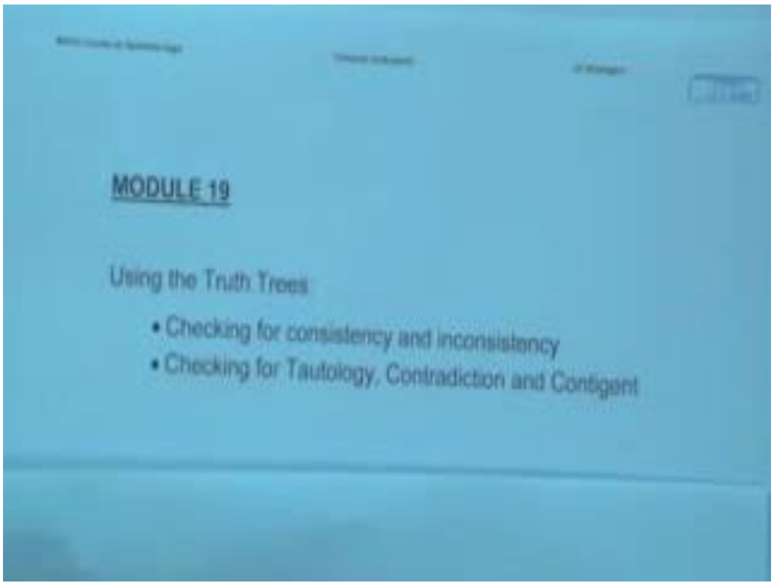
Symbolic Logic
Prof. Chhanda Chakraborti
Department of Humanities and Social Sciences
Indian Institute of Technology, Kharagpur

Lecture - 19
Using the Truth Trees
Checking for Consistency and Inconsistency
Checking for Tautology, Contradiction and Contingent

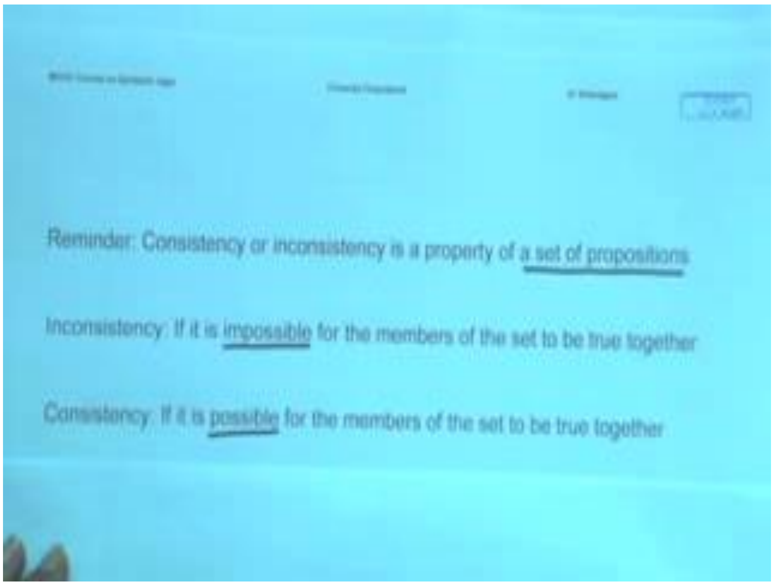
Welcome back! To module nineteen of symbolic logic NOC course. So, we are learning the truth trees and we are going to follow up on the various tasks that we want to present to it.

(Refer Slide Time: 00:36).



So, I hope by now we know how to construct the truth trees and we have been discussing that in the previous modules. Now, it's time to pose some questions and get the procedure to do certain things for us. So, just like in the truth table, remember, truth table we first learnt the table and then we asked the table if it can give us certain answers. So, here with the truth trees, again we are going to ask whether it can show us for example, consistency and inconsistency of certain set of propositions and whether it can also classify given propositions or statement forms as tautology, contradiction, contingent and so on and so forth. So, today's task is to learn how the truth tree can help us to do these two tasks. One is about consistency, inconsistency and other is about classification as per their truth values.

