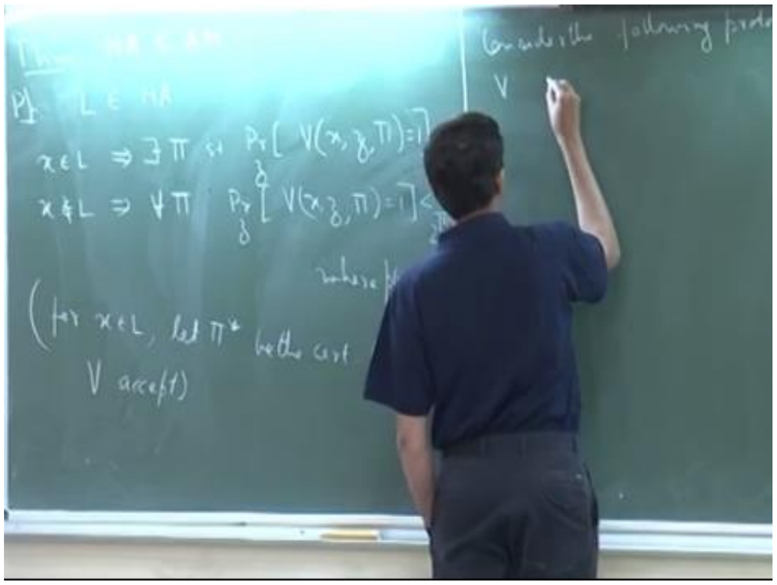
Computational Complexity Theory

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

Lecture -33
Introduction

(Refer Slide Time: 00:21)


Thm: MA ⊆ AM
Pf: L ∈ MA
x∈L ⇒ ∃π st Pr[V(x,z,π)=1]
x∉L ⇒ ∀π Pr[V(x,z,π)=1]<
where
(for x∈L, let π* be the cert
V accept)
Consider the following proto
V

So, the first thing that we will see today is the following theorem that a Merlin Arthur kind of a protocol can be simulated by an Arthur Merlin protocol. So, Arthur Merlin is more powerful in some sense. So, this is not very difficult to see. So, let us prove this. So, first let us recall. What a Merlin Arthur protocol is so let L be a language in MA. So, what does it mean? So, it means that for all x, x is in L so what does that mean by definition?

So, it means that so Merlin who is the prover in this case, so since x belongs to the language there is some proof or there is some certificate which the prover sends to the verifier; And then the probabilistic verifier can verify whether that is a correct certificate and ends up accepting x. And if x does not belong to the language then with high probability the verifier would reject for all possible certificates that the prover can send.

So, if x is in L let me call the certificate pi in this case. So, there exist a certificate pi such that the probability over all random string z that the verifier accepts x. So, the verifier now has accepts two x the random string z And also the proof that merlin had send is equal to 1. So, this means that the verifier accepts so this probability I will assume it to be 1. So, although in the definition of interactive protocols we just had it two third one, third kind of probability.

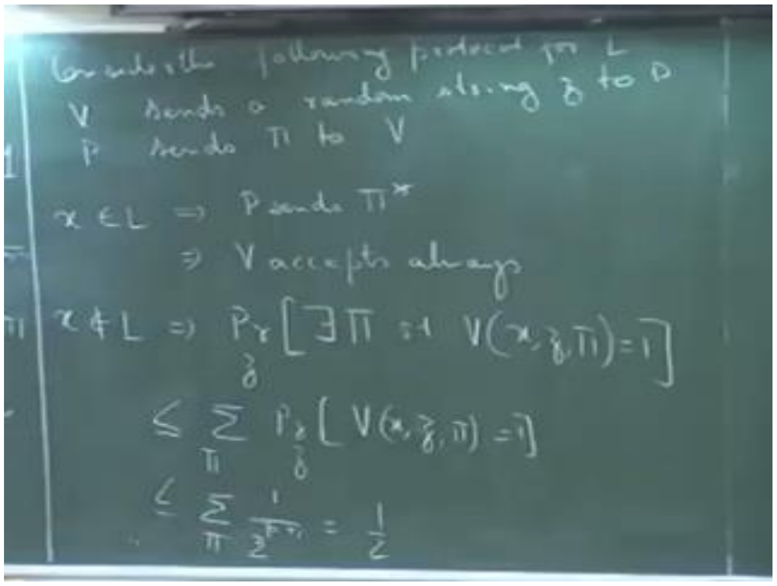
But if you look at I think problem four in your third assignment there actually you can show, that the acceptance probability can be boosted all the way up to 1. So, that is part of your assignment and the idea for that is basically very similar to the Sipser–Gacs proof that we had discussed where we showed that BPP is contained in sigma 2. So, what you the way this can be shown is I mean this probability can be boosted by another protocol which may have more rounds.

