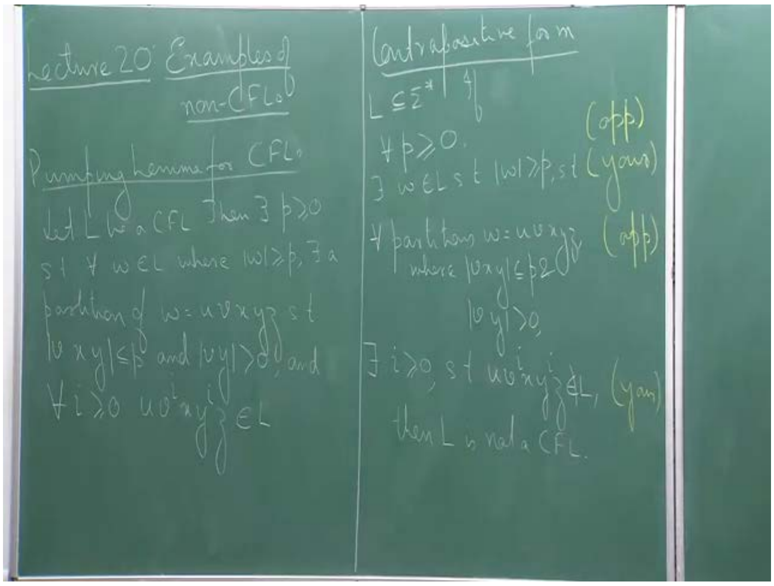
Theory of Computation
Prof. Raghunath Tewari
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture – 20
Examples of non-CFLs

Welcome to the 20 lecture of this course. In our last lecture, we saw the pumping lemma of context free languages, and we also saw a group of the pumping lemma, why the lemma is correct. So, the proof basically involved taking a context free grammar for the pumping lemma, and for every string that is there in the language that has length larger than a certain constant, where the constant dependent on what our grammar was.

We showed that the string can be partitioned into a sequence of 5 strings, such that they sets 5 certain property; and then if we construct some other strings that are based on these 5 strings, each one of them belongs to the language.

(Refer Slide Time: 01:13)



Let me first write down what the pumping lemma was just to recall. Let L be a context free languages then there exits p greater than 0 such that for all w belonging to L with where length of w is greater than or equal to p, there exit a partition of w equal to u v x y z such that the middle portion that is v x y is at most p and v y is strictly greater than 0; and for all i greater than or equal to 0, u v to the power of i x y to the power i z also belongs to the language. So, to show that a language is not context free, we will use the contrapositive form of this theorem.

And once again like in the case of regular languages, we will treat the contrapositive form as a game between you and an opponent, where your objective is to prove that the language is not context free, and your opponent object is to defeat you. So, if you are able to win then you end up showing that the language is not context free; if you do not win, if your opponent wins, it does not show anything, but it does not show that the language is not context free.

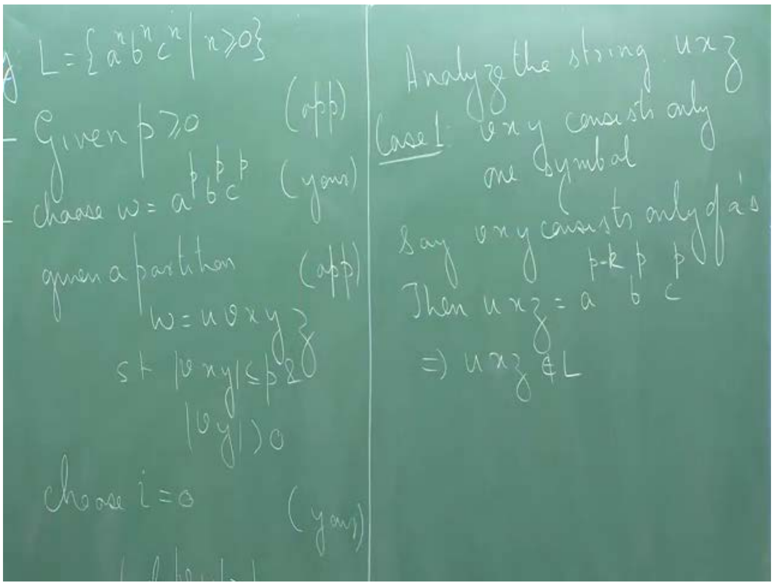
So, one important point that I want to make in this context and also in the context of regular language is that, so suppose if a language is not context free for example, if a language is not context free even then it can satisfy the conditions of the pumping lemma. So, please keep that in mind. In other word, if the conditions of the pumping lemma are not satisfied, it can still be the case that I mean if the conditions of the pumping lemma are not satisfied it can still be the case that the language is, it is not a context free language.

