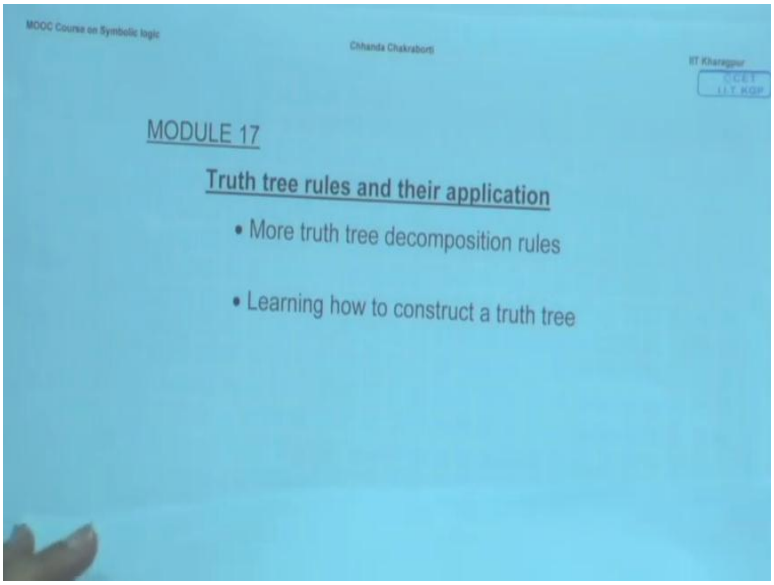
Symbolic Logic
Prof. Chhanda Chakraborti
Department of Humanities and Social Sciences
Indian Institute of Technology, Kharagpur

Lecture - 17
Truth-Tree Rules and their Application
More Truth-Tree Decomposition Rules
Learning how to Construct a Truth-Tree

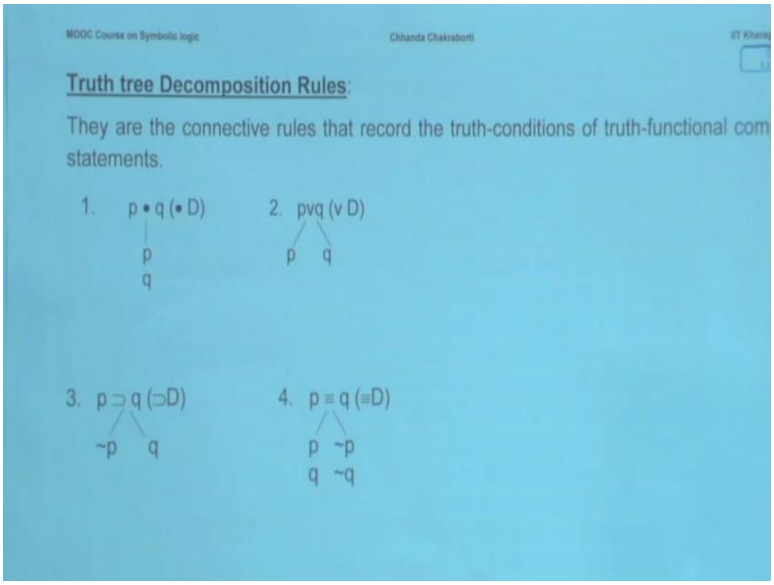
Hello and welcome to this module this is, our module 17 of symbolic logic course. So, we remember I hope that we were learning the truth tree procedure and I just got you started on that the truth tree procedure.

(Refer Slide Time: 00:43)



So, will continue with the procedure a little bit in this module also this is our module 17 and where are going to learn more about the truth tree rules and their applications. So, I have introduced you only four of these decomposition rules, but as there are more connectives left. So, we need to learn a little bit more on this rules and as we do that after that comes how to construct a truth tree. So, will go together on that and slowly try to construct trees together. So, together means that you can you can watch me doing it, but then again it is better, that if you also have a pen and pencil and a paper ready to, if necessary to immediately start doing this along with me because the results are in front of you. So, you can check the result as well as your performance vis-a-vis what is shown. So, we are going to learn about truth tree rules and the decomposition rules.

