

**Computational Complexity**  
**Prof. Subrahmanyam Kalyanasundaram**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Hyderabad**

**Lecture -63**  
**Summary and Concluding Remarks**

(Refer Slide Time: 00:15)

Lecture 63 - Summary and Concluding Remarks  
26 October 2021 2:29

Goal: Understand Computation. What can we compute/decide if we had  $x$  amount of resource  $y$ ?

Resources: Time, Space, Randomness, Interaction, No. of gates, depth etc.

Topics Covered:

Hello and welcome to lecture 63 of the course computational complexity in fact this is the last lecture of the course and there will not be any technical content. We will just summarize what we have seen over the course and I will share some concluding remarks which I wanted to share. So, let this what I just want to summarize what we have seen. So, overall what was the goal and how have we gone about trying to get closer to that goal.

So, the goal was to understand computation. So, how much can we compute or decide compute in the sense of a function and decide instance of a language if we had  $x$  amount of a resource  $y$ . So, where we considered several resources. So, when I said resource at the beginning maybe this the only resources that we would have been familiar would have been space and time in terms of computational resources.

But then during the course we saw other resources like randomness was a resource that we could use during the computation interaction towards the end of the course we saw interaction being

used as a resource when you when you have the power of interaction we could that could be a resource. In the circuit model the number of gates the depth of the circuit and the fan in of the circuit these were resources. So, these are the resources that we saw we used we try to understand the computation.

So, on the basis of these resources we classified computational problems into complexity classes. So, if there is a language that can be decided in polynomial time then it is called this language belongs to the class  $P$  if a language that can be decided using a polynomial space or it is it is  $P$  space and so on a polynomial time but then random randomized computation two-sided error it was BPP and so on interaction we saw IP AM and so on.

So, these are the resources that we saw and we try to understand many of these complexity classes and try to gain some understanding of it. So, let me just try to list down what all we have seen during the course. So, the topics covered we saw. So, we started with time complexity where we saw things like NP completeness. We started with P, NP and P completeness coculeven theorem polynomial hierarchy then we saw space complexity.

There we saw complexity classes like L, NL, P space P space completeness and so on. We saw results such as savage's theorem savage's theorem then the result of NL equal to co-NL which was Immerman's selection theorem. This is what we saw in I think roughly this is the highlights of what we saw in the space completeness.

**(Refer Slide Time: 03:32)**

\* Space Complexity:  $L, NL, PSPACE, Savitch's theorem, NL = co-NL$   
 \* Oracle TM : PH using oracles, SLS.  
 \* Randomized Comp :  $RP, co-RP, BPP, ZPP,$   
 $BPP \subseteq \Sigma_2 \cap \Pi_2, BPP \subseteq P/Poly$   
 \* Circuits :  $P/Poly, AC, NC, TM, taking advice$   
 $Karp-Lipton, Parity & AC^0$   
 \* Power of counting : #P, completeness, Permanent is #P comp,  
 $Fodor's theorem, PH \subseteq P^{#P}$   
 \* Communication Comp : Basic model, simple techniques.  
 $KW$  relation & connection to circuit depth

Then then we saw oracle turing machines very briefly even though just for one or two lectures we saw what can oracle what are oracle turing machines we saw a polynomial hierarchy using oracles in fact I can also list that in time complexity section I think I already listed that. Then we saw baker Gill Solvay theorem which said that which use oracles to show that P versus NP cannot be settled using diagonalization it cannot be solved using diagonalization.

So, even though the proof used oracles the result is nothing to do with oracle's then we saw random randomized computation one sided error two-sided error and so on. So, RP, co-RP, BPP, ZPP we saw how to boost the probability of success or reduce the probability of error we saw that BPP is contained in sigma 2 and pi 2 we saw that BBP is contained in P by poly. So, this is what we saw in randomized computation then we saw circuits which was another fundamental model of computation where we saw classes like P by poly AC NC.

