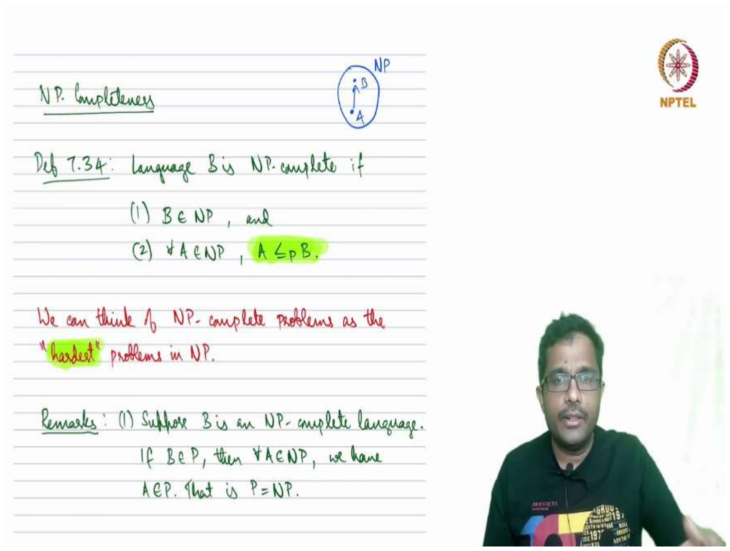
Theory of Computation
Professor Subrahmanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Polynomial Time Reductions - Part 1

(Refer Slide Time: 00:16)



NP. Completeness

Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

Hello and welcome to lecture number 49 of the course theory of computation, this is going to be a quick but very important lecture. In the previous lecture, we saw the notion of polynomial time reductions. So, we said that A is polynomial time reducible to B, if there is a polynomial time computable function f such that whenever f takes w that is in A it maps to something that is in B and whenever f takes w that is not in A, it maps to something that is not in B. So, that is the notion of polynomial time reductions.

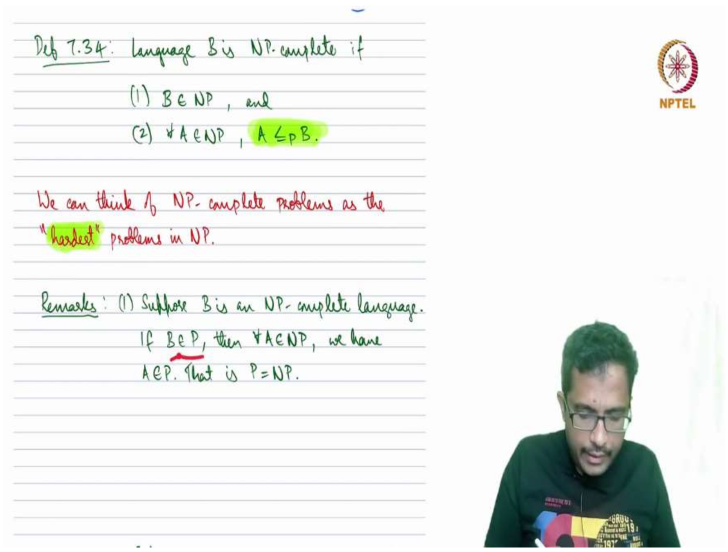
So, it is exactly like mapping reductions, but for the fact that there is a polynomial time bound on the machine that computes the reduction. In this lecture, we will define what is NP complete. So NP completeness is a very very important concept as you may have heard.

So, we say that a language B is NP complete, if the language is in NP and all the languages in NP are reducible to B. So, suppose this is the class of all languages in NP. So, now B itself must be in NP and all the languages A in NP must be reducible to B.

So, this means that, if you can decide B, and then we know that all the languages in NP are reducible to B. So, if you can decide B then indirectly that gives a way to decide any language A that is an NP. So, two things, first of all B must be in NP and all languages in NP must be reducible to B. So, what it means is that, if we can have we have a way to decide B then we have a way to decide any language in NP because of this reduction that we are mentioning here all the languages A in NP must be reducible to B.

So, because of this, it is convenient to think of the NP complete languages or NP complete problems as the hardest problems in NP. Because if we have a way to decide this problem, let us say an NP complete problem, then we have a way to decide any problem in NP. So, it is because of the reduction from all the languages in NP to B.

(Refer Slide Time: 03:24)



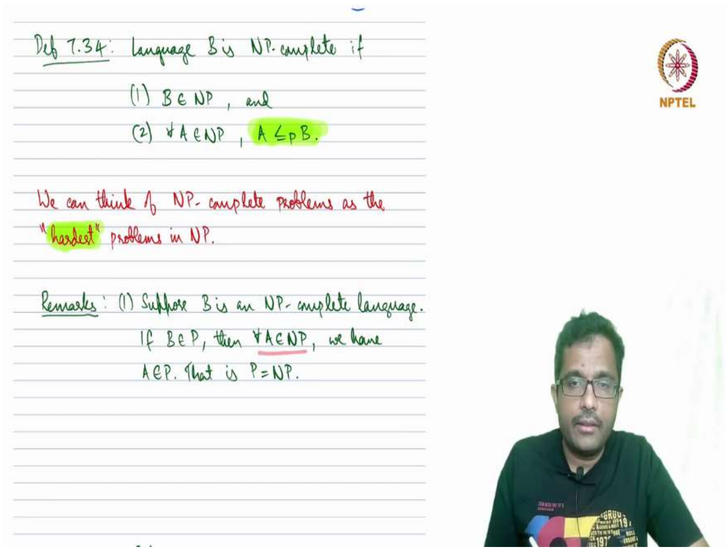
Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $\underline{B \in P}$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.



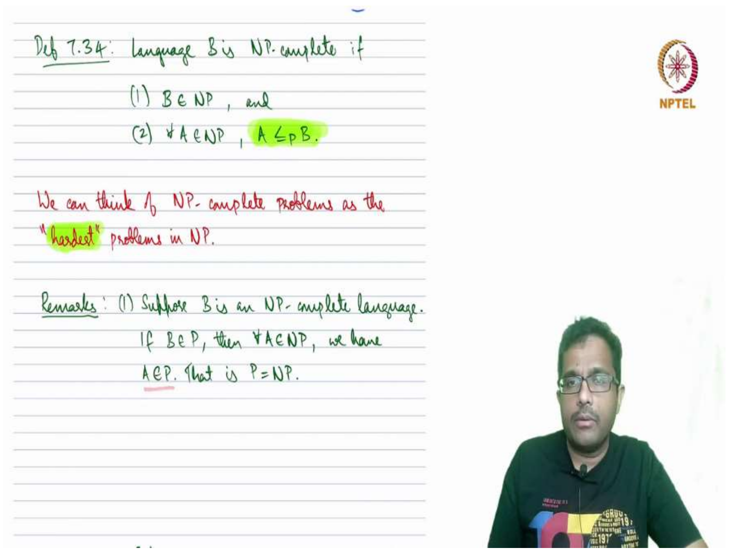
Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\underline{\forall A \in NP}$, we have $A \in P$. That is $P = NP$.