(Refer Slide Time: 01:52)



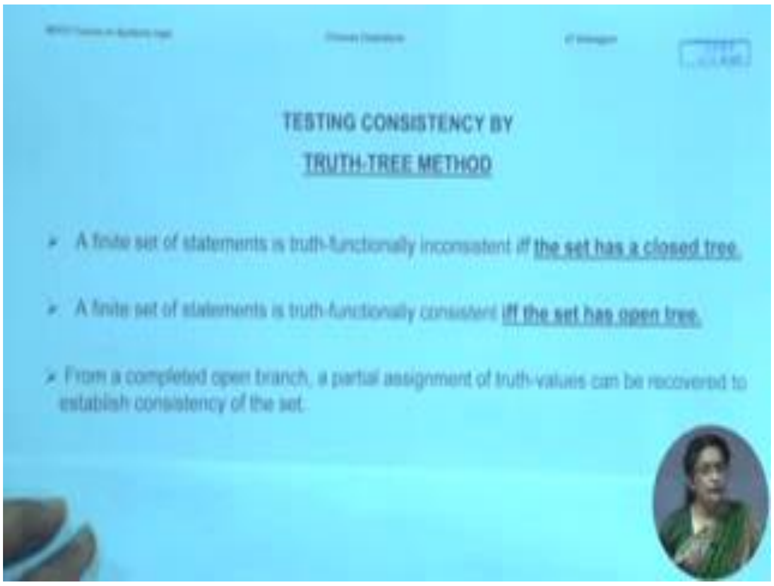
So, that is what is on the agenda for today on module 19.

Just to remind ourselves what consistency, inconsistency entails. You may remember that it is actually a property of set of propositions. So, given a set of propositions, we can ask whether the set is consistent or inconsistent. And if you recall, then the set is inconsistent if it is *impossible* for all the members of that set to be true at the same time. In a simultaneous manner or in a compatible manner, whichever way you understand. But there must not be even a single possibility when all of… all the members are going to come out is true. That is when we call the set inconsistent. Remember that. And then consistency is just the opposite of that; namely, it should be possible for the set members to be true together. How many such possibilities? The answer is always *at least one*. So, minimally speaking, the set is going to be consistent if there is even one chance when all the members of the set is coming out to be true.

Now, if this is the situation in… in what we say is the property of a set of propositions, then how can the truth tree show that to us? So, in that I will again remind you that when you are doing the truth…remember this is what I have said again and againand I will repeat it again, is that whenever we are doing the tree, as you are listing a proposition in the tree, whether in the root or whether in the branch, *you are claiming or assuming it to be true*. So, given that, if the set is inconsistent, you know, there must not be, there cannot be even a single branch when you are going to have all the members of the set to be true. Alright? Because, if the set is inconsistent then the members cannot be true at the same time. Therefore, if you do the tree of this kind of a set, there is not going to be even a single branch when every member of that set is going to be true. So, we will utilize this insight to fill the definition of inconsistency in terms of the truth tree.

Similarly when you are looking into consistency, we are going to look into the completed open branches whether there is even one. And if there is one completed open branch, it tells you…that completed open branch when you follow that, you are going to find the truth conditions under which the members of the set of the propositions are going to be true. So, if the set is consistent, you are going to see in the tree *at least one completed open branch* of this kind. Keeping these together, now we can define what consistency and inconsistency the true truth tree method is going to be like.

(Refer Slide Time: 04:54)

