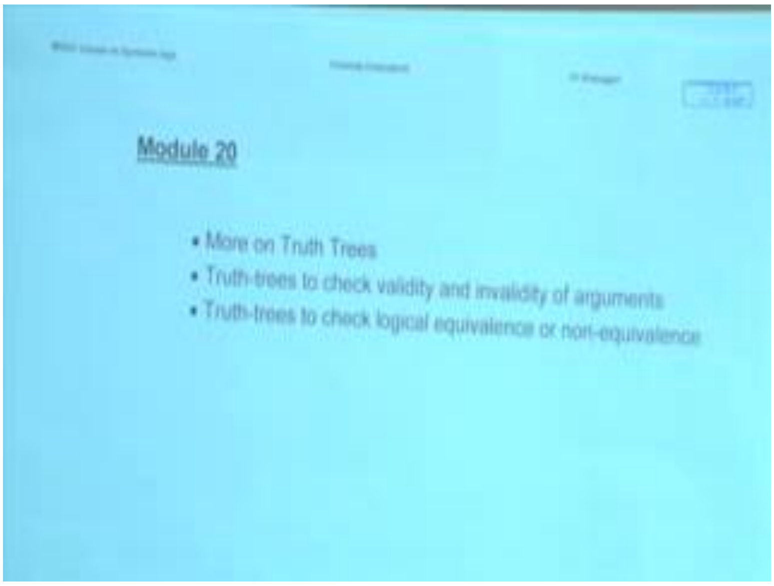
Symbolic Logic
Prof. Chhanda Chakraborti
Department of Humanities and Social Sciences
Indian Institute of Technology, Kharagpur

Lecture - 20
More on Truth - Tress
Truth – Trees to check Validity and Invalidity of Arguments
Truth – Trees to check Logical Equivalence or Non – Equivalence

Hello. We are doing truth-trees these days so you have back again with little bit more on the truth-trees.

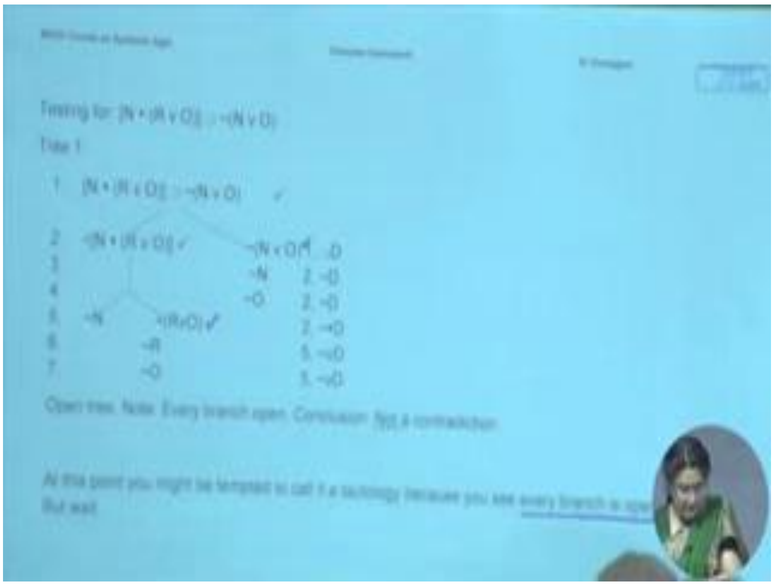
(Refer Slide Time: 00:30)



And this would be our last lecture lesson on truth-trees so we are trying to finish the tasks that we can ask it do. In the last module we were looking into the categorization of propagation. So, we will be speaking a little bit on that also by truth-trees. And then today we are specifically going to look in to the validity and invalidity determination.

So, arguments are they valid or invalid can we find that out by using truth-trees. And the last task would be whether given to propositions the truth-tree is able to tell us whether this is logically equivalent or not. So, our task for Module 20 is like this that we will be doing follow up on the earlier module a little bit and then go in to two more tasks.