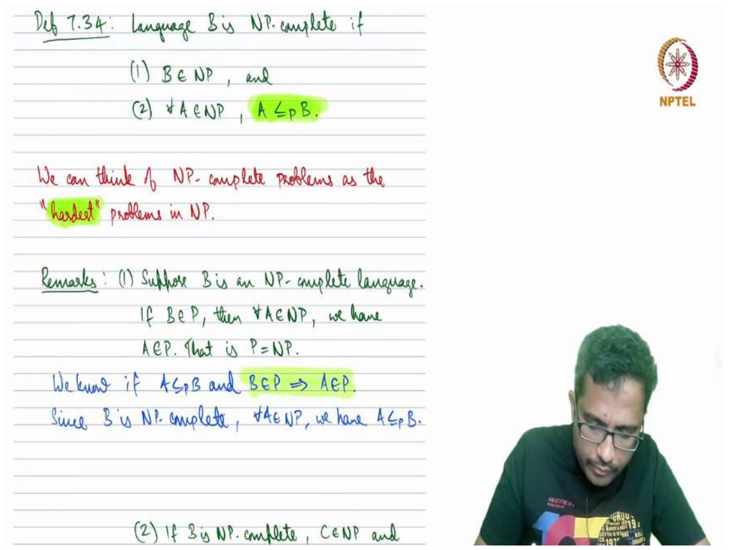
Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.



Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

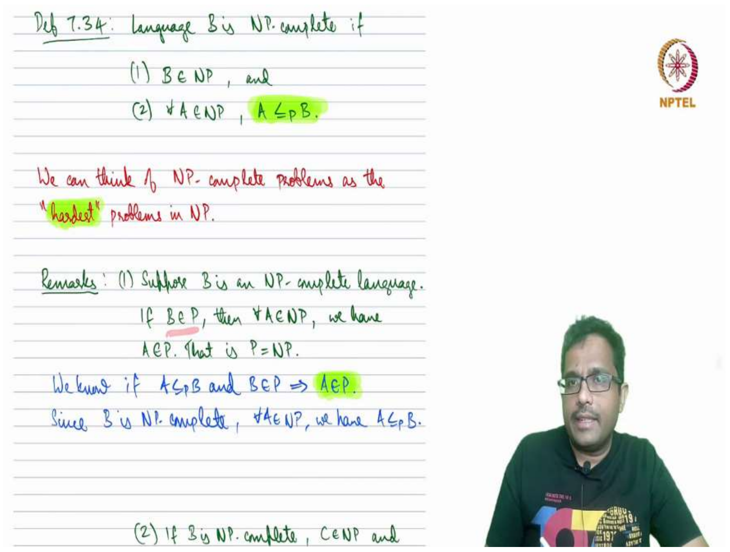
We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP complete, $\forall A \in NP$, we have $A \leq_p B$.

(2) If B is NP complete, $C \in NP$ and



Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP complete, $\forall A \in NP$, we have $A \leq_p B$.

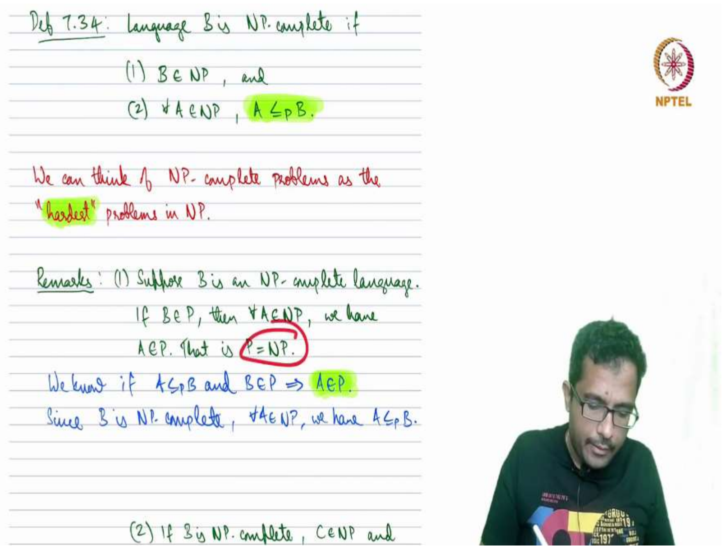
(2) If B is NP complete, $C \in NP$ and

So, let us formalise this these points. So, there are two remarks that I make, suppose B is an NP complete language. Now, if B is in P, meaning if we find a polynomial time algorithm for B, if you find a polynomial time algorithm for B, then we have a polynomial time algorithm for all languages in NP, all languages A in NP are solvable in polynomial time.

Which means that all the languages in NP are solvable in polynomial time, which means that P is equal to NP, why is this? Because we already have from last class, we know if A is reducible to B in polynomial time, and B is in P, then we have A is in P. So, now, we know that all languages A are reducible to B. Since B is NP complete, for all A in NP we have A reducible to B in polynomial time.

So now because of this the above conclusion is true. All the languages in NP are reducible to B. And suppose we have a polynomial time decider for B. That is the assumption. If B is NP complete and If we find a polynomial time decider for B, then we have polynomial time deciders for all languages in NP.

(Refer Slide Time: 05:08)



Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language.

If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

(2) If B is NP-complete, $C \in NP$ and

Meaning we have P equal to NP. So, you may recall the P versus NP question that we asked a couple of lectures back. So, the question was, is P a proper subset of NP or is P equal to NP? So, we know that P is contained in NP. So is P a proper subset or is it a is it equal. These are the two possibilities.