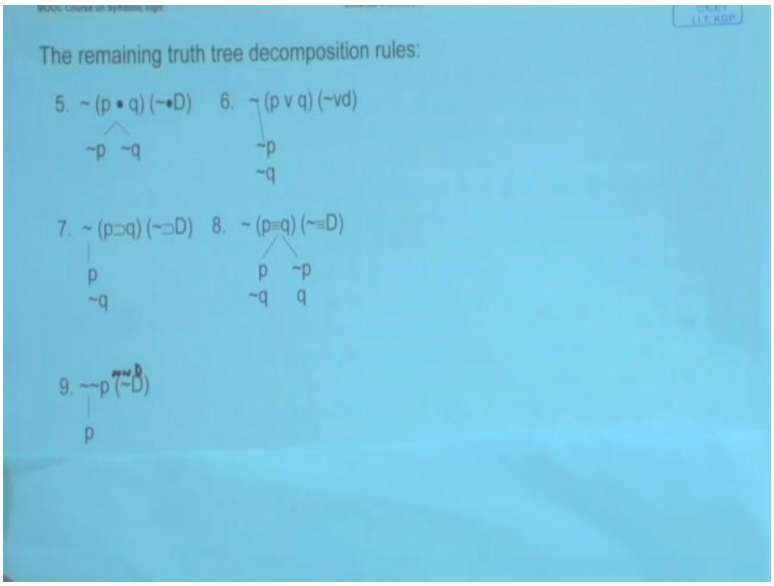
(Refer Slide Time: 01:46)



We started this. So, I will just freshen your memory a little bit and say that, you know what we were looking last were truth tree decomposition rules. How to decompose from propositions in a truth tree and we found out for example, that these are four rules which tells us how to decompose a dot proposition a wedge proposition a horse shoe proposition and a triple bar proposition. So, the main connectives as you can see are all over and each of these I have referred to you and have explained to you will be referred to by this kind of names the dot D or the wedge D or the horseshoe D composition or the triple bar D composition rules. So, try to understand them instead of memorising them try to understand the rules. So, that they stick to you better and then learn to apply them as we go along will take examples and will try to show you.

But this is not the entire set of the composition rules you need to know it more because there are many more connectives still left. So, let us take a look into those here are the remaining truth tree decomposition rules.

(Refer Slide Time: 03:09)



The next two talks about when you have a situation. When you have a situation when you have the negation of a dot this one talks about when you have a negation of wedge and apparently these rules are also called that this is negation dot D different from the earlier seen rule of dot D what are we asking we are asking when is the negation of a dot truth ask yourself. So, when is negation of P dot Q true means when is P dot Q false correct and we know that P dot Q is false. When either one of them is false one of the conjunct is false or when both of them are false. So, you capture in the truth condition as saying this that either will not P true or not Q true right.

So, this is your delta dot decomposition rule where you are saying when is till the dot true this is the rule called till the dot decomposition similarly this is your till the wedge decomposition rule where you are asking when is till the P wedge Q true right. So, in other case when is P wedge Q false and answer is very clear when both this chance are false that s when the wedge is going to be false in every other condition it is going to come out as true remember from the truth. So, that is what we have written. So, this tilde is true when not P true not Q is true in other words when P is false and Q is false and that is what is captured is a linear branch for the rule till the wedge decomposition probably.

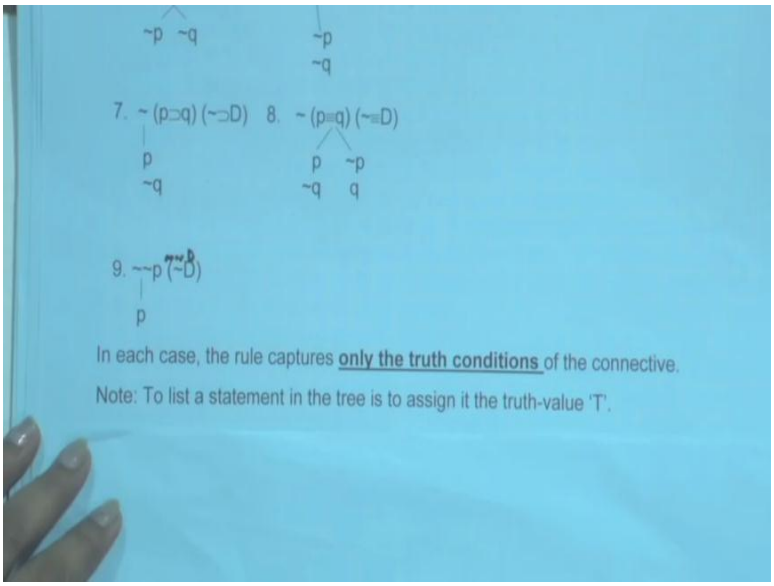
Why do we need to know this because you may encounter in your tree also the negation of the dot proposition or the negation of a wedge as a proposition? So, when you have encountered this you need a rule two decomposition that and here are the two rules to