(Refer Slide Time: 01:25)



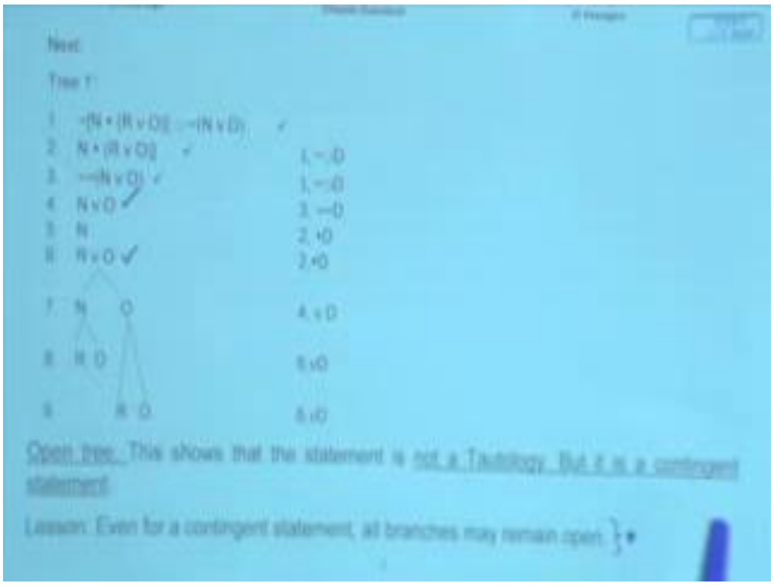
I said that we need to do a little bit more on the categorization because I do not know whether you have captured or you have grasped the truth-tree procedure completely by now. And it may be very same thing for you to infer what it can do or how to read the tree. In order to correct any kind of misconception that there might be in your reading of the truth-tree I thought that this kind of an example or this discussion bit help you little bit. This is connected to our module 19 discussion on the categorization propositions into tautology counter decision contingent.

Here is an example where we are doing trying to do a truth-tree determination, what kind of a proposition this is. So, here is N dot R wedge O horse shoe tilde N wedge O. And if you take this proposition and try to do a tree write on that. So, remember to tick it and it will give you this kind of a result because it is a horse shoe proposition. You have two branches; again I will remind you that unit to hold one of the branches while you are doing the operation on the other. For me I always work on the left hand side branch first except to you, but if you are doing that, if you what to reverse the process then let us do the right hand branch this time.

So here is this, what we have done is this one we have the decomposed into this. And here is the result at the left hand side and so on. This one is decomposed and we have resulted in this, and this is the result of decomposition of this one, and this is the result of decomposition of this one. What you see? You see completed open branches everywhere. Every single branch of this tree is a completed open branch. So, you could have stopped right here also and it will give you an answer, but what is that answer that you have obtained at this point. The logically correct answer is that it is not a contradiction, because the tree has not closed down. Even just the small branch would show that.

Are you entitled from this to jump and infer that since every branch is opened it is got to be a tautology, because what we have said is that if it is a contradiction all the branches are going closed down and you going to result into a closed tree. Now, with that it might seem to you that therefore in case of tautologies what will happen every single branch is going to remain open. Is that how safe is that, how correct are we in inferring this. And therefore, it is important to see that what would be the result if we get this proposition and do the tree.

(Refer Slide Time: 04:58)

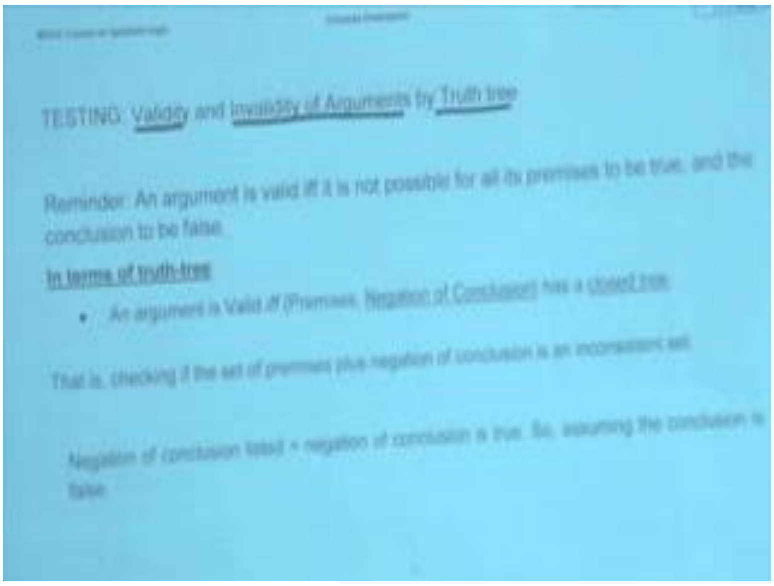


So, here is the negated proposition. There is tilde in front of this and you can now decompose it like this. Remember to put tick on the once that you are decomposing. Here comes slowly the decompositions of the other things as we are going along. So, this is tilde tilde decomposition and then comes the second line decomposition, this is the line that you are decomposing. After that comes the necessary branching. And here is your line number 4 wedges decomposition. Then comes the 6 namely; this is line number 4 this is line number 6 this is what you have decomposed and the result looks like this.