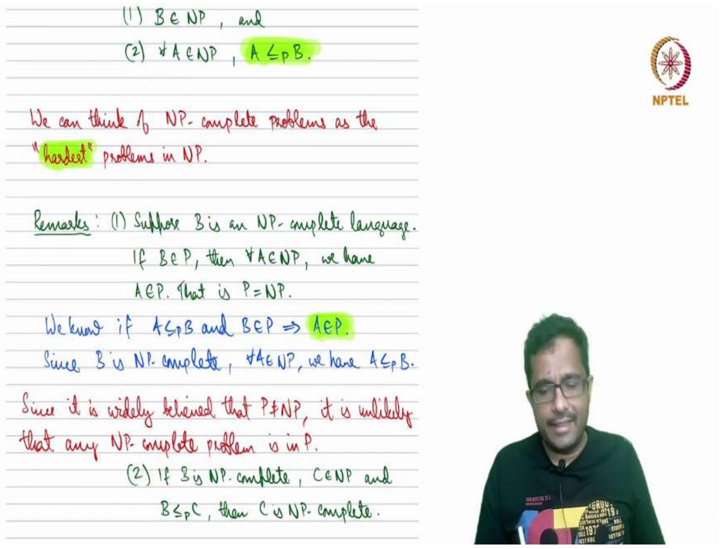
So now, suppose it is the case that P is equal to NP, what would be one way to show it. How would we show that P is equal to NP. One way to show P equal to NP will be to take an NP complete problem and show that this problem has a polynomial time decider.

So if you want to, if you want to attempt it, I am not suggesting or recommending to attempt but if you want to attempt to show that P is equal to NP. This is the way to go about this. Because, if you show a polynomial time decider for any language, any NP complete language, automatically this implies a polynomial time decider for all the languages in NP.

So, this is one point. So, this is sort of the significance of NP complete languages. If you can show that just any NP complete language, if you can show a polynomial time algorithm for that, that would imply the whole of P equal to NP. And since it is believed that P and NP are distinct, P is not equal to NP.

Since that is a commonly held belief, we usually do not expect any NP complete language to have a polynomial time decider, because such a decider would imply that P equal to NP. So, whenever we think of an NP complete language, we think that it is very unlikely for it to have a polynomial time decider.

(Refer Slide Time: 07:04)


(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

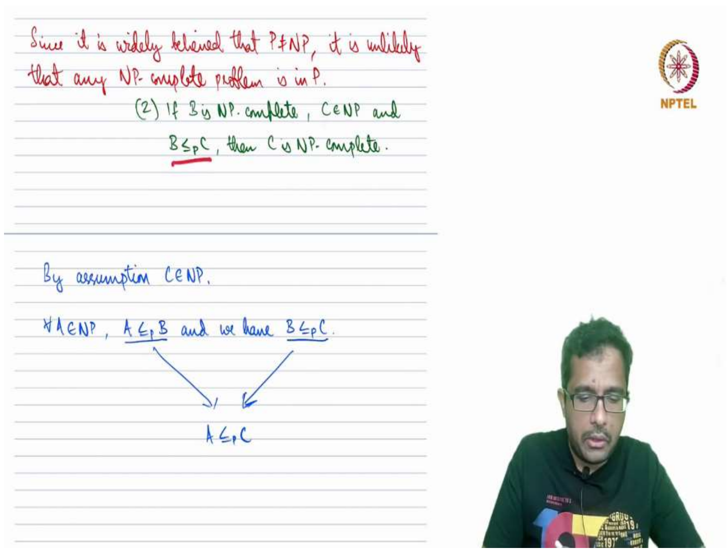
Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

Since it is widely believed, so, I do not want to make a conclusive or everybody does not believe this. P is not equal to NP it is unlikely that any NP complete problem is in P. So, whenever there is an NP complete problem, it is very very unlikely that it is P. So, this is one remark.

(Refer Slide Time: 08:08)



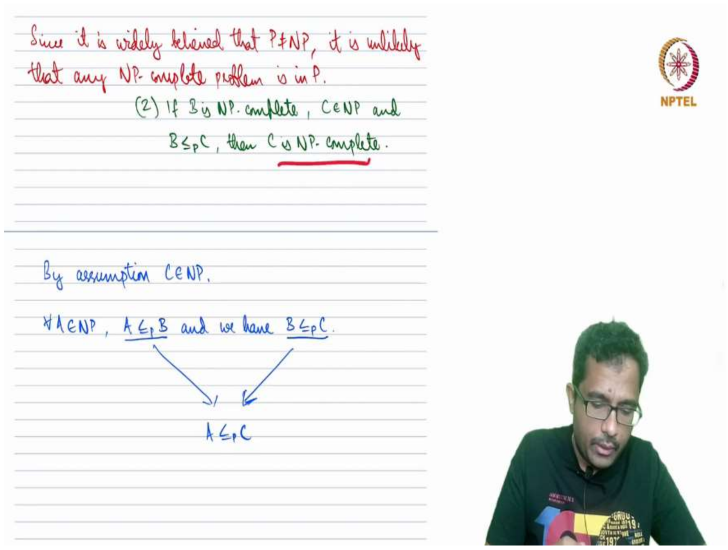
Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $\underline{B \le_p C}$, then C is NP-complete.

By assumption $C \in NP$.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$$\begin{tikzcd} A \le_p B \arrow[dr] & & B \le_p C \arrow[dl] \\ & A \le_p C & \end{tikzcd}$$



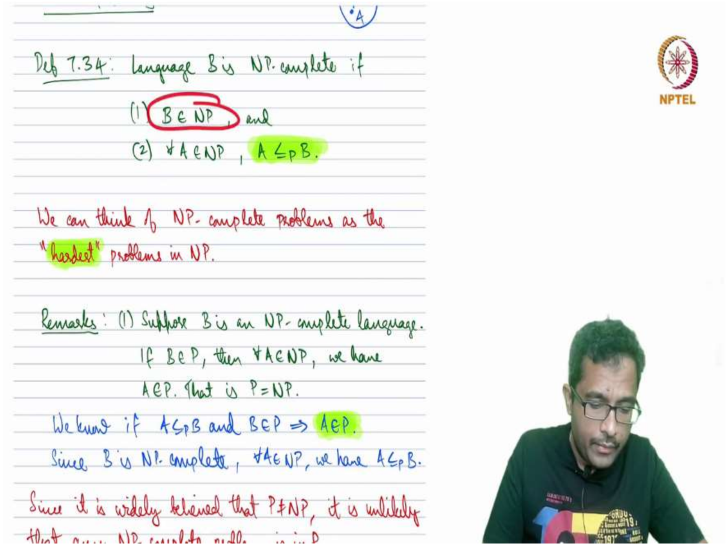
Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \le_p C$, then $\underline{C \text{ is NP-complete}}$.

By assumption $C \in NP$.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$$\begin{tikzcd} A \le_p B \arrow[dr] & & B \le_p C \arrow[dl] \\ & A \le_p C & \end{tikzcd}$$



Def 7.34: Language B is NP-complete if

(1) $\boxed{B \in NP}$, and

(2) $\forall A \in NP$, $A \le_p B$.