Lets look at the contrapositive form. So, L is some language if for all so once again if for all p greater than 0. So, because this move is quantified by the for all quantifier therefore, this will be a move of your opponent. So, you do not have any control over this move; your opponent is giving you a number p. So, this is your opponents move, there exits w in L such that length of the w is at least p. So, here this is the there exits move. So, you can chose, you can chose any w in the language L whose length is more than p. This is your move. Such that for all partitions w equal to u v x y z, where v x y is at most p, and v y is greater than 0.

Now your opponent, once again this a your for all move. So, your opponent is partitioning the string. So, whatever w you pick, your opponent can partition into any set of five strings u, v, x, y, z, which satisfy these two conditions. So, once again you do not have any control over what the partitions will be, they can be any thing, but you can be assured that whatever the strings are they must satisfy these two conditions. For all partitions, there exits i greater than or equal 0, such that u v to the power i x y to the power i z does not belong to L.

Now given this partition, you can choose an i that you like greater than or equal to 0 such that this string does not belong to L. So, once again this is your move. So, if you are able to show this then, L is not a context free language. So, what we will show today is we will show how to use this pumping lemma to show languages are not context free.

(Refer Slide Time: 08:23)



So, our first example will be the language L, which consists of strings of the form a to the power n b to the power n c to the power n; for n greater than or equal to 0. So, the string is over a three alphabets I mean over the alphabets has three symbols a, b and c. And we look at all strings which have an equal number of a(s) followed by equal number of b(s) followed by an equal number of c(s). So, one thing that I mentioned when we where talking about not regular languages is that a regular languages do not have the capability to count something, and we said that in the case of context free language, for example, we can accept the language a to the power n b to the power n.

And the reason why that is happened is because we could give a grammar for it; and apparently, it seem that in the case of context free languages, we can do some counting, but that is not very precisely true. So, even in the case of context free languages we cannot count and remember a certain quantity, because if we could count how many a(s) are there then I would remember what that number is. And then when I count the number of b(s), I will check whether those two numbers are equal. And when I get the number of c(s), again I check whether those two numbers are equal and that would end up showing that this languages context free, but that is not quite what happens.

So, in the case of context free languages, in some sense, we able to compare two quantities at any given point of time during our computation, we can compare two quantities whether they are same, whether one is more over that, one is less or whatever, but the moment we need to compare more than two quantities at a give point, context free languages do not work. So, this an intuitive way of understanding the power of context free languages. So, but for the sake of proof, we have to use pumping lemma.

Now let us look at this. So, first your opponent make some move. So, your opponent gives you a number p greater than or equal to 0. So, this is the move that your opponent makes. Now you have to choose a string that is in the language whose length is more than p. So, what string do we choose. So, we are going to choose w, which has p number of a(s) followed by p number of b(s) followed by p number of c(s); clearly this belongs to the language and has length more than p.

Now, given a partition w equals u v x y z, what can I say about v x y. So, what are the different cases that can arise. So, essentially proving a language is not context free using the pumping lemma, builddown to tackling all possible different cases that can arise, and how intelligently you can form the cases, so that you can handle all possibility by considering the minimum number of cases.

So, here let us considered the following case. So, what we will do is, so first let me show first let me tell you what i we are going to choose. So, this is your opponent moves, this was your move. Now your opponent chooses this; such that v x y is at most p and v y is greater than 0. So, what we will do is we will choose i equal to 0. And let us try to analyze the string. So, we will analyze the string u v to the power 0 x y to the power 0 z.

