the strings are comprising of 1's.

So, the only once I tell you the length, then you know the string, because when I say the length is p square, it just means that the string is entirely 1's, string is entirely 1's and string is p square 1's. So, consider the string and will show that it cannot be pumped.

So, if x suppose we can pump it which means if S can be split as xyz, then by condition 3, we have the length of x y is at most p, which means the length of y is also at most p because x y is the superstring of y. So, like the y is also at most p. So, it follows that now, the length of xyz, xyz, the length of xyz is the length of S, which we know is p square.

This is strictly less than the length of x y^2 z, y is that, because y is not empty. So, when you add one more y, you are adding something to the string, so, it adds length. So, condition 2 say is that y is non empty. Now, what is the length of x y^2 z, what is the length of x y^2 z?

The length of x y^2 z is the length of xyz and the added string y, the length of the added string y, length of xyz is the name same as the length of original string S, because that is what xyz is which is p square, and then you add the y. But we know that we just said that x y is at most p hence y is at most p.

So, this is at most p square plus p. So, the length of x y square z is at most p square plus p. But this is strictly less than p plus 1 whole squared. What is p plus 1 whole squared? It is p square plus 2p plus 1. So, p^2 + p is strictly less than this. So, we did some bunch of calculations. So, what does it amount to?

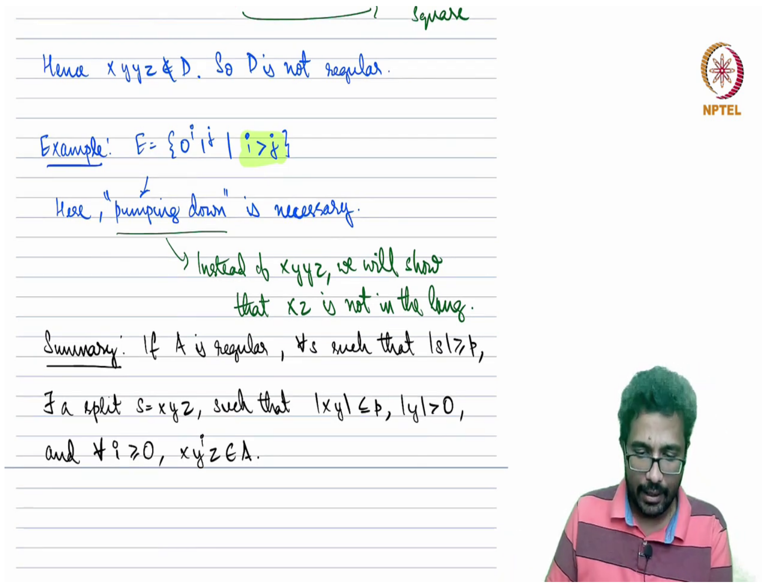
So, we started by saying that the length of x y square z is strictly greater than that of xyz, and the length of xyz is p square. So, the length of xyyz is strictly greater than p square. And we ended by saying that it is strictly less than p plus 1 whole squared. So, the length of xyyz is strictly greater than p square and strictly less than $(p + 1)^2$.

So, it is in between two perfect squares, p^2 and $(p + 1)^2$. And we know that after p^2, the next square is $(p + 1)^2$. So, it is lying between two perfect squares. So, we know that this implies that xyyz, maybe I will just use a different colour here. This is not a length is not a perfect square.

Because it is neither p square not p plus 1 square it is something in between. Hence, xyyz is not in the language because the language consists of all the strings whose length are perfect squares. Hence, this is not regular. So, again, we just to quickly recap, we assume that it is regular, so, there is a pumping length.

Now, we consider the string 1 power p square, which means length p square. Now, by condition 3, we knew that y is of length at most p. Condition 2 tells us that y is not empty. So, y is something between 1 and p, the length of y and then we reasoned to show that xyyz is of length strictly greater than p square and strictly less than $(p + 1)^2$. Hence, the length of xyyz is not a perfect square and hence, it is not regular.