Now tell me are we complete, are these first of all open branches, are these completed open branches and we see that they are. Each one of this is also open completed open branches. What is that tell you? Remember the same one when you did directly on the tree it has not closed down, every branch was opened. Had it been a tautology what did you have seen? The negative proposition would also result into a closed tree, it has not. It is therefore you are entitled tree it is a contingency statement, because neither the tree on the proposition itself as resulted in a closed tree nor the tree on the negative proposition is resulted into a close tree. So, this is a legitimate contingency statement.

But what is the lesson to learn here, that even for a contingent statement as you can see every single branch may remain open. As you can see in the first one that when we saw this is the proposition, this was the scenario what you had earlier. So, given this let us not quickly jump into any conclusion until you have done the further test. At this juncture all you are entitles to say is that it is not a contradiction period. This test further proves whatever result you want to give in this case in turns out be a contingency statement.

(Refer Slide Time: 07:54)



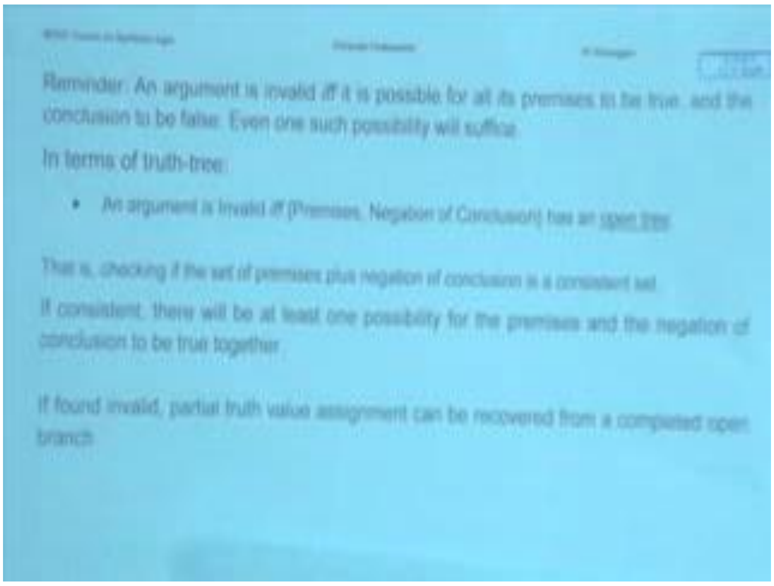
So, this I hope will be helpful lesson to remember by and then now we are going to move into our new task for today namely; validity and invalidity of arguments by truth-tree. We are going to solve the problem of whether an argument is valid or invalid by doing the truth-tree on the argument itself. Let us remind ourselves when we call an argument valid, when it is not possible, all it is premises to be true, and it is conclusion to be false.

In terms of the truth-tree what you will do, we will somehow reduce this task into a consistency checking. What we will do is this that we will try to form a set with the premises and negation of the conclusion. Once more, the premises are given the conclusion is given, for the truth-tree what you need to do is to form a set where the premises are the members and the negation of the conclusion is also a member. What are you saying, what are you checking here, whether the premises truth is consistent with the truth of the negation of the conclusion.

What does that mean? What we are trying to see is whether when the premises are all true whether the conclusion can be false or not. So, whether it is consistent to claim the premises are true conclusion is false. If that is not consistent then you are going to have a closed tree which means that there will not be even one situation when the premises are true and the conclusion would be false, which is exactly what we need to establish when the argument is valid. If it is invalid there will be at least one occasion when the truth of the premises and the falsity of the conclusion will be consistent. This is going to be our modus operandi for this. Only thing that you need to sort of get hold off is that when you are listing the negotiation of the conclusion in your tree, you are claiming that the negation of the conclusion is true which means what the conclusion is false.