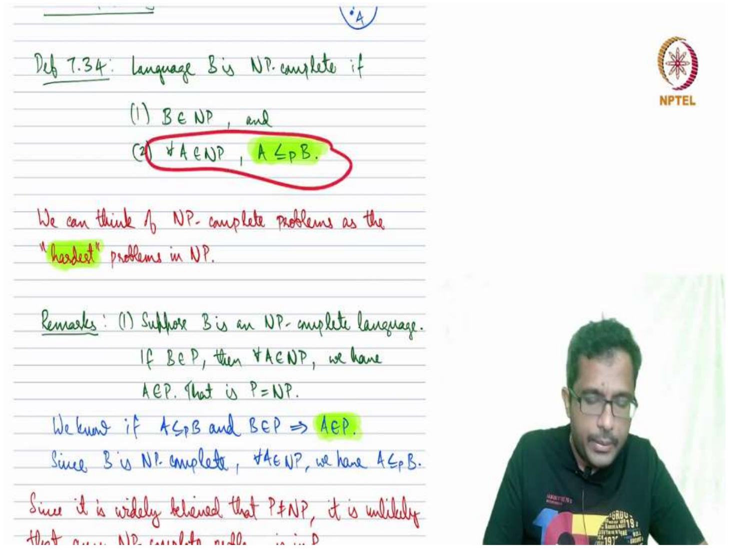
We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \le_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.



Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

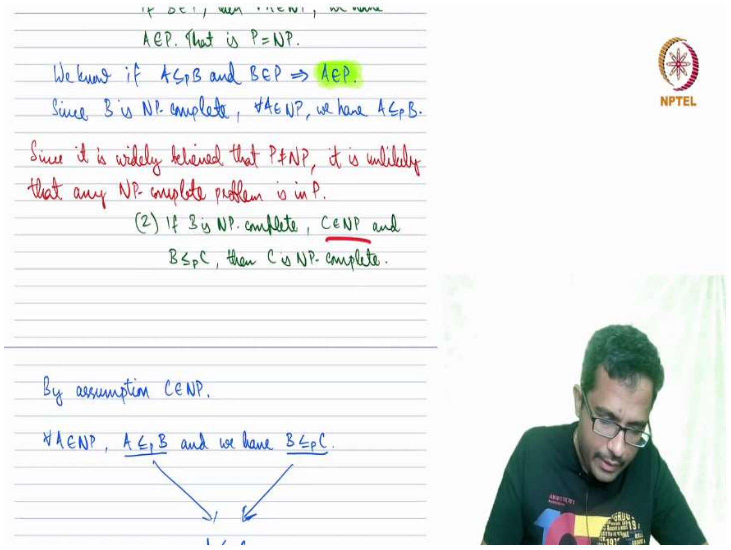
Remarks: (1) Suppose B is an NP-complete language.

If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely



$A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

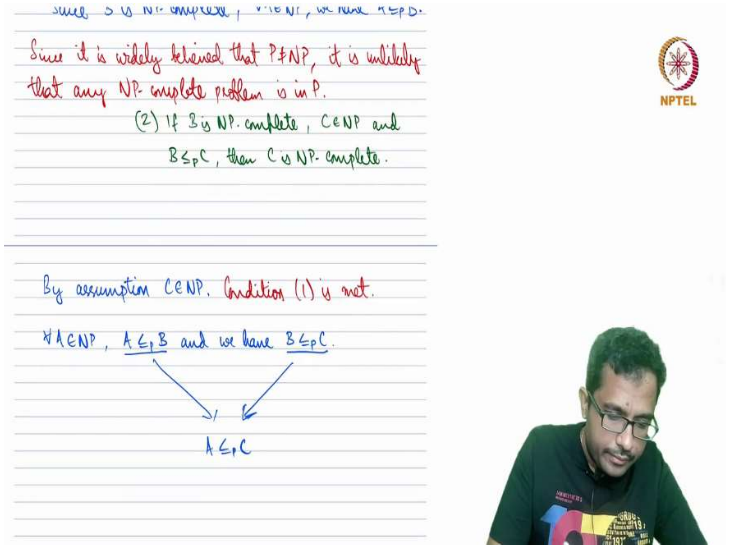
By assumption $C \in NP$.

$\forall A \in NP$, $A \leq_p B$ and we have $B \leq_p C$.

The second is that suppose B is NP complete. And suppose there is another language C is that is in NP. B is NP complete and suppose there is another language C in NP and if we have B reduces to C in polynomial time.

So B is NP complete, suppose there is another language C that is in NP and B reduces to C in polynomial time then C is also NP complete, why is this? Basically, to show that C is NP complete we need to show two things one is that C is in NP and all the languages A in NP reduced to C. So, by assumption C in NP. So, condition one is meet.

(Refer Slide Time: 08:33)



Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \le_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

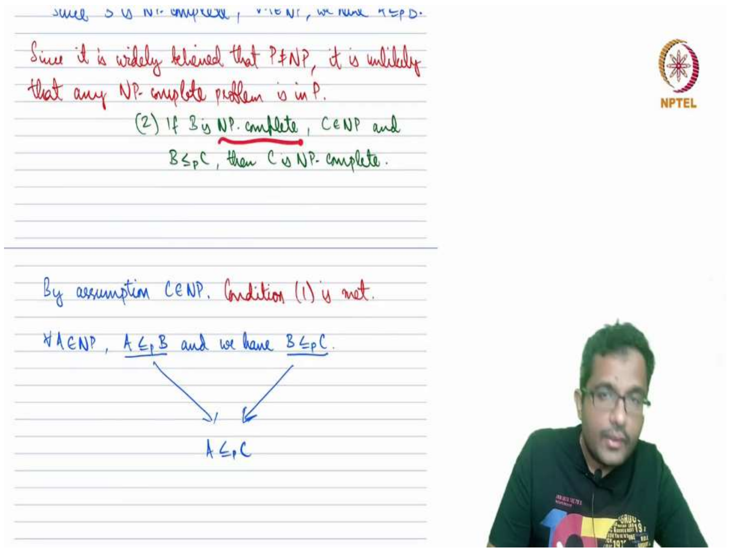
$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$$\begin{tikzcd} A \le_p B \arrow[dr] & & B \le_p C \arrow[dl] \\ & A \le_p C & \end{tikzcd}$$

NPTEL

So, condition one is met now, we have to show condition two.