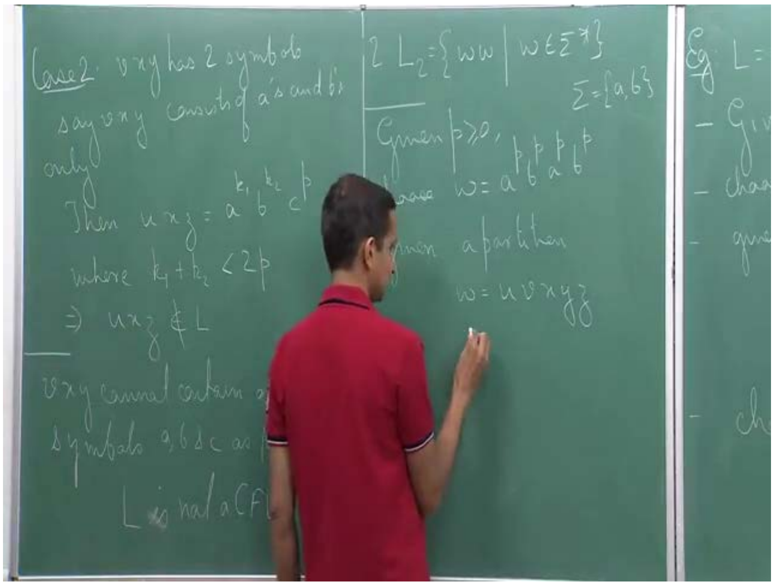
So, another way of writing that is u x z, because v and y are absent.

So, what are the cases that can happen? So case 1 is when v x y consists only of one symbol. So, observe that we do not have any control over what u and z are going to be. So, what if v x y consists only of a(s) or v x y consists only of b(s) or v x y consists only of c(s). So, any three of the following can happen actually. So, if or in v x y can consists only of c(s) also in which case u is basically has all a(s) and all b(s) may be some c(s).

So, case 1 is when v x y consists only of only one symbol. So, if v x y consists only one symbol what happens, if I remove v and y from that string. So, what we know is that the length of v y is greater than 0. Let may be I can just write, so this was your move. Let v y have length equal to some k, where k is a number that is grater than 0. So, say v x y is equal to a to the power l or say v x y consists only of a(s). In this case, what happens to this string? So, in the string u x z, I do not have v y now, which means I do not have k many a(s). So, this implies that u x z is what.

Initially, it had p number of a(s), but now i have removed k. So, it is a to the power p minus k; and b(s) and cs remain the same. So, b to the power k, c to the power k, I am sorry b to the power p and c to the power p, and because k is greater than 0 strictly than 0 which implies that u x z is not in the language L, because it does not have the form equal number of a(s), b(s) and c(s). So, this is the case when we have only one symbol.

(Refer Slide Time: 17:32)



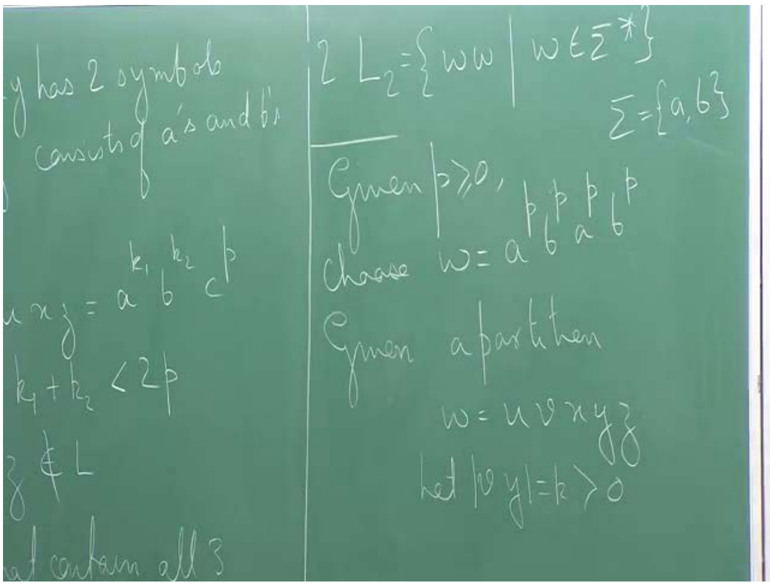
Now, considered the case when v x y has 2 symbols. So, before going into this aspect, let me just point out one more thing. So, here I assume that v x y consists of only of a(s), but if i had taken v x y to consists only of b(s) or consists only of c(s), the argument will be symmetric I mean instead of a(s), I will have lesser number of b(s) or lesser number of c(s). So, suppose v x y has a 2 symbols, and let v x y be some a to the power maybe say v x y consists of a(s) and b(s) only. Observe that it cannot contain all 3 symbol a, b and c; the reason being that the length of v x y is at most p, so and because each of these blocks a(s), b(s) and c(s) have p symbols, therefore v x y cannot contain all three of them, so it can contain two of them.