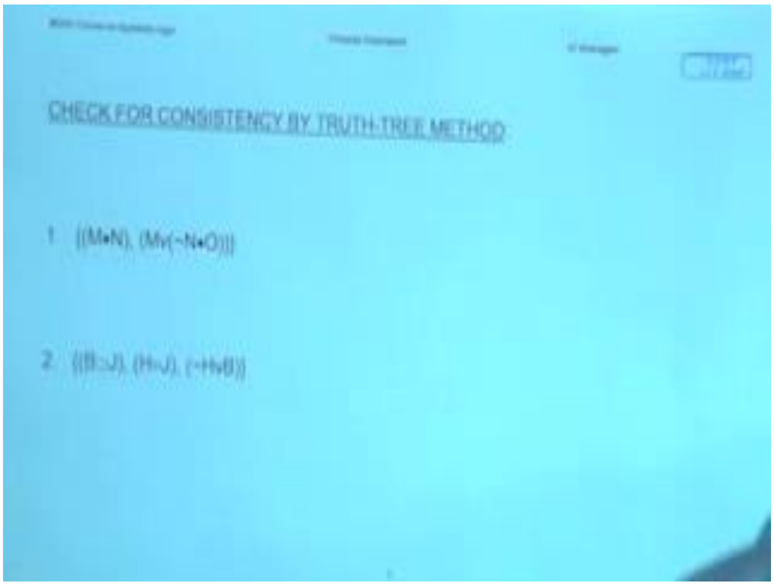


So, we will define it like so. That a finite set of statements is *truth-functionally* inconsistent *if and only if* the set for the statements has a closed tree. We will remind ourselves, closed tree means a tree which has no open branches. All its branches are going to be closed. So, there is not even one possibility where you will find the truth conditions for the members of that set of propositions. There will not be any situation when you are going to have the members of the propositions… the set to be true at the same time. So, this is what we are saying when we said the set is going to have a close tree.

On the other hand, as we said, if the set happens to be consistent, then the test for that in truth trees going to show what? That the set has an open tree. Remind yourselves that having an open tree means that there is going to be at least one completed open branch in that tree. Right? And that should settle the question whether the set is consistent or not.

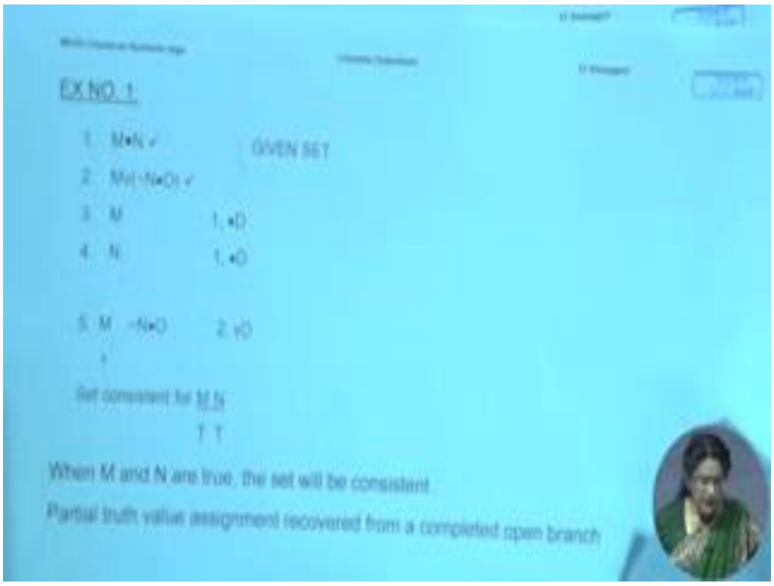
So, if you have understood the basic concepts and then if you have understood how the tree works, then it's obvious that this is how you are going see it in the truth-tree making. And remember that, if you have an open tree, there is going to be at least one completed open branch. And I remind you that from a completed open branch you can recover partial truth value assignments for all the components in that tree which makes the set all the members coming out to be true. So, whenever you see a completed open branch, try to recover the partial truth value assignments. And that will tell you what conditions, under what conditions the propositions are all going to come out to be true. Let's take a look into an example so that we can… we can see this.

(Refer Slide Time: 07:08)



Again, I would request you that you keep your pen, pencil and paper ready, so that you can do this along with me as we do this. Here is the first problem. So, this is a set of two propositions. One of them is $M \bullet N$ the other one is $M \vee (\sim N \bullet O)$, except that there is a parenthesis around $\sim N \bullet O$. So, two membered set. Is it consistent or inconsistent? We are going to find out. And the other set has three members. So, $B \supset J$, $H \equiv J$, $\sim H \vee P$. These are the three members and our question remains the same, whether this is a consistent set or an inconsistent set? And just do the tree, the tree should speak to you. So, I will ask you… I will give you little bit of a lead time to start at least on the first problem. Why don't you try this while we try to set it up? And then we will actually match our results, so that it is easier for you to grasp where you may have gone wrong or where you are correct. I am sure you will be able to handle this by now. But let's see.