(Refer Slide Time: 08:52)



Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

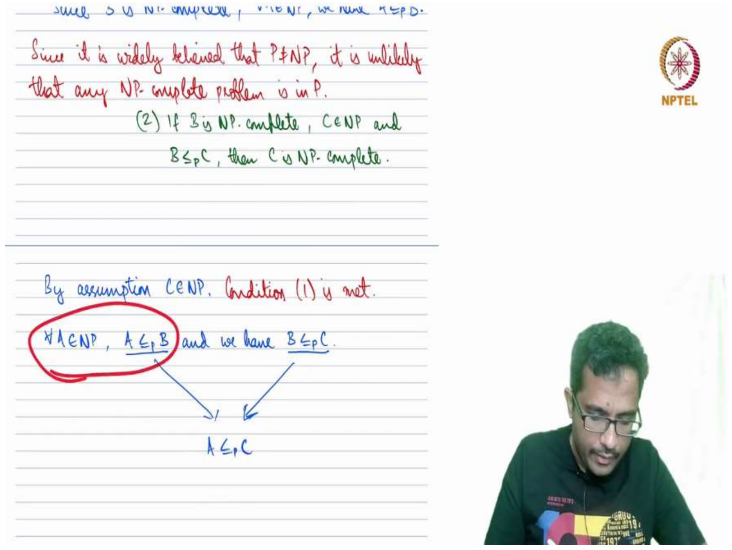
(2) If B is $\underline{\text{NP-complete}}$, $C \in NP$ and $B \le_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$$\begin{tikzcd} A \le_p B \arrow[dr] & & B \le_p C \arrow[dl] \\ & A \le_p C & \end{tikzcd}$$

NPTEL



Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

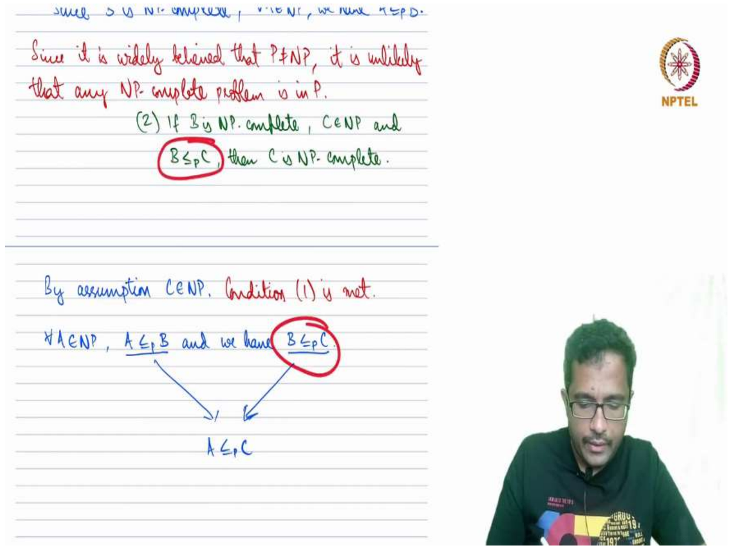
Since it is widely believed that $P \ne NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \le_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$$\begin{tikzcd} A \le_p B \arrow[dr] & & B \le_p C \arrow[dl] \\ & A \le_p C & \end{tikzcd}$$



Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

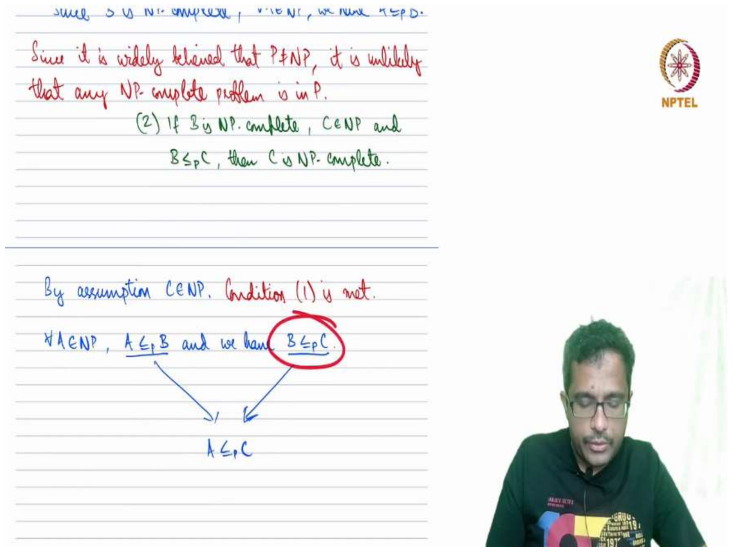
Since it is widely believed that $P \ne NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \le_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$A \le_p C$



Since B is NP-complete, $\forall A \in NP$, we have $A \le_p B$.

Since it is widely believed that $P \ne NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \le_p C$, then C is NP-complete.

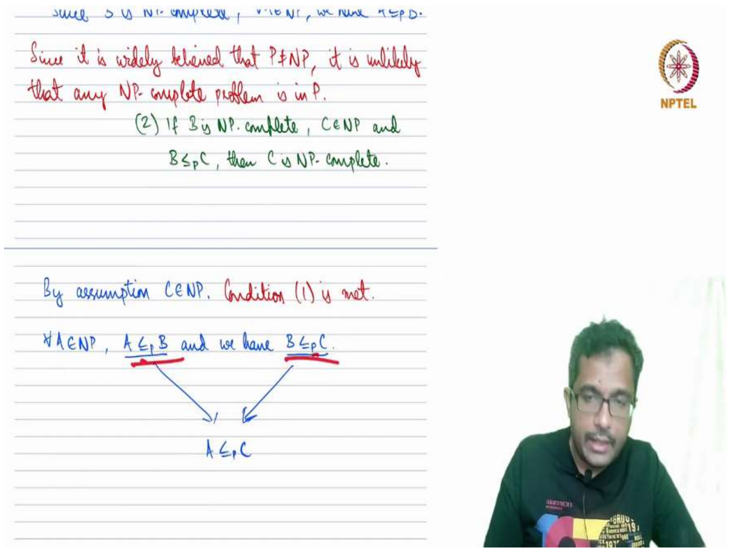
By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \le_p B}$ and we have $\underline{B \le_p C}$.

$A \le_p C$

So, now we have to show for all A in NP, A reduces to C. By assumption again, B is NP complete, B is NP complete. So, we know that for all A in NP, A reduces to B. This comes from B being NP complete. And by assumption we have B reduces to C.