So, with that we are going to now set up the truth-tree to for this task. Remember that negation of the conclusion is not given; you have to add a negation to the conclusion and list it yourself in the truth-tree. So, that is going to be an addition to your truth-tree. Let us see an example so that we can talk about it, but before that the invalidity let us go over that.

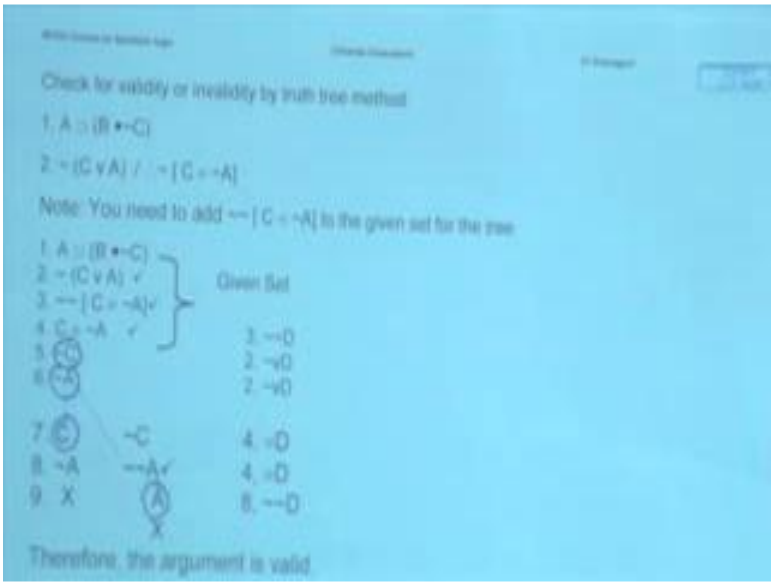
(Refer Slide Time: 10:30)



So argument is invalid when there is one possibility, even one possibility is there where all the premises are true but conclusion is false. In terms of the truth-tree what will happen that the same set the premises with the negation of the conclusion will be an open tree. There will be at least one completed open branch where their consistency of this set is shown. So, this is our way to show arguments validity and invalidity.

And remember that if you find a completed open branch from that branch you can recover partial truth value assignments to show when the arguments would be invalid in this case. So, the completed open branch will do the job of that particular row in your truth table where you show the premises are true and the this kind of situation that conclusion is still false. Let us take now the example so that we see this happening.

(Refer Slide Time: 11:40)



Suppose we have this problem in front of us. So, these are two premises and here is the conclusion, so what you are going to do now line number 3 in these trees is going to be the negation of the conclusion. This whole thing is your conclusion, tilde C triple bar tilde A. When that is negated what you are going to have is tilde tilde C triple bar tilde A and that has to be added to the given set for the tree. So now, you are checking whether one two and this new statement whether that is going to be a consistence set or inconsistent, whether it is going to have a closed tree or opened tree. And accordingly you can tell whether the argument is valid or invalid.

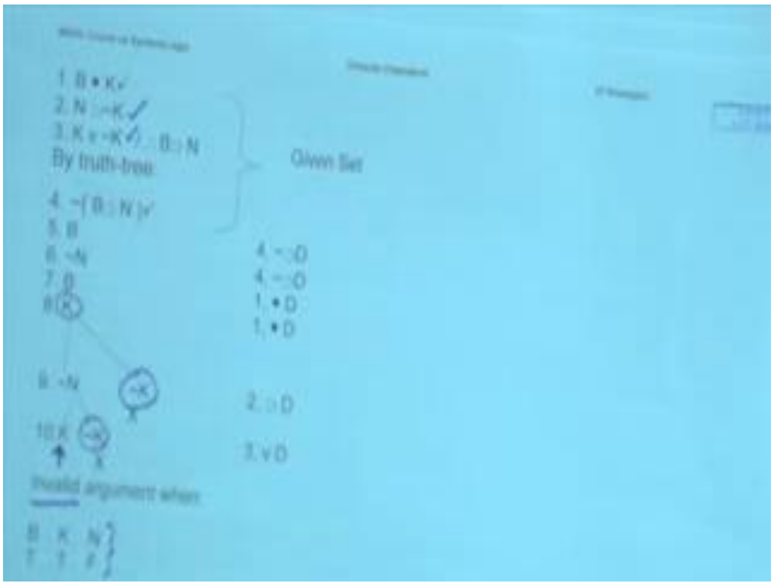
So, here is the new set. These are two premises and here is the negative conclusion, rest is as usual you will do the tree has as usual so here is our first decomposition tilde tilde tree. The movement you do it remember to tick it on and then here is the solution of line number 2. We see some potential possibility of closing quickly so we will follow that. So, this is your line number 2 tilde wedge decomposition.