(Refer Slide Time: 08:31)



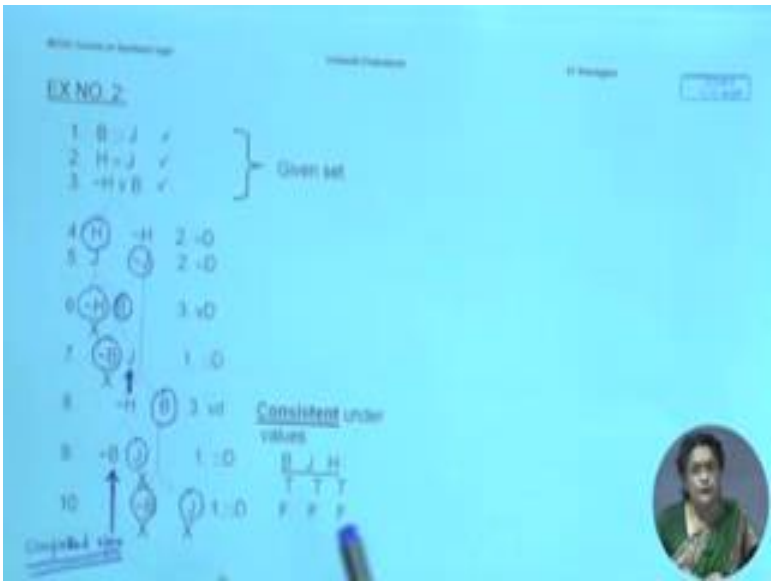
Ok? So, the first tree. We are going to set it up first. Remember the given set has only two members. M • N, M ∨ (N • O) and I will remind you that whichever one you are trying to decompose, you should know *why* and remember to put a tick mark against that just to tell yourself that this one is decomposed already. So, you probably thought that this one is not bifurcating and this one is. So, let's do number 1 first. So, if you have done it, then this is the result of your decomposition on line number 1 and this one gets a tick like so. After that comes this, and then we have decomposed it like so; and the answer is the justification is given here also.

Please remember to put tick against decomposed propositions. Now you ask yourself whether we have found a completed open branch at this stage or not? And I had asked you to look both in the left branch as well as the right hand branch. And the answer is that on the left hand branch, that is a completed open branch. How do we know that? Every decomposition that was supposed to have been done on this, is done already and all that is left on this branch are literals, nothing but literals. So, on this branch there is nothing more to be done. Fine? If that is the case you have found the answer here. You have found an answer, because all you needed is at least one completed open branch in order to know what kind of set you are dealing with. So, this is the completed open branch. On that basis you can call the set consistent.

Now, your question is: Do I need to decompose this further? The answer is: It is not necessary. Because, the answer that you wanted to get is made obvious by this branch already. You have found this completed open branch and that should settle the question. So, on top of that if you want to do that, that is separate, but that would be redundant. Even if it is open, you know the answer; even if it is closed, you know the answer. So, at this point that's where we will stop the tree.

Now what comes is this that from this completed open branch not only we can claim that the set is consistent, but we can also tell under which truth conditions, this set is going to come out of true.So, let's take a look. You are falling through and the literals are M and N. Which tells us when M is true and N is true, this is going to come out as true and this also is going to come out as true. Use your knowledge of the connectives to see why that must be true. So, we will just write this small truth table right from the completed open branch. This is nothing but recovered truth values from the completed open branch. So, this is our answer that whenever M and N are true the set will be consistent. And this set is shown by truth tree method to be a completed open… sorry a consistent set. So, this was our first problem. See, I mean it's not difficult.

(Refer Slide Time: 11:56)



Let’s try the next one. So, now, we are on the example number 2 which has three members, and we have B ⊃ J, H ≡ J, ~H ∨ B. Now this is slightly a different situation where there is no clear choice. It seems to you… it is may seem to you that there is no clear choice where to start. Because each of these , if you recall the decomposition rules then each of these is going to generate bifurcated branching. So, before you start on, it is better to ask yourself which one should I start with and why? By now you should know the results of the decomposition. If you do on any of this you should know what you are going to yield as the branching out result.

So, overall I would say that the line number 2 is the most probably… time-taking one in the sense that it is going to generate two lines. So, if you leave it postponed for any subsequent steps, then you have to repeat the result on every open branch. So, instead of that, let us get it over with. So, we are going to do it like so. So, line 4 and 5 both are by triple bar decomposition (≡D) and we need to write the rule also twice. Remember you now you have generated two branches, and please remember to put a tick on the decomposed line number 2. Now what ? Let me remind you that I have asked you to hold one branch, while you are doing the operation on the other. So, for example, if you are going to now tackle the left hand branch, keep the right hand branch on hold. Here is the result of line 3. This is the bifurcated branching result. So, under these we have now put ~H, B. Why we chose it? Because apparently it is implied we can generate a closed branch, because we knew that if we… if we branch it then we are going to have ~H and here is H. So, we can close it down.