(Refer Slide Time: 09:08)



Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

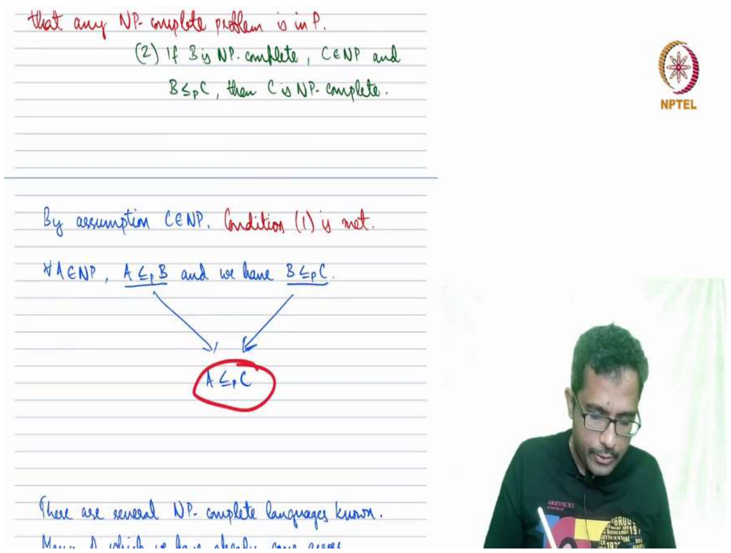
Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \leq_p B}$ and we have $\underline{B \leq_p C}$.

$$\begin{tikzcd} A \leq_p B \arrow[dr] & & B \leq_p C \arrow[dl] \\ & A \leq_p C & \end{tikzcd}$$



that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

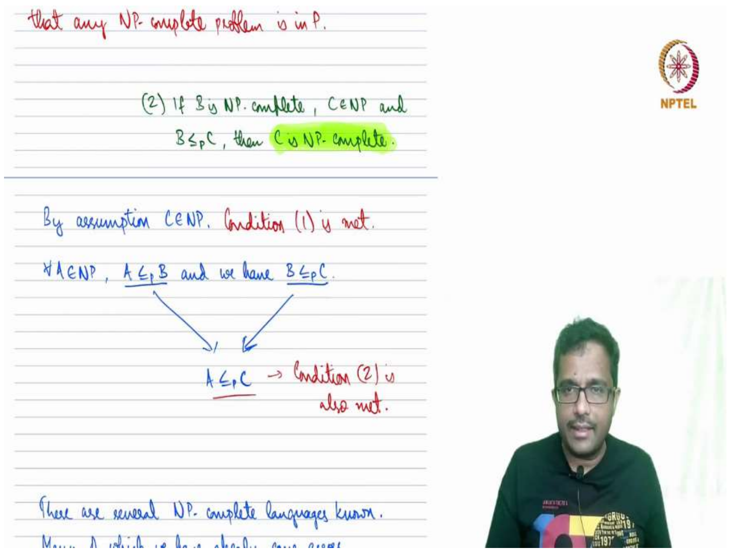
By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \leq_p B}$ and we have $\underline{B \leq_p C}$.

$$\begin{tikzcd} A \leq_p B \arrow[dr] & & B \leq_p C \arrow[dl] \\ & A \leq_p C & \end{tikzcd}$$

There are several NP-complete languages known.

[illegible]



that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \leq_p B}$ and we have $\underline{B \leq_p C}$.

$\underline{A \leq_p C}$ → Condition (2) is also met.

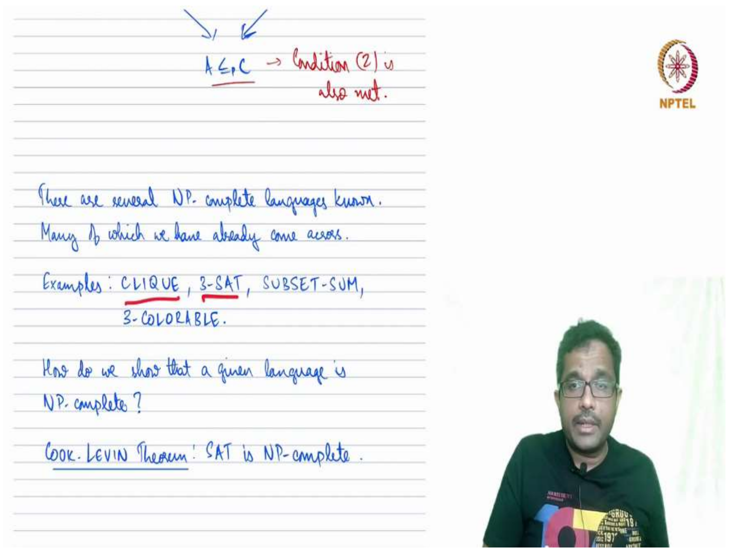
There are several NP-complete languages known.

So now combining these two reductions A reduces to B and B reduces to C. So, last lecture we talked about the transitivity of reductions. If A reduces to B and B reduces C you can like compose this these reduction functions to obtain a reduction from A to C. This implies that A reduces to C. So this shows that condition two is also met.

So, these are the two conditions for C to be NP complete and thus we have C is NP complete. So now this gives us another way to show that a language is NP complete.

We first show that language is in NP and then show that another NP complete language reduces to this. So, that is why this remark is kind of significant. So, now, we know several NP complete languages. So, this we consider to be the hardest problems in NP. And the thing is that many of them we have already seen in this course, even though I did not say that they are NP complete languages.

(Refer Slide Time: 10:32)



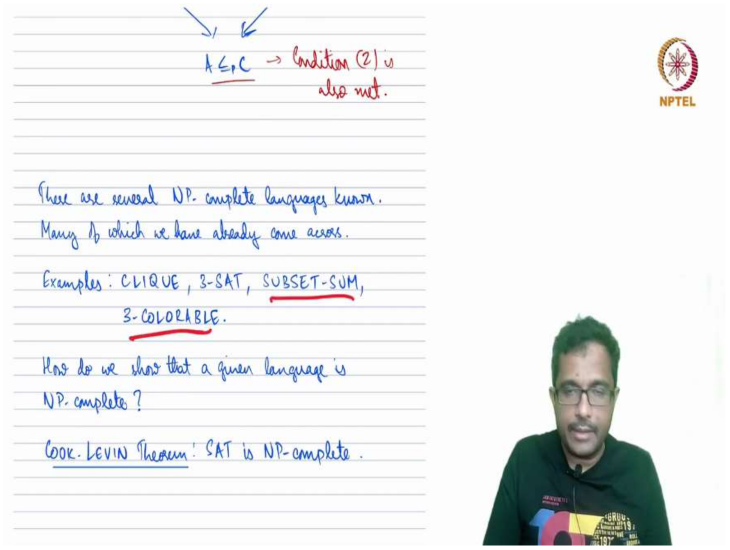
$A \leq_p C$ → Condition (2) is also met.