So, it does not contain c(s). So, then u x z has a form some a to the power k 1 b to the power k 2 c to the power p, where k 1 plus k 2 is strictly less than 2 p, which implies that u x z is not in the language L. Because once again v and y have length either of them I mean at least one of them have length strictly greater than 0. Therefore, either I have lesser number of a(s) than p, or I have lesser number of b(s) than p or maybe both, and which implies that k 1 plus k 2 is strictly less than 2 p which implies that the language cannot be in L. And once again the same argument will happen if instead of a(s) and b(s), v x y consist of b(s) and c(s).

And the third thing is that v x y cannot contain all three symbols a, b and c has its length is at most p. So, this proves that L is not a context free language. So, this is essentially the process to show a language is not context free using the pumping lemma. Let us look at one more example. So, consider the language L, let call it L 2, which has strings of the font w w, where w is any string in sigma star, you take any string in sigma star. And now look at all strings of the form which have a string followed by a reputation of the same string. So, the exact same string repeats.

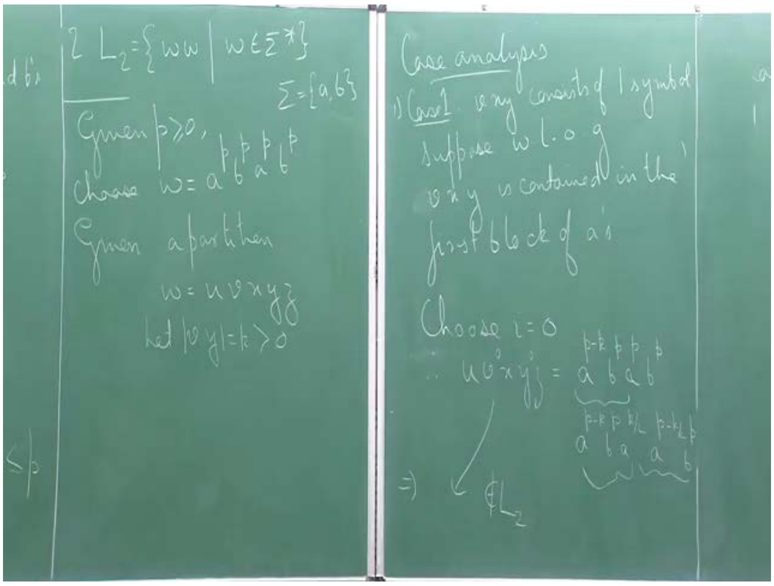
So, once again will prove that this language is not context free. So, given I mean sigma can be anything here, but for just for the sake of simplicity, we will prove this when sigma is consists of the symbol a b. Suppose sigma has two symbols a and b. So, given p greater or equal to 0, I will choose a string w. So, do not confuse this w with this w; this is for the only for the definition of this language. So, I want to choose a string whose length is more than p and it belongs to L 2. So, I will choose a to the power p b 2 the p a to the power p b to the power p. So, if i look at the first part of the string in a second part of the string, the first half and the second half, so they are the same. So, I have a to the power p b to the power p on both the parts. Now given a partition w equals u v x y z, what are the things that can happen.