So, it is not just a one round or two rounds, protocol in this case but then the number of rounds can again be decreased. So, I just assume perfect probability in this case and if x does not belong to L then no matter what proof the prover sense probability that V x, z pi is equal to 1. Again we will amplify the rejection probability in this case which means that we will just assume that this is less than 1 over 2 to the power p, x plus 1 where p of x is the length of the certificate that the prover sends.

So, again this we have already seen in various contexts; that for any randomized algorithm you can boost the success probability, and the boosting is basically done by taking multiple trials. So, the crucial point here is that so suppose the prover sends a certificate which has some length p of x. When you do this probability boosting the length of the certificate does not increase;

It is only that the length of this random string which increases. Because it is the verifier who does this trial the verifier gets a certificate pi and then he conducts many trials and then based on the results of those trials he can finally say yes or no with a very high probability. So, the point is that, this I mean the length of this string can be assumed to be some fixed polynomial apriori. So, it is only the random string z which gets increased because of this boosting.

(Refer Slide Time: 06:32)



Consider the following protocol for L

V sends a random string z to P

P sends π to V

$x \in L \Rightarrow$ P sends π^*

$\Rightarrow$ V accepts always

$$x \notin L \Rightarrow \Pr_z\left[\exists \pi \text{ s.t. } V(x, z, \pi) = 1\right]$$

$$\leq \sum_{\pi} \Pr_z\left[V(x, z, \pi) = 1\right]$$

$$\leq \sum_{\pi} \frac{1}{2^{p+1}} = \frac{1}{2}$$

So, how do we show that L is in AM? So, now consider the following protocol. So, for x in L let pi star be the certificate that makes the verifier accept. So, there can be many certificates but let us just fix some certificate and we will just denote that with pi star. So, what the verifier does first is will consider an AM kind of a protocol? So, the verifier sends a random string z to p. And the prover it sends pi to verifier.

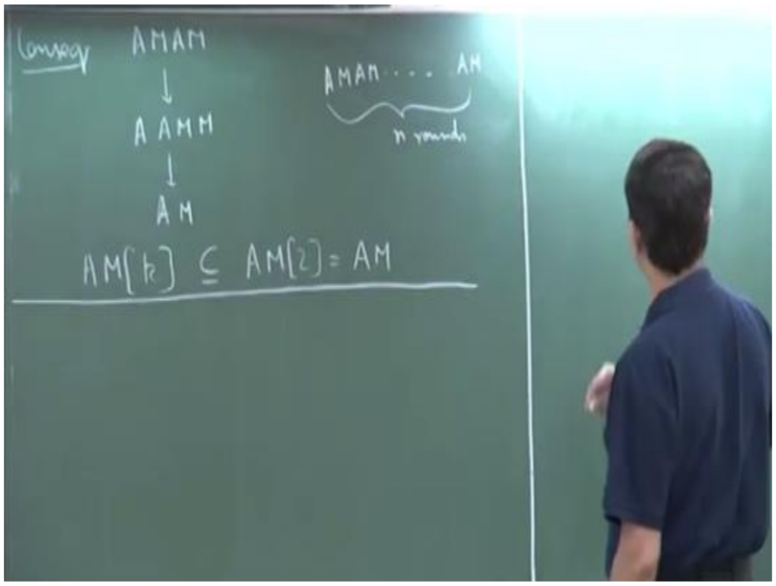
So, now suppose we will consider two cases suppose x belongs to the language. So, then we know that no matter. What string the verifier sends the prover will always send back this pi star? Because if x belongs to the language there exists some pi star and the prover is all powerful he always knows this thing. So, he will just send back the correct pi star and so the prover sends pi star which implies that V accepts always.