(Refer Slide Time: 35:55)



Another example or exercise for you. Consider all the strings 0 power i, 1 power j, some 0's followed by some 1's where the number of 0's is strictly greater than the number of 1's. So, this is strictly greater. So, this is not the B, B had the number of 0's and number of 1's were equal here we are saying it is strictly greater, this also is not regular the one novelty that you will find when you are trying to do this is that in this case, you will have to pump down.

Meaning pump down means in all these cases here we have considered xyyz and showed that that is not in the language, pumping down I mean, we will have to consider instead of xyyz we will show that xz is not in the language. So, xz is also x y power i z where i is equal to 0. So, instead of pumping up meaning adding more copies of y we are pumping down we are removing the existing copy of y. So, this is y this example will be interesting. You can try it out.

(Refer Slide Time: 37:29)

Example: ...

Here, "pumping down" is necessary.

Instead of $xyyz$, we will show that xz is not in the lang.

Summary: If A is regular, $\forall s \in A$ such that $|s| \geq p$,

$\exists$ a split $s = xyz$, such that $|xy| \leq p$, $|y| > 0$,

and $\forall i \geq 0$, $xy^iz \in A$.

We use the following steps to argue that a language is not regular.

→ Assume that the language is regular.

→ By Pumping lemma, there exists a pumping length p.

→ Choose a string $s \in$ language s.t. $|s| \geq p$.
(Usually the creative step)

→ Show that this string s cannot be split as $s = xyz$, satisfying the three conditions.

→ Contradiction!

Examples $\quad B \subseteq 0^*1^*$

1. $B = \{0^n 1^n \mid n \geq 0\} = \{\varepsilon, 01, 0011, 000111, \ldots\}$

Assume B is regular. Then there exists a pumping length p, by pumping lemma.

Now consider $s = 0^p 1^p$. This string $s \in B$ and $|s| \geq p$. By pumping lemma, it follows that s can be written as $s = xyz$, satisfying the three conditions. We will show that this is not possible.

$$s = \underbrace{000\ldots0}_{p}\underbrace{111\ldots1}_{p}$$

There are three possibilities.

(1) y has only 0's. $\Rightarrow s' = xy^2z = xyyz \notin B$.
has more 0's than 1's.

(2) y has only 1's $\Rightarrow s' = xyyz$ has more

is entirely comprising of 0's. Hence $xyyz$ must have more 0's than 1's.

---

(2) $C = \{ w \in \{0,1\}^* \mid w$ has an equal number of 0's and 1's $\}$ $\{01, 10, 1010, 101100 \ldots\}$

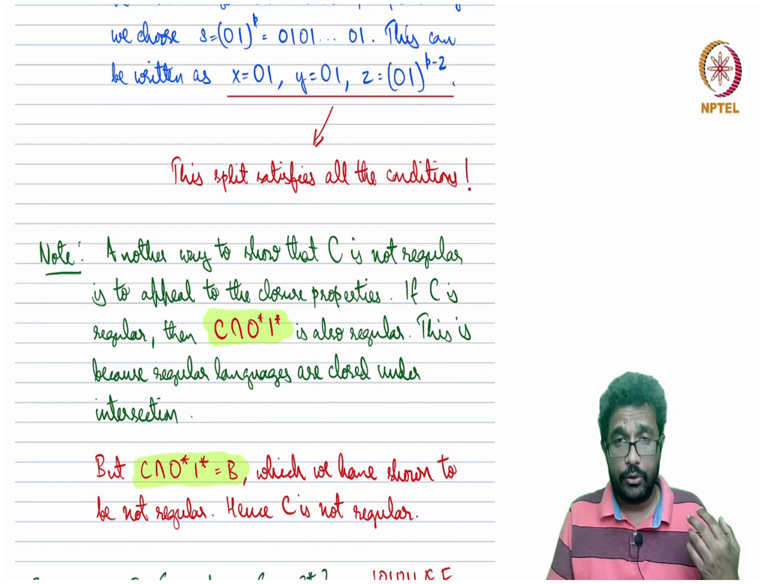
If C is regular, there is a pumping length p.