start you by similarly these two are negation of horseshoe and negation of triple bar situation. So, what we are asking here is when that tilde P horseshoe Q true in other words is. When is P horseshoe Q false do you remember the horseshoe truth ever if you do you know that P horseshoe Q is false when P is true, but Q is false. So, that is what we have captured. When is still the P horseshoe Q true when P is Q and not Q is true not Q is true is means Q is false. So, this is your tilde horseshoe decomposition rule here comes that tilde triple bar decomposition rule.

So, triple bar when is it true you have seen it in triple bar decomposition rule, but this is when is negation of a triple bar true in other words when is P triple bar Q false. So, whenever there is a value mismatch of the equivalent terms that when you have the triple bar is false. So, similarly we have captured it in two sets of conditions as you see that two braches either when P is true or Q is false which means not Q is true. So, that is one set when, P is true and not Q is true or when not P is true which means P is false and Q is true got it.

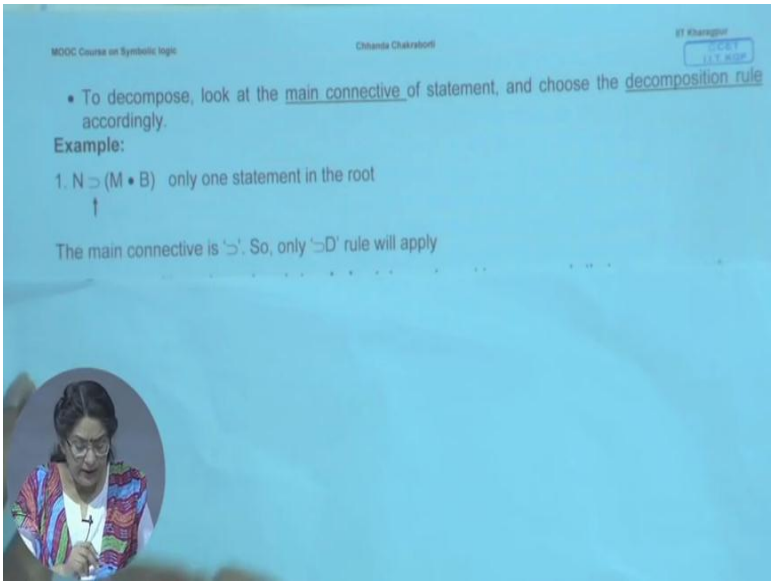
So, when P is true, but Q is false or when P is false, but Q is true that is what we have captured it here in the true negation of triple bar decomposition rule and finally, you needed this also when is negation of a negation truth when you have the proposition. So, it sort of cancels each other. So, when is tilde? Tilde P true the answer is when is P true and this is going to be known as sorry that is the mistake that should be tilde, tilde decomposition. So, this is what will complete our total truth tree decomposition rule you have 9 of these because there are nine possible kinds of a con connectives that you can encounter the truth tree and when you do, you have the entire set of rules to apply to it.

(Refer Slide Time: 08:05)



The matter that I will repeat because though, I have said it, but it is important that you take it in is that the rule will capture only the truth conditions not false D condition. So, you have to be creative in a binary logic domain to use the tilde and the lateral to express the false figure condition, but the rule will only show the truth conditions and one small to remind you listing anything in a tree is to claim that it has a true. So, with that will go forward with our further things that we can do in the truth, tree let us try to understand this.

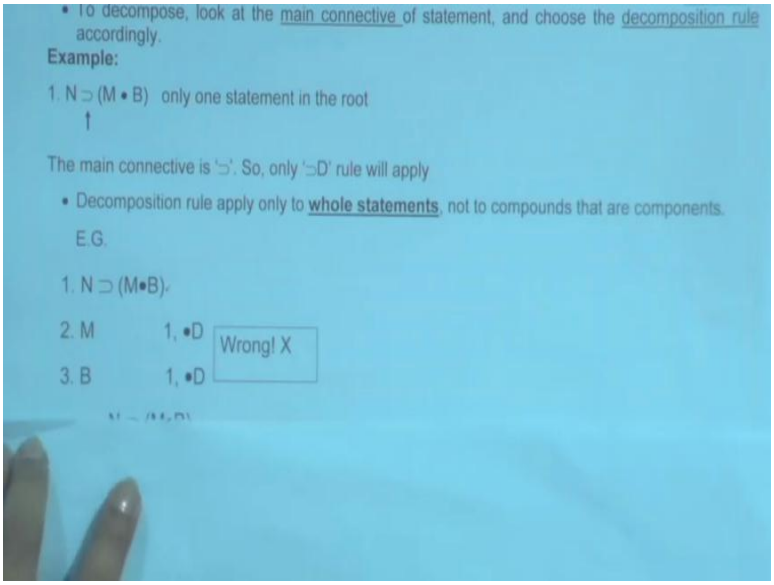
(Refer Slide Time: 08:48)