And, but this one is still open. But this is line number 3 $\vee$D. We put a tick on the decomposed line. What is still left? Now line number 1. So, now, under these we are going to repeat line number 1 and its result. So, here is a result of line number 1 decomposition. Again you find that here is ~B, J. When you find the ~B, you know that this branch is going to close. So, this is closed, this is closed, but please note this one is open.

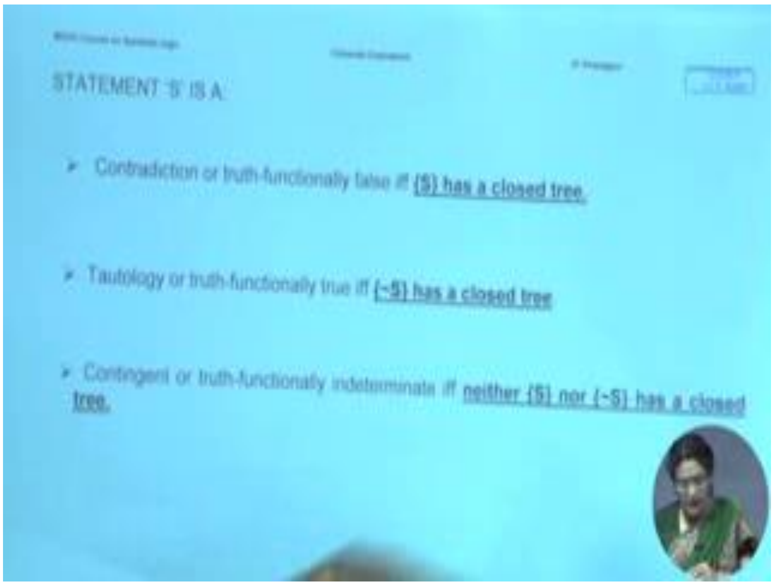
At this juncture, you can ask yourself : Is this a completed open branch or not? These are closed, but is this completed? Completed open? The answer is yes. Every decomposition that you had to do is over at this stage. Right? And the branch only contains now literals. So, you have found at least one completed open branch, which makes the set consistent. And you can recover the truth value, the partial truth values from this completed open branch.

The further question is: If I have found the completed open branch by just by doing the left hand side, do I need to do the right hand side? The answer is: Not necessary. Because, the question that you had in mind can be settled by just this result. So, here you can say that when whenever J is true, B is true, H is true, the set is going to come out as consistent. And that will be a perfect answer to you.

But if you want to, you can always do the right hand side. And here the answer is going to be like this that we have to… we have to repeat the decompositions one by one on here as branches happen. And you may find that this would close down, this would close down, but here is the ~B , ~H, ~J and that branch is completed open. Everything that needed to be done as far as decomposition is concerned is finishedand everything is now reduced to literal. And it would give you another set of truth values under which the set would come out as consistent. That would have to be whenever B is false, J is false, H is false. So, either of these values would settle our question that the set is consistent.

I have to remind you now that you know it's very important as you can see as you are advancing in your tree-making that it is important that you circle, encircle the literals which is giving you the closed branch so that you remember, so that your search is complete and you can see, follow through every branch to sort of get the answer quick. So, that's one thing. The second is that search for not just the closed branch, but also the completed open. In these cases, completed open will give you a lot of answers, logically important information. So, we have found that this set is consistent under this kind of values. Either this or this would make the set consistent.

(Refer Slide Time: 17:33)



Let's move on to the next task, which is whether we can categorize, given a statement whether we can categorize it as tautology, contradiction, or contingent. Now in truth table what you did is to look into the final truth table and see what kind of truth values are shown there. That option is not here. In terms of truth trees we have to understand whether the statement is of this kind or that kind. So, I will again…I will… we have to fall back upon our understanding of the truth trees, and I will remind you that the truth tree lists only truth conditions. So, given that, now you try to think about a contradiction. A contradiction is that kind of a statement or statement from which only has false substitution instances.