And we saw the model that were turing machines take taking advice and we saw that that was equivalent to the circuit model we saw results such as Carplift and theorem which said that if SAT has polynomial size circuits then the polynomial hierarchy collapses. Then we saw parity is not in AC 0 which was like one concrete lower bound that we have in circuits then we saw the power of counting if we could count what could we do.

So, till now we had seen decision problems. So, if you could count if you count the number of accepting parts what would you do we saw we saw mainly two things one was the piece of sharp P completeness. We defined the class sharp P sharp P completeness and the fact that permanent is sharp P complete permanent is sharp P complete. And then we saw that we saw today's theorem which was a very surprising or very interesting result that the entire polynomial hierarchy is contained in P with a sharp P oracle.

So, this indicates how powerful is counting. So, if you could do sharp P then you have the entire polynomial hierarchy at your disposal then we saw communication complexity. We saw we saw the basic model and lower some simple techniques basic models some simple techniques we saw the we saw the Kasmir Vectors and relation and connection to circuit depth we saw the we saw the monotone depth monotone depth lower bound for matching.

And finally we saw interactive proofs like what could be shown if we could interact with the brewer instead of just taking a static proof and verifying it yourself. So, what is interactive proofs that was interactive proofs. So, there we saw the model IP which was interactive proofs am where the interactive proofs were the coin the randomness has to be made public then we saw that public coins and private coins are kind of equivalent if you have a private coin protocol.

You could convert it into a public coin protocol then we saw that sharp SAT is an IP and we also mentioned and I also gave the outline of IP equal to P space even though I did not really we didn't really see the proof in full detail but most of the details were similar to the sharp site is contained in IP proof. So, these are the kind of high level idea of what all we did of course all of this many of these results were involved like half a lecture to one lecture to some of them actually span two lectures and many sub topics I may not have mentioned but this is like the high level topics.

**(Refer Slide Time: 08:46)**



zero knowledge proofs. So, this is also a kind of an interactive proof system where there is a provider verifier etcetera but there the provers goal is to convey something convey that something is in the language to the verifier without having the verifier learn anything about it.

So, this is why the prover convinces the verifier but he does not give any other information except that this the statement is true. So, this seems like a very strange thing to say because if I want to convince you that some statement is true but I do not want to give you any more information how is it possible because the statement that  $x$  is true will usually have a proof but then I want to convince you that statement is true but then I do not want to give you any more information.

So, how do I prove it without giving any more proof. So, this is a small area is an area of zero knowledge proof which is very interesting another area that we did not devote any time to is cryptography hard functions let us say or maybe one way function sometimes people call it one way functions. So, basically cryptography is to securely encrypt messages. So, how do you encrypt messages.

So, that the other party or no intruder can detect see what is happening or even if they get hold of the message they will not it should not make any sense for them and it should be difficult for them to decode. So, there is this notion of one-way functions. So, meaning it is easy to to encrypt but then it is not easy to decrypt. So, so it is. So, it is like a one way road where it's easy you can go on one way but cannot come back that way the opposite way.

So, one direction of computation should be easy but the inversion should be hard. So, again this is an interesting area of and complexity theory and you may see some other course offering covering some of these topics another interesting area that we did not cover is a pcps and hardness of approximation sorry. So, there is a one way to tackle NP completeness or to one way to deal with NP completeness is to.

So, if some problem is NP complete what is the next best thing that you can do if you only have polynomial time to deal with it. So, one thing that you can do is what is. So, called

approximation algorithm. So, you cannot get the optimum solution but you can get let us say twice the optimum. So, for instance if you are trying to get the find the smallest set that has a certain property. So, let us say the smallest set has type size 10 for instance the vertex cover has size 10 but you assure that you will produce a vertex cover but that is not too that is not too large.

So, you will say that I will give you something at most twice the size of the smallest vertex cover. So, approximately optimal so this is what is called approximate algorithms and approximation algorithms and PCP's are what is called probabilistically checkable proofs. So, it is an offshoot of interactive proofs we saw that we saw in the last week in week 12 and PCPs interestingly gave rise to the theory a lot of improvement in in the area of the hardness of approximation.