So, now you know when we said that how do we decompose the answer is that you pick a decomposition rule that fit is to your main connective right. So, you are dealing with main connective as the dot then, you all know that you need the dot decomposition rule here if you are dealing with the wedge proposition you need the wedge decomposition rule as simple as that lets take a example. So, here is for example, the only statement that is given is N horseshoe M dot B, there are two connectives here, but I hope you can immediately make out which one is the main connective which one is the horseshoe fine.

So, in here the only appropriate rule of decomposition is going to be the horseshoe on the other hand if you do this. So, only horseshoe decomposition rule will apply here you will say that what about this dot how can I get into that the answer is once you are decomposed the horseshoe only after that the end of the dot inside M dot B will become accessible to you not before that that is the order you get at the step a main connective in order to get into the sub connective.

(Refer Slide Time: 10:09)



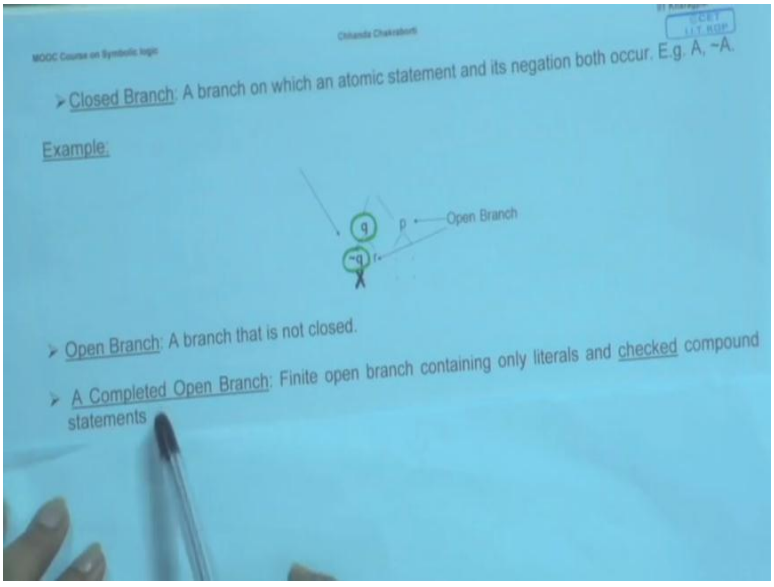
So, that we have to remember. So, otherwise there will be errors for example, say this is what we also meant when we said earlier and remember I said, I will explain it is that the decomposition rule apply only to whole statement and not to the parts not to the parts of this compound there may be compounds which are component, but you cannot access this until you are broken open the main connective same point it is the same point, but let us try to see this.

So, here is for example, a proposition route now by mistake if, you think that I want to get it here and you say M and B what are these what you have done is to generate branches from here by what you have done is operational line one this is one by dot decomposition dot decomposition will give you M and B. So, two separate lines as you see two separate numbers and you are justifying it that I got it from one by dot D decomposition is that correct now answer is wrong for everything that I have explained in the last few seconds may be is that you cannot even access this dot until you are broken the horseshoe open.

So, I will repeat that the only rule that will apply here the decomposition rule is the horseshoe decomposition rule why because horseshoe is the main connective gets me. So, in other words you need to be careful about this kind of procedures that you cannot step ahead to make it convenient for you have to do it in certain order and the main connective is the right place to start with for the decomposition and there are enough rules to take care of all kinds of mankind.

It is time now to go over the other technical parts of the tree and this is going to be something to sort of get used to we have seen. So, for that there will be the route then there will be decomposition process following the decomposition rules that we have just learnt as a result of decomposition from the root there will be branches right.