Because V has now its own random string which it has generated earlier. And it has the correct certificate pi star so it just runs the verifier algorithm and it will accept x. But what if x is not in the language, so, the probability that the verifier accepts. Let us see with what probability the verifier will accept in this case; So, for random string z that was generated by the verifier the probability that it accepts is that there exist some certificate pi such that V x, z pi ends up accepting and applying the union bound.

This means that if I this probability is less than sum over all certificates equal to1. And how much is this probability? So, by our assumption this probability is 1 over 2 to the power whatever this p is p plus 1. But now how many certificates are possible? So, the since the length of this certificate is p of x. The total number of such certificates is 2 to the power p of x which means that this is equal to half.

So, if x is not in L for any random string. No matter what certificate the prover sends the probability of acceptance is at most half. So, this gives us a Arthur Merlin protocol for Merlin Arthur kind of a language.

(Refer Slide Time: 12:18)



So, this has a very nice consequence so what this allows us is always to switch a Merlin Arthur protocol to Arthur Merlin protocol. So, suppose we have something like AM, AM kind of a protocol. Where verifier sends some question the prover replies back; again the verifier sends something else. And again the prover answers back and based on which the verifier takes a decision.

So, what this can so what we can do here is that this MA can be replaced with an AM. From this what we get is that AAMM. But then this is nothing but the verifier asking two questions? And the proofs are answering back with 2 answers so this is the same as AM. So, in fact what this allows us is that is the following corollary; that if we have an Arthur Merlin protocol with k rounds.

This is contained in 2 rounds which is by definition the class AM. And not only for Arthur Merlin and even if you have a Merlin Arthur kind of a protocol with k rounds everything can be sunk down to just a level 2 kind of a protocol. So, this is a nice corollary. But the constraint is that it has to be constantly many rounds. Because if you look here I mean the assumption that we make is so let me put it this way.

So, what happens if k is not constant? So, let us say k is n or something it depends on the input length. I mean why will this not hold? It is similar to the constructing different turing machines but where does this proof break down no that is fine; I mean so suppose if I have something like this AM, AM so on AM where the number of such rounds is let us say n. What will happen? So, the point is that how many times will I have to apply this theorem to get AM in this kind of a protocol?

How many times? About n about order n; so n by 2 or probably n by 2, something like that. And the point is that it is about order of n but each time I apply this theorem; what I end up with a random string which is polynomially large in the size of the input. So, if I apply it n times; what I will end up getting is a very large random string and the random string itself will be exponentially large.

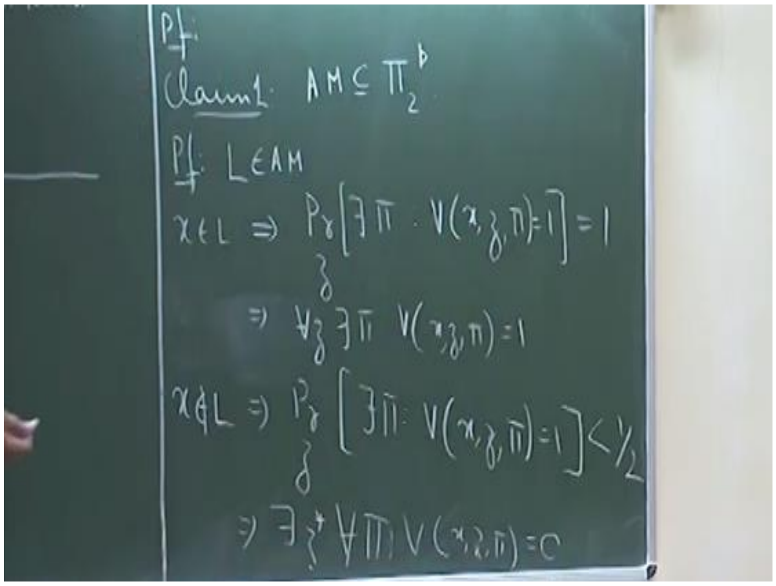
In other words what it would mean is that this verify will no longer be able to even read the random string in polynomial time. Because whatever we do I mean we cannot allow the verifier to go beyond polynomial time. So, this random string that we start off with that length has to be bounded by polynomial. So, that is why at least this strategy fails if we have more than constant rounds.