So, we people were able to make statements such as this you cannot this you cannot even approximate this language within this factor unless  $P = NP$  and so on. So, that was an interesting area and that is also very interesting in fact at some places there have been entire courses on PCP's and hardness of approximation. And so and same is true for even cryptography there are courses another interesting topic that is pseudorandomness. So, there are and may be derandomization there are somewhat related topics.

So, pseudo random objects are something that look like random but not are not completely random and um. So, there are like like graphs or other objects sets or sequences which are pseudo random and sometimes it is interesting to understand how they are generated or how you can get pseudo random objects because sometimes it is helpful for d randomization. So, what is d randomization.

So, we saw randomized algorithms or randomized complexity classes and I mentioned in the beginning of just this very lecture that randomness is a resource. So, can we can we so, so it because it is a resource we do not want to we want to try to minimize how much randomness we use. So, can we reduce the randomness of a certain that is used by a certain algorithm. So, that is that is the idea behind de-randomization to reduce the randomness or sometimes maybe even remove the randomness entirely.

So, if you can always remove randomness from a randomized algorithm that would mean that. So, it is like trying to understand randomness. So, we know that BPP contains P but is BPP strictly more powerful than P we do not know is there something that can be solved only using randomness but cannot be solved using deterministic polynomial time we do not know but if any algorithm you can remove the randomness without blowing up the time that means randomness is not really a useful resource.

So, this is one other interesting topic couple of more interesting topics. So, one is quantum computation quantum computing. So, quantum in the quantum world bits are not stored as 0 and or 1 they are stored as superposition of 0 and 1 and then and then when you make a measurement then it becomes 0 or 1 with some probabilities and then using that you are able to do some computations and you need specialized you need slightly different techniques to understand the powers of quantum computation.

So, for instance factorization can be efficiently solved in the quantum setting. So, quantum computation is an interesting topic in itself I am just talking about the computation part of it not the physics aspect of it which is also an extremely interesting field in the physics aspect and one more point that I want to mention is Boolean function analysis. So, this is not really a sub area of complexity theory but it is a related area.

So, you can look at Boolean functions as in functions that are of the form  $f(x) = \sum_{i=0}^n a_i x^i$ . So, the function is 01 in this case. So, this is a Boolean function. So, now we will try to understand this a certain function or a class of functions by studying it. So, one of the popular tools is the Fourier analysis. So, given a Boolean function there is a there is a way to look at it from another domain by doing a Fourier analysis.

But it is not the same Fourier analysis that is using signal processing but it is similar but it is slightly different. And this actually has a lot of applications in complexity theory as well as other as other areas like sometimes combinatorics as well. So, this is another interesting associated

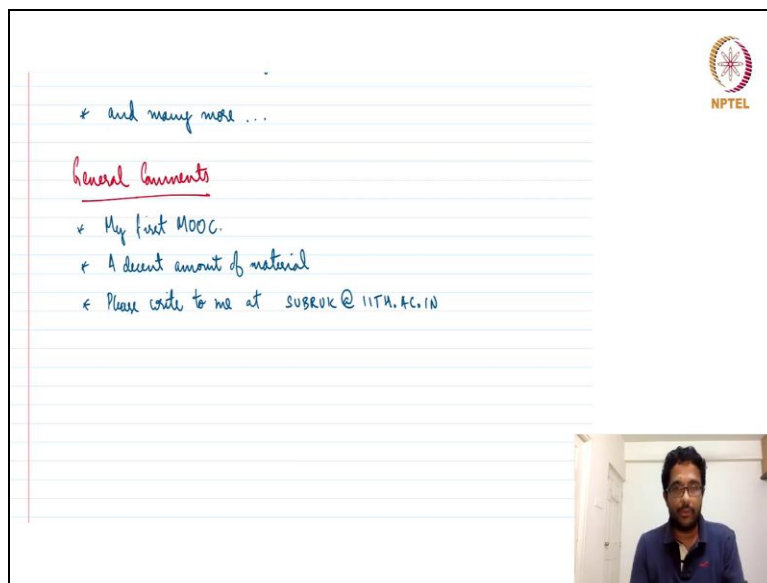


area and many other and many more not the only set of topics many more topics are there that are interesting but then of course we only have we only had 12 weeks.