(Refer Slide Time: 12:45)



Now, some of these are going to be what is known as closed branches remember you have heard this term earlier also closed branches what is a closed branch a closed branch is a branch on which as a result of decomposition you may find a literal and it is negation both I will repeat that. So, a closed branch is a branch on which as a result of your decomposition you might find a literal and it is negation both. So, for example, you might find a and not a four this is when you have such a situation that in the same branch you have both a literal and it is negation you know that it is a closed branch try to understand this remember what we said about listing whenever you are listing a statement in the truth tree you are claiming it to be truth what is the situation in the closed branch that you have found a literal and it is negation both listed.

So, you have found both a and not a listed which means what you claiming that in this possibility or in this branch a is true not a is also true is that even possible in our binary logical word the answer is no we cannot have both a and not a true at the same time that is our basic law in this binary logical word. So, which what happens then if you find such an rather absurd situation logically absurd situation binary proposition and logic then you can declare that this branch is closed for what closed for any further logical operation in it means that you are ruling it out terminating it and you are saying that this is this is not something that we will continue it this is what closed branches are I will just show you.

So, for example, that this is the small example of course, that you know suppose you are doing the tree here are the roots somewhere about and then as a result of decomposition please take a look closely here, is that what you have is that on the left hand side somehow you have generated Q and then later on again for decomposition you have generated not Q can you see. So, when you have this situation take a look this branch is a closed branch this branch is closed branch why is it closed branch because in this both not Q and Q have appeared as a result of decomposition from about note and when you do that when you do in when it is closed branch how do you indicate that it is closed the usual way to do that is to put this cross under it. So, this cross means that we are not going to do any more operations on that this has to be put under the closed branch.

And this circling is also important to remind you also your reader this is the reason this branch is closed. So, you encircle the components of the literals which have lead to the closure of the branch did you get that. So, follow the branch through and speak out the literals because of which the branch is closed down like. So, so this your closed branch

example, but when you are doing this it may happen and new you need to understand this can we zoom it further to show this picture relatively clearer to show that this when one branch closes down it is not necessary that every branch will closed down right for example, here you have r can you see that.

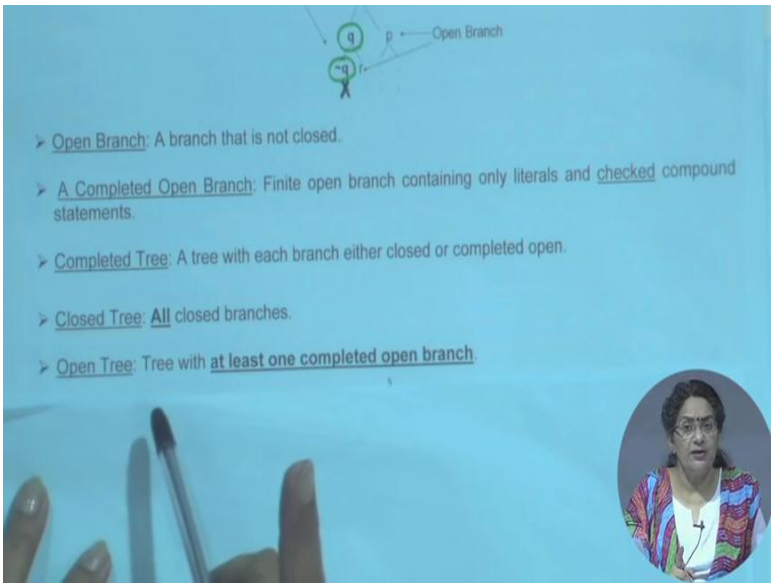
So, this bifurcation that is happened from Q on the left hand side has generated not q, but on the right hand side there is an r is that a closed branch no it is not it is not because no such situation has happened yet on the branch similarly take a look at this point you have Q coming here on the left hand side and P on the right hand side and notice that in here there is no effect of this closed branch. So, that is also possible that is also possible that you may have some branches closed in the tree were as there may be the other branches which are remaining open as if not closed. So, that is our next thing to learn

What is this open branch and open branch is that branch which is not closed in this kind of chance and it should the branch is this is what well mean by closed branch this is what well mean by open branch there is something about completed open branch. So, branches can remain open and branches can be open and completed there is the difference what is the difference a completed open branch is a finite open branch which contains only literals and check compound statements check means that a tick mark has been given against them I show you an example, but a completed open branch and let us take it in what is that in which all decomposition have been finished how do we know that because it contains only literals remember the decomposition stops when you have come down to the level of literals. So, if this is in open branch which contains only literals and whatever compound statements are there if those are all checked as in ticked when do you tick it when decomposition is been done on it.