There are several NP-complete languages known. Many of which we have already come across.

Examples: CLIQUE, 3-SAT, SUBSET-SUM, 3-COLORABLE.

How do we show that a given language is NP-complete?

Cook-Levin Theorem: SAT is NP-complete.



$A \leq_p C$ → Condition (2) is also met.

There are several NP-complete languages known. Many of which we have already come across.

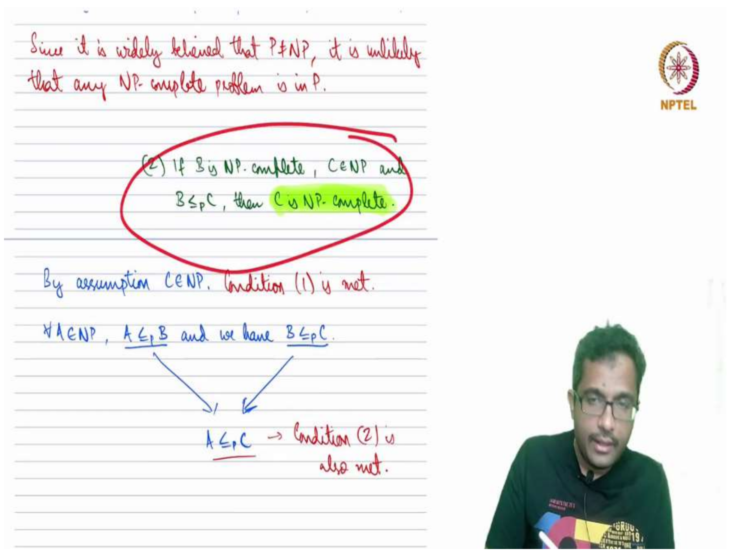
Examples: CLIQUE, 3-SAT, SUBSET-SUM, 3-COLORABLE.

How do we show that a given language is NP-complete?

Cook-Levin Theorem: SAT is NP-complete.

For instance, CLIQUE is an NP complete language. 3 SAT is an NP complete language, SAT is an NP complete language, SUBSET-SUM is an NP complete language, 3 COLORABLE is an NP complete language, all of this we have talked about many times in this course already.

(Refer Slide Time: 10:48)



Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

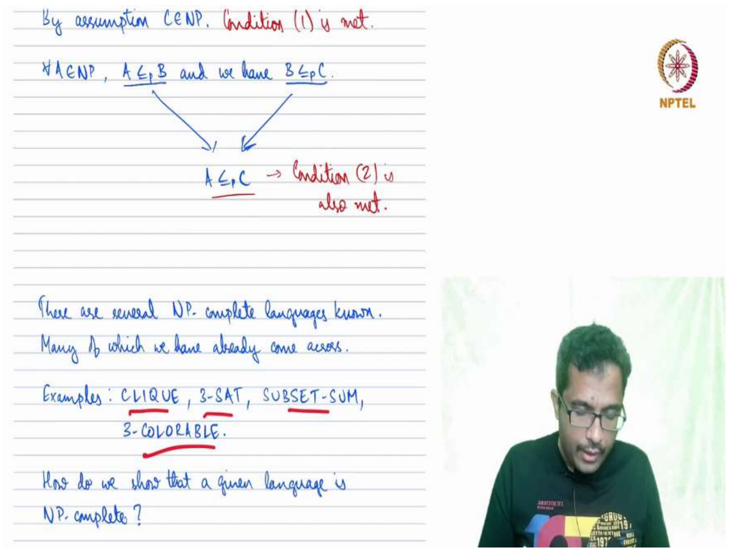
$\forall A \in NP$, $A \leq_p B$ and we have $B \leq_p C$.

$A \leq_p C \rightarrow$ Condition (2) is also met.

Now, how do we show that a given language is NP complete? So, this gives us a way. Suppose we know that a certain language is NP complete. Let us say we know SAT is NP complete. Now, to show that CLIQUE is NP complete, we just have to show that CLIQUE is in NP and show that SAT reduces to CLIQUE.

So, to show that a certain language is NP complete. Let us say CLIQUE is NP complete. We first show that CLIQUE is in NP and then show that a known NP complete problem reduces to CLIQUE in polynomial time. So, this is how you usually show that a language is NP complete.

(Refer Slide Time: 11:26)



By assumption $C \in NP$. Condition (1) is met.

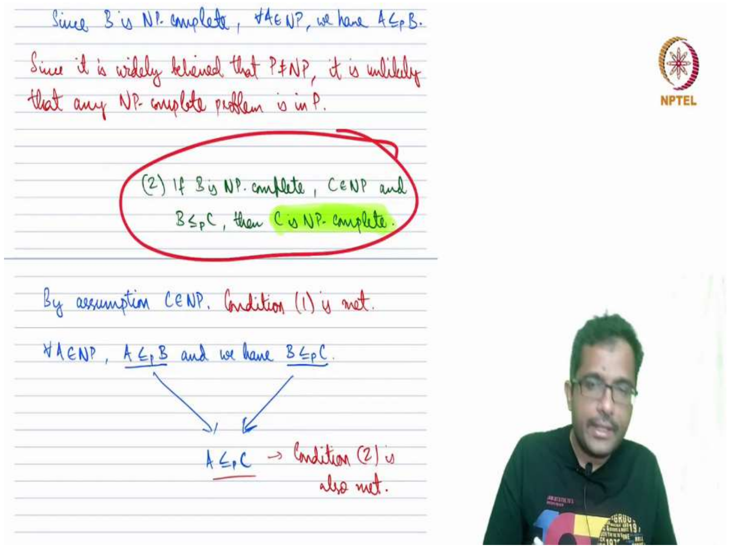
$\forall A \in NP$, $A \leq_p B$ and we have $B \leq_p C$.

$A \leq_p C \rightarrow$ Condition (2) is also met.

There are several NP-complete languages known. Many of which we have already come across.

Examples: CLIQUE, 3-SAT, SUBSET-SUM, 3-COLORABLE.

How do we show that a given language is NP-complete?



Since B is NP complete, $\forall A \in NP$, we have $A \leq_p B$.