And I had to pick some topics that need to be that needed to be covered and hence I went with whatever we have in the course and but then there are so, many other interesting topics that that we could think of include that we could have thought of including or including or if you are interested I would certainly say look up each one of them and try to read about it or learn about it. And in fact each of these statements that we said each of these bullet points that we actually covered and even in these within these bullet points there are much more things to be explored. So, you could you could try to read up or read up or understand it yourself.

Again this is a introductory graduate level course in introduction to the area. So, or may be an invitation to computation complexity. So, if you are interested then there is a lot more that you can learn and understand and I hope many people are excited by this area and many of you go on to learn and do much more in this area this course is just the starting and finally some more points some general comments.

**(Refer Slide Time: 20:30)**



The slide features a white background with a blue grid pattern. In the top right corner, there is the NPTEL logo, which consists of a circular emblem with a stylized 'N' and 'P' and the text 'NPTEL' below it. The main content of the slide is handwritten in blue ink. It starts with a bullet point: '\* and many more ...'. Below this, the text 'General Comments' is written in red ink and underlined. This is followed by three more bullet points: '\* My first MOOC.', '\* A decent amount of material', and '\* Please write to me at [surbok@iitah.ac.in](mailto:surbok@iitah.ac.in)'. In the bottom right corner of the slide, there is a small rectangular video inset showing a man with glasses and a blue shirt speaking.

So, one is that this is the first time I am delivering a MOOC or the first time I am also teaching on over the NPTEL platform. So, there could have been issues with my with the way I delivered stuff or with the way I chose stuff. So, there could have been many many things that could have

been done better. So, but then that is the challenge is that that. So, and I do not have real time feedback while teaching but I know that many of the students may be learning from different backgrounds.

So, I have tried to try my best to explain the content to as detailed as I could possibly but at the same time I also wanted to do justice to the to the course material it is a graduate level course or which is a post graduate or Ph.D level course in computational complexity. So, we I also want to do justice to the course contents. So, I had to balance between these two. So, my first MOOC over and first one over NPTEL and in fact I think we covered a decent amount of material we covered a decent amount of material in this course. Not very much but it's a decent amount that we have covered.

And so, if you have I will be happy to hear any feedback from any of you I know the NPTEL has a mechanism to collect feedback but then I also welcome you to write to me directly. So, please write to me at uh. So, my email id is subruk at IITH dot ac dot in if you Google it should not be too hard to find subruk at IITH dot ac dot in. And feel free to let me know what contents you like what could have been done better and are there some content that is some topics that are completely hopeless that you would like me to redo I can of course check with NPTEL and if there is a provision to replace some of the video lectures.

If there is sufficient if people feel that it should be replaced with a different presentation. And another question that I have to think of when I am when I am designing what to teach and even if I decide I will teach this theorem. So, there could be a specific theorem let us say permanent is sharply complete there are if you look at textbooks and if you look at different lecture notes at different places there are multiple ways some even though the proofs is essentially sometimes the proof is essentially only one but then there are multiple ways the same proof can be presented.

And so I have to make a choice of which of these presentations is the best to be for this particular mode of interaction. So, there are so many choices that one has to make while teaching. So, the point is if you have any feedback or comments I always welcome feedback and comment and feel free to be honest and candid with me on that. And anything about anything about the speed

of the course about the pace of the course is too fast too slow because this is like a static delivery without feedback it's not like we are interacting.

So, this we just saw interactive proofs. So, of course we have interactive live sessions but then that is that is not it is not like every after every lecture we have one track decision like we had like five six interactive sessions throughout the twelve weeks. And even again if you are if you are a viewer who is just happen to watch the lectures on YouTube or the NPTEL platform but not really registered and learning and you happen to reach the last lecture of the course and you are hearing me say this you are also welcome to write comments to me and yeah I think that's about it please feel free to write feedback.

So, any feedback is welcome any feedback is more than welcome yeah and I hope you have you had fun learning the course that is all from me, thank you.