Because, so suppose we; start off with a random string which has let us say size n square. When we amplify this probability to be 1 over 2 to the power p x this random string; becomes some n to the power k because of this amplification. Because you need some polynomially many rounds to get this amplified. The next time whenever when I do an another replacement of another MA protocol with an AM protocol.

Again an n to the power k length random string gets concatenated with it. So, each time n to the power k thing gets concatenated with it. So, if we have about order n rounds the total size of this random string will be much larger. So, let us look at a different kind of a result now. Where exactly do these classes stand I mean AM, MA and all these other protocol I mean all these other different kinds of protocol;

Where do they stand with respect to for example the polynomial hierarchy and other known classes. So, last time we showed that graph non-isomorpism is in AM. So, what if graph isomorpism is NP complete?

(Refer Slide Time: 18:01)



Pf.

Claim1. $AM \subseteq \Pi_2^p$

Pf. $L \in AM$

$x \in L \Rightarrow \Pr_z[\exists \pi . V(x,z,\pi)=1] = 1$

$\Rightarrow \forall z \exists \pi \; V(x,z,\pi)=1$

$x \notin L \Rightarrow \Pr_z[\exists \pi : V(x,z,\pi)=1] < 1/2$

$\Rightarrow \exists z \forall \pi \; V(x,z,\pi)=0$

So, if graph isomorpism is NP complete then the polynomial hierarchy actually will collapse to the second level; So, we know that graph isomorpism is in NP and graph non-isomorpism is in co NP so that is easy to see. But if graph isomorpism is as hard as an NP complete problem then it has the implication that, the polynomial hierarchy will collapse. And since this is not believed to be true this assumption is also not believed to be true.

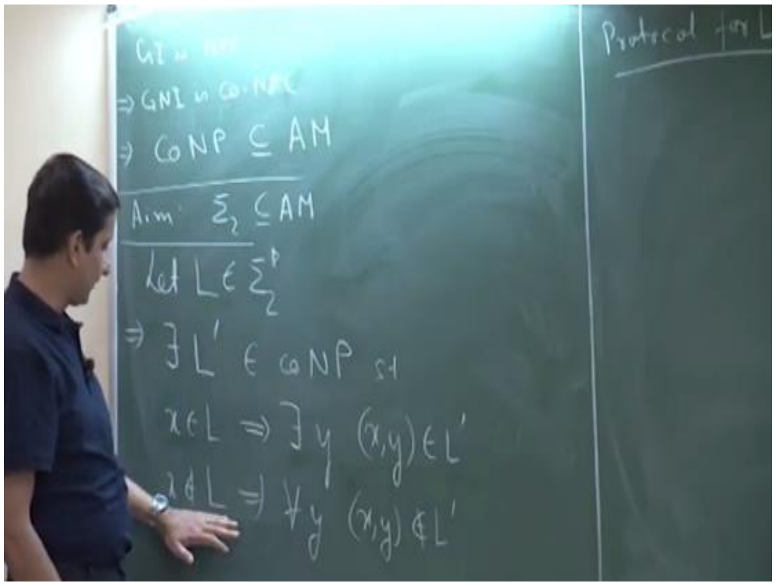
So, let us see the proof of this theorem so let us make the following claim first, AM is contained in pi 2 p. So, why is this true? So, again this is not very difficult to see it kind of follows by definition. So, let us look at a language in AM which means that if x is in L then the probability that for a random string z, there exist a correct certificate such that the verifier given x, what was the notation said and pi ends up accepting is 1 and if x does not belong to L.