And then comes this, this is your triple bar D decomposition and we see that when decomposed it will give me something like this and right here you can see C and tilde C will result into a closed branch. This is tilde C tilde tilde A which when further decomposed with tilde tilde D decomposition gives you A and here is not A, so the branches are also stopped. Branches are closed down therefore the argument is valid. You just showed that this set cannot be a consistent set. You just said that it cannot be when the premises are all true the conclusion can be false or the negation of the conclusion can be true so that shows the argument is valid.

This is how we show validity of an argument not a single branches open. Now if you closely looking into the tree as you can see that it has closed down even before you have done line number 1, true. Because we chose those statements first which we thought would result into quick closure of the tree. And line number 1 was not even necessary. So, there is a lesson write there that when you are doing the tree it may be so that you can get the results. See here the result is already given. Had there been any open situations then you still needed to do number 1, but right here you have closed it all, you have you generated branches that are automatically shut down.

So, when you have obtained the result already have derived the result there is no reason why you still have to do line number 1. There may be such trees where every single given statement need not be decomposed. That is the further reason why ask I say that you need to do trees strategically and with the little bit of an open eye.

(Refer Slide Time: 15:22)

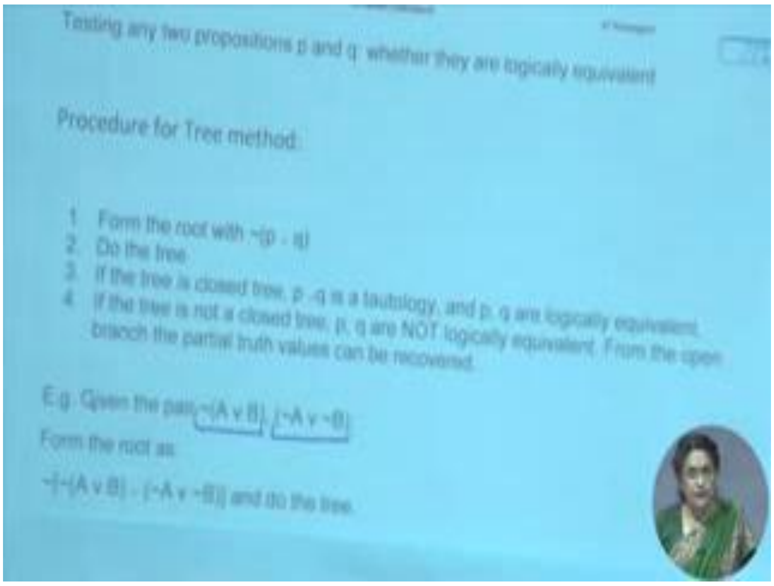


Finally, we come to this one; let us do another example before we leave the topic. Here is an argument, D dot K N horse shoe not K K or not and therefore B horse shoe N. And by now you know how to set the tree. How do you do that? Line number 4 is going to be negation of this conclusion. So this is what you are going to add to the tree, now this is your given set. You do not include the conclusion; you include only the negation of the conclusion. And the rest is the tree as usually, so please will free to do that you do not have to do it as I have done, but please finish the tree.

This is decomposing on line number 4 and then we do it on line number 1 and then comes the branching this is line number 2 and we put a tick mark right here and then already we have generated closed branch. This one is still open so, we do a little bit of decomposition line number 3 on this one. And when you have done that you find that this one would closed down K N not K both are here, but this is a completed opened branch. From that we can now say since we have at least one completed open branch in this tree the argument has to be invalid. And not only that we will be able to tell when the argument will be invalid. How do we do that? We recover the truth values from here, you know what value K has to be because it is listed it as to be true so that we can write here true. This is tilde N because it is listed therefore N has to be false and then comes B, B has to be true.

So, this small truth table here can show us when the argument is going to be also invalid. This is how we show by truth-tree how the arguments validity and invalidity can be shown.