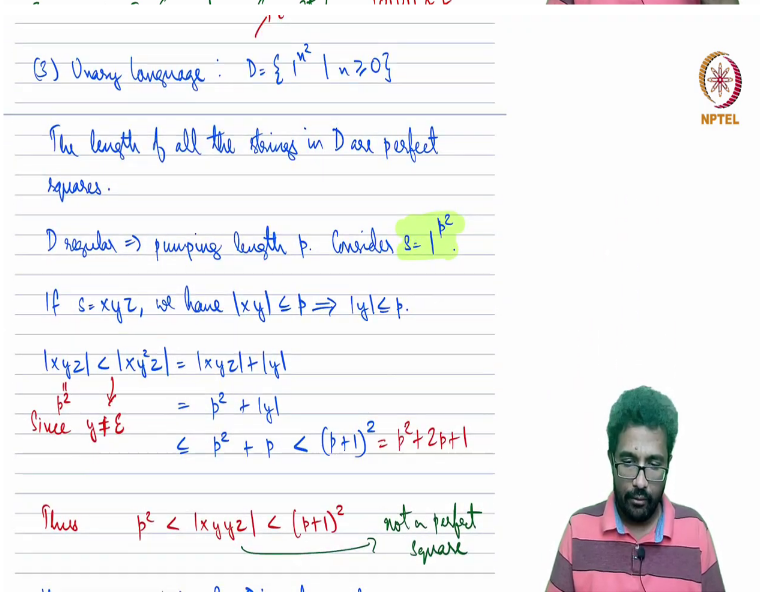
Consider $s = 0^p 1^p \in C$. Also $|s| \geq p$. By pumping lemma, s can be "pumped". Suppose $s = xyz$.

The condition (3) states that $|xy| \leq p$. This implies that $xy = 00\ldots0$. Hence $y = 0^l$ for some $l \geq 1$.

$\underbrace{000\ldots0}_{p}\underbrace{11\ldots1}_{p}$

So $xyyz$ contains more 0's than 1's.





So, finally, I think we have seen three examples just to summarize, maybe I will just bring it a bit below just to summarize. What is pumping lemma? Pumping lemma says that if A is regular, then for all S maybe I will edit this is a bit for all S in A such that the length of S is at least p there is a way to split S such that x y is at most p, y is greater than 0, and for all I, x y^i z is an A. So, this is the pumping them again so, I am just reiterating it.

So, it has the following form. So, there is a for all here. So, for all the S so, if A is regular for all the S this has to be satisfied then there exists a split there has to be some way to split it. So, there could be multiple ways of splitting it, but all we require is some way to split it such that again there is a, for all.

So, to show that a language is not regular we kind of we have to negate this. So, we have to show that it is, there exists a string, so this string is entirely our choice. So, instead of for all S to refute for all S we have to show there exists a string and for that string for all splits this is not true. So, for that split sorry for that string, we have to refute all possible splits such that these two conditions and we just have to show one i for which this is not satisfied.

So, we in all the examples that we worked out the i was 2, so we said that xyyz is not in the language, but then there could be other cases 2. So, this can be thought of as a game, you tell me the language that you want to show it is not regular i will pick the string. Now, you tell me whichever split possible then I will show you where that split falters.

So, will I have to show that it is my duty to pick the string, but whatever split you come up with it will not be possible it will not meet the condition. So, whatever you come up with, I can find some issue with that split. So, just summarizing, we showed how to we demonstrated how to use pumping lemma.

So, this is the standard template we assume the language is regular, then we choose a string that is of, that is in the language and that is of length greater than or equal to p and show that this string cannot be split, again and again repeating this. So, the choice of string is entirely ours we can choose any string that is in the language, but then we have to argue a reason that whatever way to split it, it cannot satisfy all the three conditions.

So, there is no way to split it as xyz satisfying the three conditions and that yields a contradiction. So, then we saw the three languages. First is 0 power n, 1 power n and then is a set of all languages, all strings that have equal number of 0's and 1's, which is a superset of the previous language. Then I said that you can also use, you can also use closure properties to show that the second language is not regular, by just appealing to the fact that B is not regular.

And by closure properties, you can show that C is not regular, you do not have to go through pumping lemma, even though we also saw the pumping lemma proof. And finally, we saw the unary language proof where we considered a unary language. And, I think that is about it for this lecture, lecture 13. See you in the next lecture. Thank you.