Then the probability that for a random string z, there exist a good certificate which makes the verifier accept is let us say less than half. But what does this mean? So, this essentially means that since the probability is one, it just means that no matter what random string you choose there always exist a good certificate. So, I can just rewrite this as for all z there exist pi such that V x, z pi ends up accepting.

Since the probability of this event is 1, it just means that it does not depend on the string z and what about this event? So, here what we know is that at most half the number of z is end up making x accept. What it means is that there are at least half the number of z; which will make the verifier reject x. So, I can just rewrite this as if x does not belong to L then there exists some z. So, let me just call this z star which will lie in this other half not the half which makes v accept but from the other half;

Such that no matter what certificate the prover had sent the verifier would always reject. So, this is not something very difficult it basically just follows from the definition of AM. So, now let us see what we have.

(Refer Slide Time: 23:16)


GI in NPC
⇒ GNI in Co-NPC
⇒ CoNP ⊆ AM
Aim: $\Sigma_2 \subseteq AM$
Let $L \in \Sigma_2^p$
⇒ $\exists L' \in$ coNP st
$x \in L \Rightarrow \exists y\ (x,y) \in L'$
$x \notin L \Rightarrow \forall y\ (x,y) \notin L'$
Protocol for L

So, since we assume that graph isomorpism is NP complete, so GI is NP, complete this implies that graph non-isomorpism is co NP complete by definition. But what do we know about graph non-isomorpism? So, graph non-isomorpism is in AM. So, if graph non isomorpism is going to be complete what does that imply? It implies that co NP is contained in AM. So, basically this is the hypothesis that we will use to get this collapse.

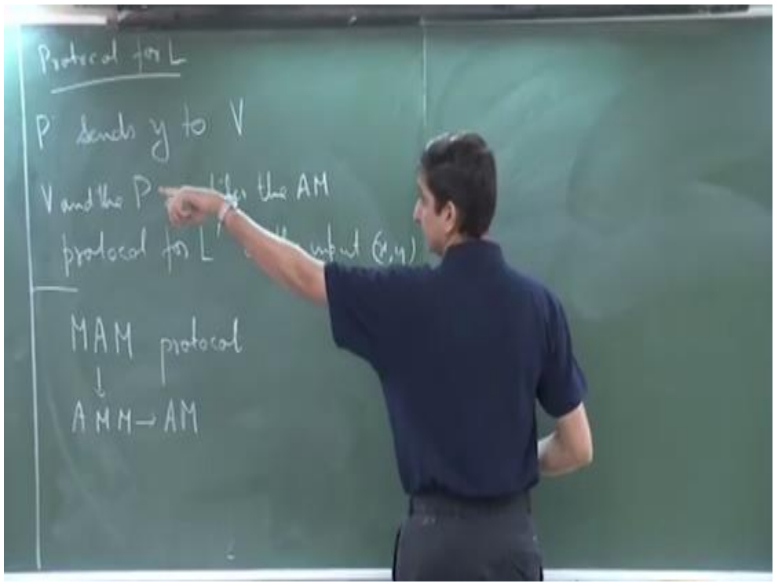
So, basically what we will show is, so our goal will be to show that so our aim is to show that sigma 2 is contained in AM. Because if you can show that sigma 2 is contained in AM by claim 1; it will imply that sigma 2 is contained in pi 2 and if sigma 2 is contained in pi 2; Then these classes are equal which would imply that the hierarchy collapses to sigma 2. So, this is what we have to show. So, let L be a language in sigma 2 p.