(Refer Slide Time: 17:39)

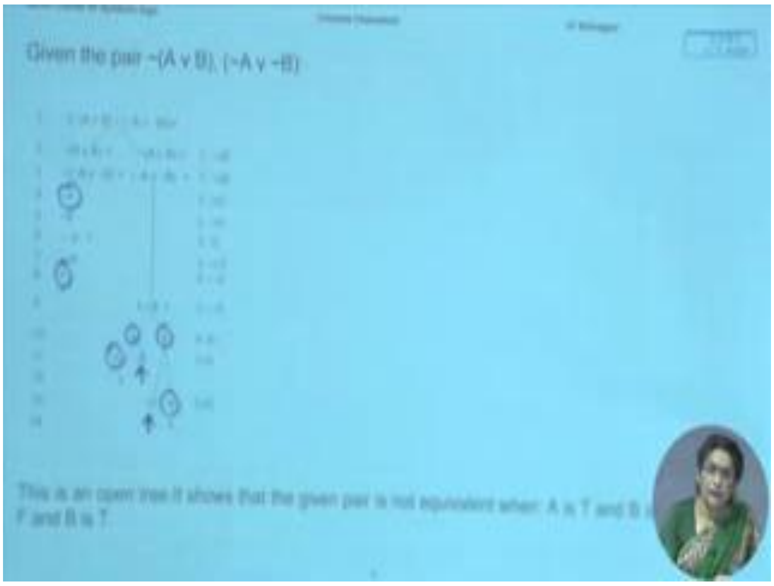


The last remaining task is logical equivalence. When our two propositions logical equivalence we know in terms of truth table when they have exactly same truth values. But what about truth-tree how do we show that? What we try to show is whether p triple bar q takes any two propositions p q, whether p triple bar q is tautology or not that is what we try to show. So how do you show, remember tautology proof. Tautology proof is that you take the proposition that you want to test tautology add a negation to it and do the tree and see whether it is result in a closed tree or not, that exactly what we will do.

So here, our given propositions are p q and we are testing whether they are equivalent or not. So, we formed this statement p triple bar q and add a tilde to it. What are we doing? Once more remaining ourselves we are testing whether p triple bar q is a tautology or not. So, just regular tree doing if it results in a closed tree you know p triple bar q is a tautology and p q therefore must be logically equivalent. And if the tree is not a closed tree then you know p and q are not logically equivalent. There is going to be at least one completed open branch in that tree from which you can tell under which truth values p and q are not going to be logically equivalent. So, this is what it is.

For example, if you have the pair this whole thing is your p and this is your q. So, this is your p and this whole thing is your q, tilde A wedge B that is p and tilde A wedge tilde B that is your q. What you do? You form a proposition like this. What is it we take p triple bar q and add a tilde to the entire propositions, so this brackets square bracket is for that to designate that inside there is p triple bar q and we add a tilde in front and then we do the tree to see whether we have logically equivalence or not.

(Refer Slide Time: 20:16)



So, let us do the tree here is the pair and here is the root of the tree, the tilde of the two propositions. If we do it correctly then this is how it is going to look like slowly, this is negation of R tilde triple bar D decomposition, and this is the left hand side and it would result like this. And it will give you a closed branch, because you have A and not A, you also have B not B, but will go take this one and will close it down. Here also you are going to have some closed branches for example, A and not A, but this one is opened. Here also you are going to have not B and B so you close it, but this one is going to be completed open.

So you know even at this stage you can stop and say that I have found my answer this one is closed, but here is a completed open branch and that tells me that the set this pair is not logical equivalent. So, over all your what you have found that if you follow this completed open branch then A is true and B is false, that is when this not going to be logically equivalent. If you follow this branch you will find when A is false and B is true these two will not be logically equivalent. This is how you show your clue is that it is an open tree and what we are trying to do is to test it whether p triple bar q is a tautology or not.

With this we will try to finish this module write here our discussion on truth-tree is over now, and there may be you need to practice a little bit on your own just to see where the operational difficulties might be. But conceptually you should not have any difficulty by now because if you are really familiar with the rules, if you have seen so many trees have been also done in front of you and I always tell you that you try to do it along with me if not then afterwards at least try to do the tree and then you see the result that has been done here, so that you get some practice on how to do the trees.

So conceptually you should be clear by now, but then you know every time when you do the practice there might be some little bit of an issue, but with practice that should go away also. But now we have finished all the tasks that we wanted the truth to do for us just like the truth table it is also able to give us clear answers on certain things as I have shown you. So, here we close module number 20.