(Refer Slide Time: 23:59)



Suppose let v y be equal to k we know that k is strictly greater than 0.

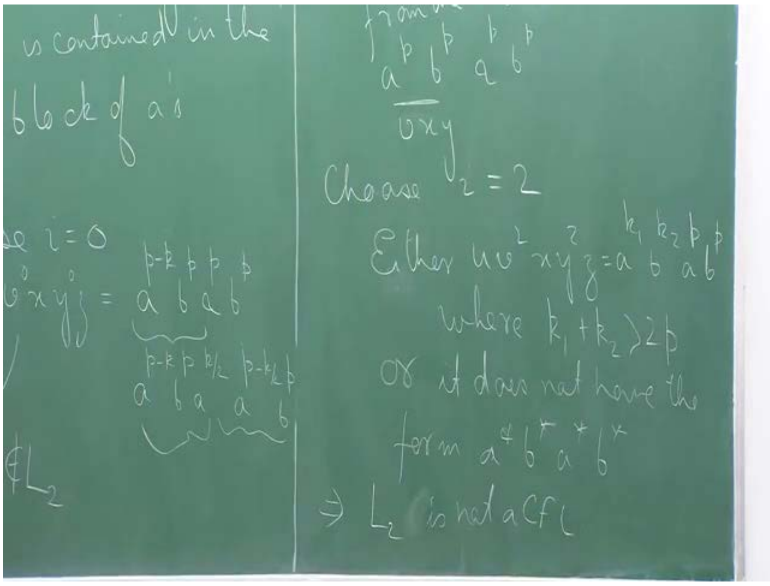
(Refer Slide Time: 24:17)



So, once again let us do a case analysis. So, firstly, v x y consists of one symbol. If v x y consist of one symbol, it either lies entirely within the first block of a(s) or within the first block of b(s) or it lies entirely within the second block of a(s) or the second block of b(s) nothing else. So, suppose without loss of generality, v x y is contained in the first block of a(s). So, what we will do in this case is that given a partition, where we have v x y which belongs to the first block of a(s), it has some number of a(s), we will choose i equal to 0. Therefore, u v to the power 0 x y to the power 0 z will be what will be the string. So, v and y together had a length k, now I have k many s less in the first block. So, it will be a to the power p minus k b to the power p a to the power p b to the power p.

Now if I divide this string into two parts, if I devided into 2 halfs, so the first half will have a to the power p minus k b to the power p a to the power k by 2. So, this total thing has length 2 p plus sorry 2 p minus k by 2; and the second part will have a to the power p minus k by 2 b to the power p; again this has 2 p minus k by 2. So, this is the first half; this is the second half; clearly they are not equal because k is greater than 0. So, this implies that this string is not in the language L 2.

(Refer Slide Time: 27:55)



Now, once again what if I look at case 2, where v x y consists of 2 symbol, and let v x y contain a(s) and b(s) from the first 2 blocks. So, basically, I have a to the power p b to the power p c to the a to the power b and b to the p. So, suppose v x y kind of comes from this region. So v x y comes from this region, now what I will choose is that I will choose i equal to 2. And what that would give me is that, that will give me a to the power now v and y can be anything I mean v and y can lie entirely within a(s), v and y can lie entirely within b I mean v can lie in a, y can lie in b, or both v y lies in b or it can happened that v lies within the block a b. I mean v has some elements from some elements from b y has some elements from a y has some elements from b.

So, what would imply is that either u v square x y square z is equal to some a to the power k 1 b to the power k 2 a to the power p b to the power p where k 1 plus k 2 is strickly greater than 2 p, or it does not have the form some a star b star a star b star. So, one of these two things will not happen. If this doesnt happen once again you can show that, it will not be of the form a to the power I mean w w; and even if this happens you can argue with a little bit of effort that it will not be of the form w w.

I will just leave the rest of it for you to complete. So, this will show that L 2 is not a context free language. So, one point that I want to mention is that I can choose i to be to have different values depending on what partition I get, because my choice of what I happens after the partition. So, I can always choose it to have a certain value, which depends on what u, v, x, y, z is given to me. So, I will stop here.

Thank you.