So, what does that mean by definition? That there exists some language L prime in pi 1 but pi 1 is nothing but co NP. Such that x belongs to L implies that there exists a y such that x, y is in L prime; and if x does not belong to L it means that for all y, x, y does not belong to L prime. So, nothing fancy here this is just the definition of sigma 2 p. So, how do we replace this with an AM protocol? So, any idea how to get an AM protocol for L now?

So, now note that you can assume this I mean you have the hypothesis that co NP can be simulated by an AM protocol and L prime is a language in co NP. How do we get an aim protocol for L? We also know that N P is contained in name we do know that. I mean if sigma 2 is contained this is what we have to show. We know that NP is contained in M; because NP is contained in MA by definition and MA is contained in A.

But we do not need so much actually it is not that deep. So, what is this language L now? So, let us look at this language L; So, we say that a string belongs to the language if there exist a y such that x, y belongs to a co NP language or x, y can be simulated by an AM protocol; and we say that x does not belong to the language if no matter what y is produced x, y will not belong to this language L bar.

(Refer Slide Time: 28:04)

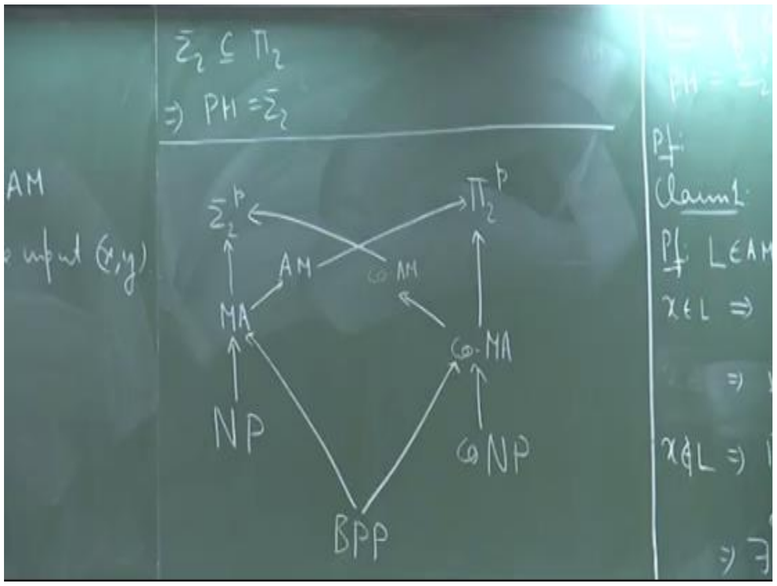


So, the protocol for L is as follows the prover just sends y to the verifier; whatever that y is I mean if x does belong to the language. Then we know that there exist a y and otherwise not; and next the verifier and the prover simulates the AM protocol whatever it is for L prime on the input x, y. So, if x belongs to L then we know that there exist a good y such that this protocol will end up accepting the string x and if x does not belong to L.

Then no matter what y the prover sends this protocol will end up rejecting x with high probability. So, what is the type of this protocol? So, this is not exactly AM protocol. I mean this is slightly more than an AM protocol. So, this part is AM but here we have an additional round where the prover first sends something. So, this is basically an MAM protocol, but again I mean by the earlier theorem that we did;

So, this can be replaced this M A part can be replaced with an AM and this would this is nothing but an AM protocol. So, what we have so far is that sigma 2 is contained in AM.

(Refer Slide Time: 30:43)



So, which implies that sigma 2, is contained in pi 2,and this implies that the hierarchy collapses to the second level. So, in other words it is not believed that co NP is contained in AM. So, we know that NP is contained in AM but AM is not so much powerful that it can contain a co NP. So; any questions about this proof or anything else? So, now let us make a small summary and let us see where these classes stand.

So, let us draw a diagram showing the relations between these classes. So, let us start with NP and co NP. So, MA is contained in NP by definition; and again MA is contained in sigma 2 p by definition M A is also contained in a AM and let us put pi 2 over here. We just argued that AM is contained in sigma 2 b. So, why is this true why is MA contained in sigma 2 p? So, the exactly it is basically the same argument as this.

