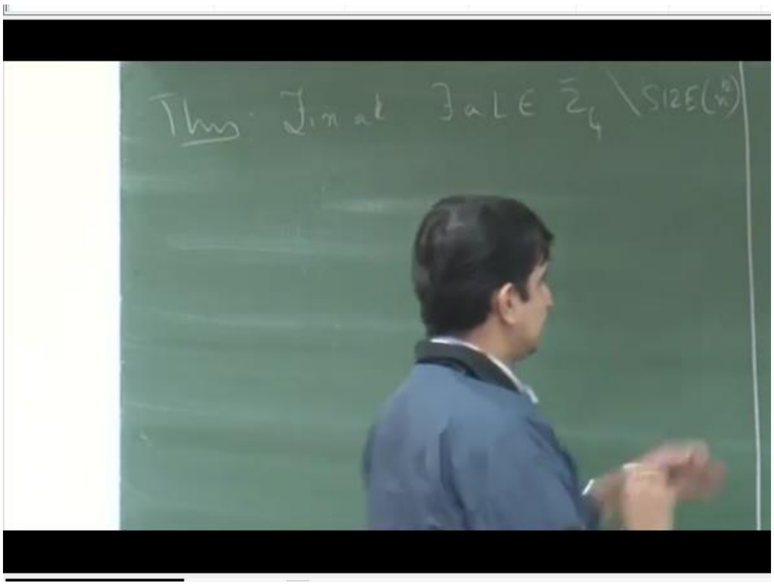
Computation Complexity Theory

Prof. Raghunath Tewari
Department of Computer Science and Engineering
Indian Institute of Technology-Kanpur

Lecture-17
Kannan's Theorem

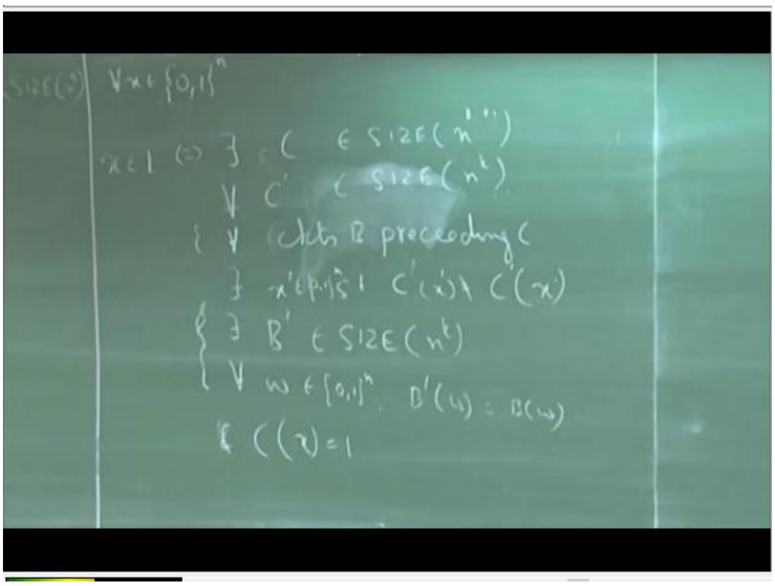
This step quickly says a few things about the result that we saw towards the end of last class**.**

(Refer Slide Time: 00:30)



So, what we saw was if you fix a k some k then there exist a language in sigma 4 - SIZE n to the power k. So, this was the result, I mean for any k you can construct or you can show the existence of a language in which lies in sigma 4 but not in SIZE n to the power k and we saw how to design this language and we also saw that why it is not contained in why it cannot be decided by any family of circuit of SIZE n to the power k. And we gave an explicit sigma 4 algorithm for this circuit. So, what was the sigma 4 algorithm?

(Refer Slide Time: 01:37)



So, for all x so let us look at x of a fixed SIZE, x belongs to L if and only if, so can somebody please tell me what the quantifiers are? So, there exist C of SIZE n well I put k here, so n to the power k + 1 such that for all circuits B preceding C, there exists some string x dash C of x dash is not equal to C dash of x dash. And also there exist some B prime in SIZE n raise to k, such that no matter what string you take w.