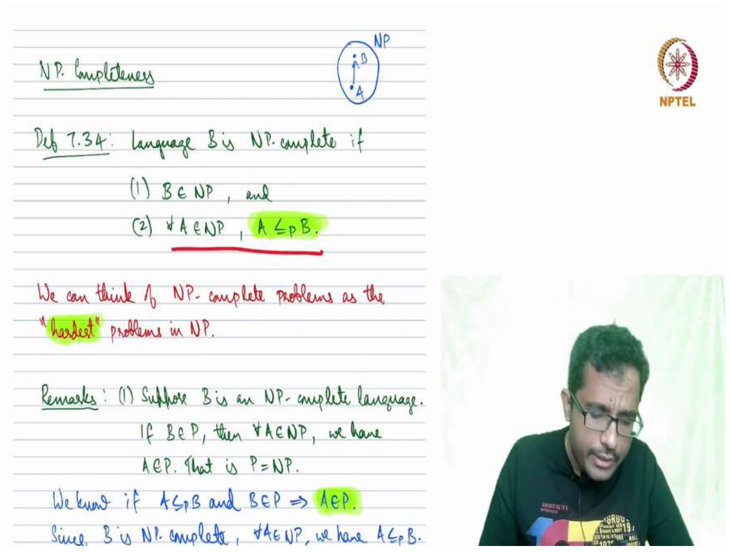
Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \leq_p B}$ and we have $\underline{B \leq_p C}$.

$\underline{A \leq_p C}$ → Condition (2) is also met.



NP-Completeness

Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language. If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

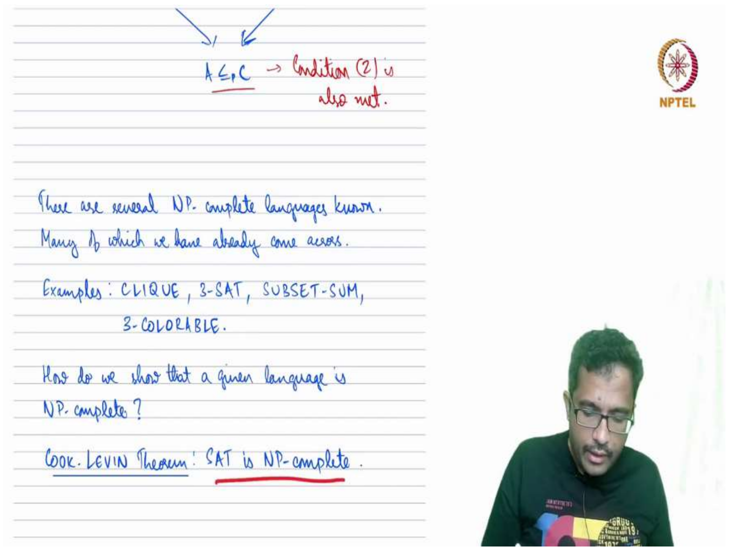
Since B is NP complete, $\forall A \in NP$, we have $A \leq_p B$.

So, like that, we can show all of these languages are NP complete. But then we need something to start with. Some language has to be the first language that we show is NP complete. So, this approach works only if there is an already language B that is known to be NP complete. So, what is the first language that we can say is NP complete?

Because the conditions for NP complete may feel a bit stringent, because we want to show that it is an NP that is not that hard. Usually, this is the easiest step and then we have to show that every language reduces to B. So, how is it possible to show that all the languages in NP reduces to that language.

So, that is not easy or that is not straightforward. So, the first language that was shown to be NP complete was satisfiability, which we saw in one of the earlier lectures.

(Refer Slide Time: 12:24)



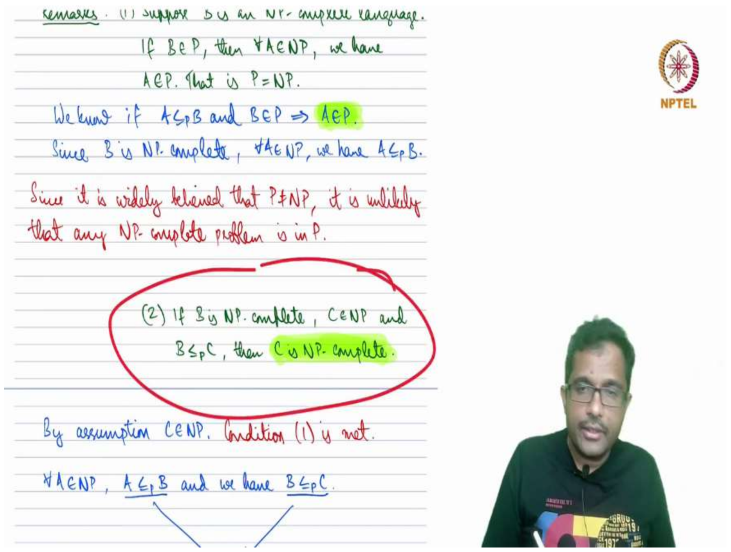
$A \leq_p C \rightarrow$ Condition (2) is also met.

There are several NP-complete languages known. Many of which we have already come across.

Examples: CLIQUE, 3-SAT, SUBSET-SUM, 3-COLORABLE.

How do we show that a given language is NP-complete?

Cook-Levin Theorem: SAT is NP-complete.



Remarks: (1) Suppose B is an NP-complete language.

If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $A \leq_p B$ and we have $B \leq_p C$.

And this result that SAT is NP complete is commonly known as Cook Levin Theory and this was proved in the early 70s independently by Cook and Levin. So, they showed from first principles that SAT is NP complete. Meaning any language in NP reduces to SAT. I think we have already seen that SAT is in NP, but to show the second part that any language in NP reduces to SAT, Cook and Levin showed this and then once we have that SAT is NP complete.

Now we can show other things like we can show that CLIQUE is NP complete by reducing SAT to CLIQUE. We do not need to reduce everything to CLIQUE or we do not need to show that everything in NP is reducible to CLIQUE. Now, we can take a reduction from SAT to CLIQUE and show that CLIQUE is NP complete.

Now and so on and so on, like to show that 3 COLORABLE in NP complete, it is enough to show that 3 COLORABLE is in NP and SAT reduces to 3 COLORABILITY and so on. So, this is how we show languages are NP complete. So, first, we start with SAT, and once we have a bunch of languages, we know to be NP complete, now, we can reduce from any one of them not necessarily from SAT.

So, that is a significance of NP completeness. So, NP completeness are considered to be the hardest problems in NP. And the two most important things is that if a certain language is NP complete, and we show that it is in P, that implies that P is equal to NP. And second, if a language is NP complete, suppose B is NP complete and B reduces to C and C is also in NP this implies that C is NP complete and this is used to show the NP completeness of languages.

And that is about NP completeness. In the next lecture, we will show the NP completeness of SAT. In other words, we will show the Cook Levin theorem. And because that is kind of a longish proof, I did not want to continue in this lecture. So, see you in lecture number 50. That is all from lecture number 49. Thank you.