In other words it just follows from definition; I mean we say that L is in MA if for x, in the language there exist a certificate which makes the verifier accept with high probability. We can also assume perfect probability which means that there exist pi such that for all z V x, z, pi is equal to 1. And in the other case there will be some if x does not belong to L. No matter what certificate you choose there will be some z which will be a bad z.

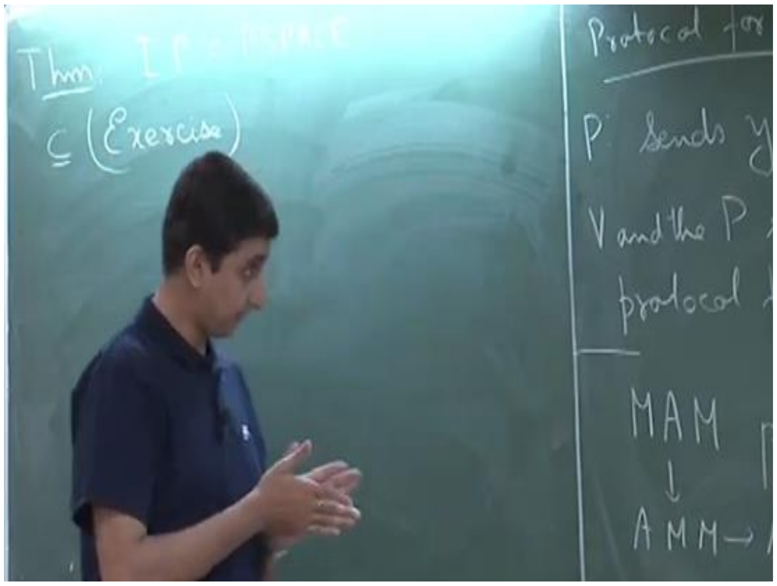
In other words basically it is the same argument. So, similarly we have the complement of these classes; we have co MA and we have co AM with similar relations. And also we can plug in BPP over here. So, what is the relation of BPP with respect to these classes? So, BPP is by again by definition contained in MA. So; because we allow a probability of error in one side. And for the same reason it is contained in co MA as well.

So, this is what our current understanding of these classes. Sigma 2 is contained in AM? No that is that is why we got the contradiction, so because we assume that graph isomorpism is NP complete we got that sigma 2 is contained in M which gave us the implication that, these two classes are the same which makes PH collapses, which is not believed to be true. Here NP is contained in MA. Because so what is the definition of MA?

So, let me just write that down, so we say that L belongs to MA if for x in L so the prover first sends a certificate; Such that the probability that V x, z pi is equal to1 is 1, and for x not in L for all proofs the probability that this is equal to 1, is less than half. So, the point is that with constant with a proof system having constant number of rounds; what we see from this picture is that? It is not very much different from our class NP and the complement classes are not very much different from co NP.

In fact the events are believed to sit below the second level of the polynomial hierarchy. But what is very surprising is that and what is known as is the following very important theorem is if we allow multiple rounds;

(Refer Slide Time: 35:16)



I mean if we allow non constant number of rounds in fact if we allow linear number of rounds I P, which is what the class is becomes same as this much larger class p space. So, one direction is quite trivial; so this part so showing that IP is contained in p space is not very difficult. So, I will just leave this as an exercise. So, how do you simulate so let us see if you have a interactive protocol having n rounds how do you simulate that using a p space machine.

That is not very difficult the difficult part is showing that p space is contained in IP. And recall that p space also contains the polynomial hierarchy. So, what this implies is IP is powerful enough to capture all of the polynomial hierarchy also. So, this was very surprising, I mean in the light of these results this came out as a very surprising fact to complexity theorists in the early 90s but this indeed is true. So, in the next couple of lectures we will see the idea behind this proof. And so let us stop here.