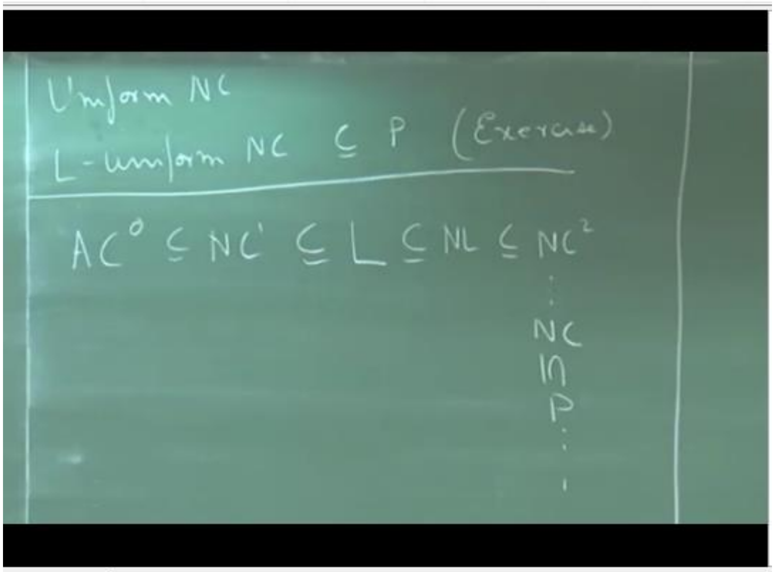
So, x prime is a string here and so is w, B dash of w is the same as B of w and C on x outputs 1. So, Rajiv raised a question that why cannot we just guess a circuit and then for all C prime, verify that C of x prime is not equal to C prime of x prime. So, that is what you are asking last time. So, anybody has any idea what the answer would be, why do we need to also look at all circuits B of this form exactly.

So, you are right. So, basically just look at that of all things. So, suppose we ignore this B thing. So, suppose we just guess a circuit C and just verify that for all C prime does there exists an x prime where C and C prime differ. The problem with that is that you do not have any restriction that the same circuit would be chosen for all inputs of the same length. So, for one particular x you might choose a particular circuit C and for another x you can go ahead and choose another circuit C.

But that is not how the language L was defined. So, L was defined basically to be accepted by a family of circuits which had a size n to the power k + 1. So, you need to ensure that by some means that the circuit that you are choosing for all x is of a certain length they must be the same. So, that is why one way to do that is to enforce that you are always choosing the lexicographically least circuit which has this property. So, that is the reason why you need to check this for all B's. So, that answers your question I guess.

So, that is all I had to say about the proof unless you have any questions. So, we talked about the classes n C and a C and more particularly we talked about the non uniform version of these classes. In other words we just said that a language is in n C if there exist a family of circuits which had a certain property. So, it is an n Ci, if it has log to the power i depth and polynomial size. So, we just talked about existence.

(Refer Slide Time: 07:15)



So, similarly continuing our discussion of uniformity we can define uniform NC as well. So, uniform NC is nothing but there exist a machine which constructs that circuit for you. So, given let us say an input 1 to the power n there, exists some turing machine. So, it suppose so I have not said what kind of uniformity do I require here whether P uniformity or L uniformity.

But suppose whatever it is, there exists a machine which runs in that resource bound and it produces the corresponding circuit for that input length. So, now the thing is that so if you look at uniform, so let us look at L uniform NC, so generally these classes are studied with respect to L uniformity whenever you study a uniform NC or any such family, any such a polynomially bounded family.

So, L uniform NC is contained in the class P. So I will leave that as an exercise, this is not very difficult to prove. So, what basically you have to prove is that given a language which is in L uniform NC there exists a polynomial time machine for it. So, what you do is you take that language, so since it is in L uniform NC there is a log space turing machine M which given input 1 to the power N will output that circuit.

So, now since you have that explicit circuit you just have to simulate that circuit using a polynomial time machine. So, please try that as an exercise. So, now let us just sort of wind up our discussion on circuits or on the families of circuits that we have seen so far. Let us look at what are the known results or what are the known connections between these circuit classes and the other classes that we have seen so far.

So, NC well of course we have a AC 0. So, this is contained in NC 1 and NC 1 is contained in the class L. So, again the reason is that what you can do is so suppose you have an NC 1 circuit. So, again here when we are talking about NC 1 we are actually talking about uniform NC 1 and the reason for that is because if we do not talk about uniform NC 1, so from what we saw last week we can construct very extremely powerful circuits which have very small size and very small depth.

For example if you take a unary undecidable language there exists a very small circuit family I mean when I say very small circuit family each circuit in that circuit family has very small description and that accepts that undecidable problem. So, when we are comparing a circuit family to a class which has been decided by turing machine it only makes sense to talk about uniform families.

So, NC 1 is contained in L, of course we know that L is contained in NL and NL is contained in NC 2. So, maybe later on I will talk more about these relations how these I mean why each of these relations are true but you can also figure it out, I mean it is not very difficult. Probably this is little more difficult than this, this is more easier. So, here you have to show that the path problem has an NC 2 circuit family.

And of course all these set inside the class NC and NC is contained in again P and then we have all the classes that go beyond P. So, I just wanted to give an idea as to how these circuit classes fit amongst the classes that we have seen prior to this. So, that is all I had to say, thank you.