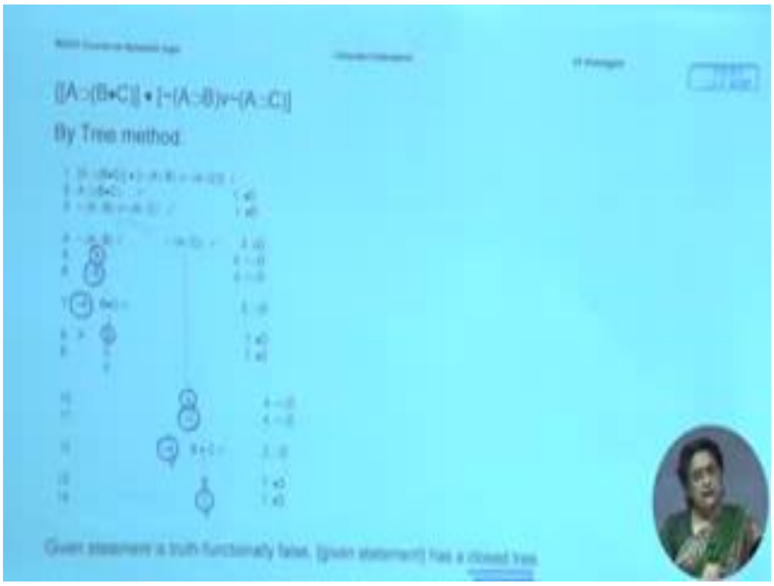
So, given that kind of a proposition, what will happen is that there is never going to be a situation when this statement is going to come out as true. So, if you are doing a tree of that kind of a statement, which is always false what will happen? That branches will close down. Because whenever you have a completed open or open branch what you what you can see is when the statement is going to be true. But if the statement is of that kind where you do not have such truth conditions, you are going to have a closed tree. So, the on that basis using that insight, we can now define it like this : That you can… if you have statement S and if you do the tree on the statement S, if it is a contradiction, what will happen? The set of S is going to have a closed tree. Alright? So, we can put it as necessary and sufficient condition also. If our proposition or a statement has a closed tree, it is contradiction. If it is a contradiction, it is going to have a closed tree. Both are true. So, this is our result for contradiction. And we will we will take an example to see that also.

On the other hand, tautology proof is not simple. And you may.. you may think or read up on this, but there is no direct proof of tautology as such. Rather in terms of truth trees what we can do is to see whether the negation of a given statement has a closed tree or not. Why? Remember if the proposition S is a tautology its negation is going to be a contradiction. And just now you have found out how to determine a contradiction. So, our modus operandi will be like this: that take a random proposition. If its negation results in a closed tree, the original proposition must be a tautology.

That is indirect proof. That is indirect proof. Why we cannot establish directly? There is a lot of theoretical discussion in that. But we will not get into that, but try to understand the logic behind what we are doing. We have already established how to show a contradiction and we know that the negation of a contradiction is a tautology. So, that is… following that we will say that if the negated statement results in a closed tree, original statement must be a tautology.

On the other hand, the third category contingent or indeterminate. What will happen is that neither the tree for the proposition will be a closed tree, nor the negated proposition is going to have a closed tree. So, this is how we will find out our classification of proposition.

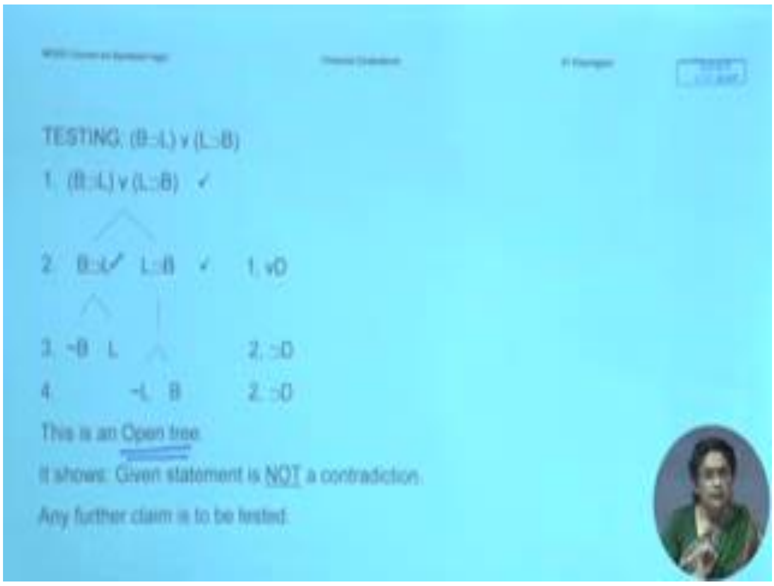
(Refer Slide Time: 21:42)



Let’s take this example. For example, you have [A ⊃ (B • C)] • [~(A ⊃ B) ∨ ~(A ⊃ C)]. It’s a single statement. It’s a compound statement, it’s single statement. Let’s see, let’s directly do the tree on this and let’s find out what will happen. So, I will let you work on this, but if you do it properly and without getting nervous or anything, or without getting into mistakes, you know, then the tree is going to result like this. What you will find, no matter how you do it, is that every branch is going to close down. You are going to have a literal and its negation in every single branch. Can I stop after doing this on the left hand side? The answer is no. Why? Because, we do not know just by showing the left hand branch whether the tree is going to be a closed tree. Closed tree means every single branch is closed. Not just some branches are closed, not just at least one branch is closed. Right?

So, we need to exhaust the possibility and if you do this, this is the result that you are going to see. Again reminding you that you need to encircle the literals which cause the closed tree. So, your result is going to be like this that it’s a truth functionally false or contradiction, because the given statement directly when it is done it has resulted into a closed tree.

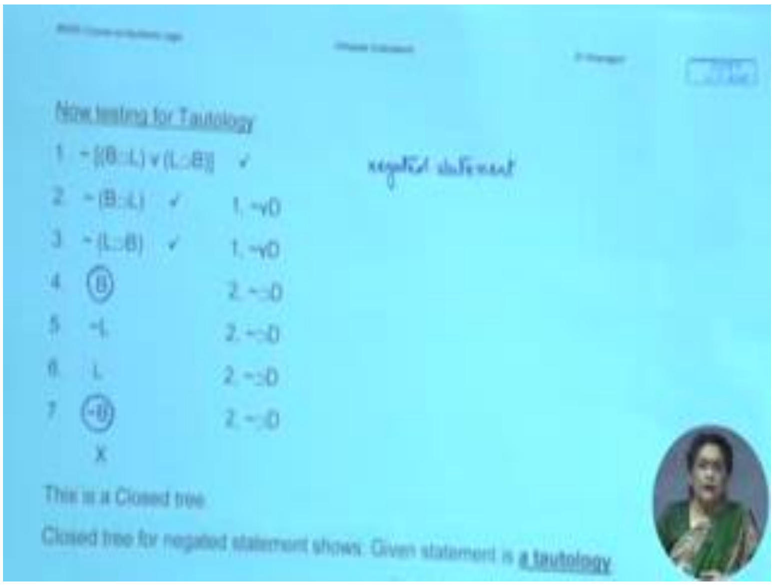
(Refer Slide Time: 23:24)



Let's try this other one. This seems like a simple problem. This is (B ⊃ L) ∨ (L ⊃ B). Now given this, you have a choice. You can do a tree directly on this one, or you can do the negation of the…, you can take a negation of the proposition and do the tree on this. Either way you will have to find out. There is no set algorithm to tell you which one for what kind of proposition. So, let's try this. So, here is if we do the tree directly on this. And this is simple tree. So, very quickly you can do it also. You are going to generate two branches like so.

When you have done that, let's work on the left hand side keeping the right hand side on hold. So, because there is still this to be decomposed on this side, so not-B, L and this one gets ~L, B. Take a look into this tree. What you have is every branch is open. But at this stage what do you know? The simplest and the most logically correct answer is that it is *not* a contradiction. Can I quickly say that this is a tautology? No, for that you have to do the other test after this, namely, you have to negate the proposition and see whether it is a tautology or not. At this stage you are only entitled to say that it is *not* a contradiction.

(Refer Slide Time: 25:04)



So, let's do this negated tree. This is your given statement. That here is a negation, and here comes the tree. If you are doing it correctly, please remember to tick or check the compound statements as you are decomposing them and here is the result. Now after this, all the decompositions are done, and please follow the branches through and you will find there is this ~B as well as B. There is also L and ~L. So, either way the branch closes down, right? When that happens on the negated statement, you are in a position to say this has to be a tautology.

So, that would end our lesson for today, that we have just learnt some tasks by truth tree. One is how whether the given set is consistent, inconsistent. We also saw whether the given proposition is tautology, contingent or contradictions.

With that, we will finish this module and will continue with more on truth trees in the next module. Thank you very much.