So, in a way completed open branch is some open branch in which all decomposition that were to be done on the branch are over finished done yet it has not closed down. So, that is what is known as completed open branch then you will see that in completed open branch from a completed branch you actually can get a lot of information out of. So, there is a difference between open branch and a completed branch and an a branch may remain open without being completed that is the first thing to understand this is why we are making this difference that there is open branch branches can be remain open without being completed, but this is the situation, where it has remained open even when all decomposition have been accomplished get it inside your head that there is a major

difference here and whatever logical information you need to recover will show you how to can be obtained from the completed open branch not necessarily from the open branch why because open branch may not remain open all decomposition steps have not been done it may close down it may it is the work is still unfinished on that. So, you have no reliable information to be obtained from that you need information from this kind of branch.

(Refer Slide Time: 20:29)

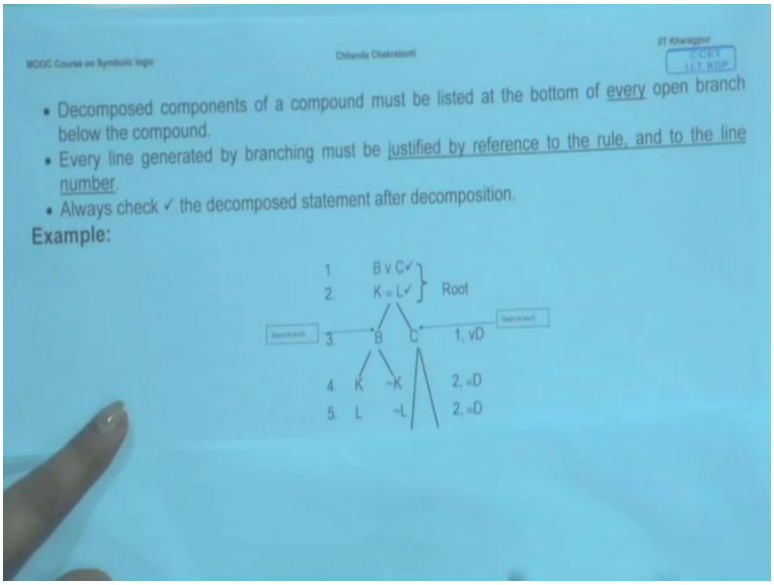


Then here comes can we can we now zoom out. So, then here comes the other thing completed tree what is a completed tree a completed tree is the tree where each branch is either closed or completed open. So, this is the tree where everything that had to be done is finished how we know either every branch is close down. So, there is no operation to be done or whatever operation was there has been finished which is why you have completed open branch and this is the kind of tree that we are looking for you need to finish the work on the tree in order to qualify as a completed tree this is a strange tree you know some times it may result and this is called the closed tree a closed tree is one which as all closed branches all closed branches.

So, can that happen well it depends on the root and you see that sometimes they speak out and this may happen in some cases not a single branch has remained open in comparison what is an open tree the open tree is the tree with at least one completed open branch. So, this is our first of all the details about this. So, let us take a look into

them all together we have just learnt what is a closed branch what is an open branch what is a completed open branch what a completed tree is what a close tree is and what is an open tree and we will try to take this knowledge further in our construction of a tree now we are going to learn how to construct a tree how to read the tree and so on so forth, but using this kinds of technical ideas.

(Refer Slide Time: 22:33)



Reminding you and we are soon going to start doing this together, but reminding you that when you are doing the decomposition the results of that decomposition must be listed at the bottom of every open branch the compound that you are decomposing the result of the decomposition must be listed at the bottom of or under every open branch that runs through their compound it is like parental property you know. So, if you think about the compound as the parental property then who gets it the all the inheritors like all the legitimate progeny or the sons and the daughters should get alright.

So, whatever branch is still open under this compound should get the result of the decomposition given to that other branches we are not running through this compound may not get the result of the decomposition will show you how. So, you because branches are going to open in different places and different places, but whenever you are doing decomposition remember this rule that result must be listed under every open branch that opens up under that compound that we are decomposing not necessarily has to be replicated for everything this we are going to see that it is important that we

mention it right it now that whenever you are generating a line remember the root is given root does not required any justification , but every line that you are generating by decomposition that is branching must be justified because is the new line that you are adding how do you justify by referring to the rule of decomposition that you have used and to the line number to which line are you decomposing on and will show you examples.

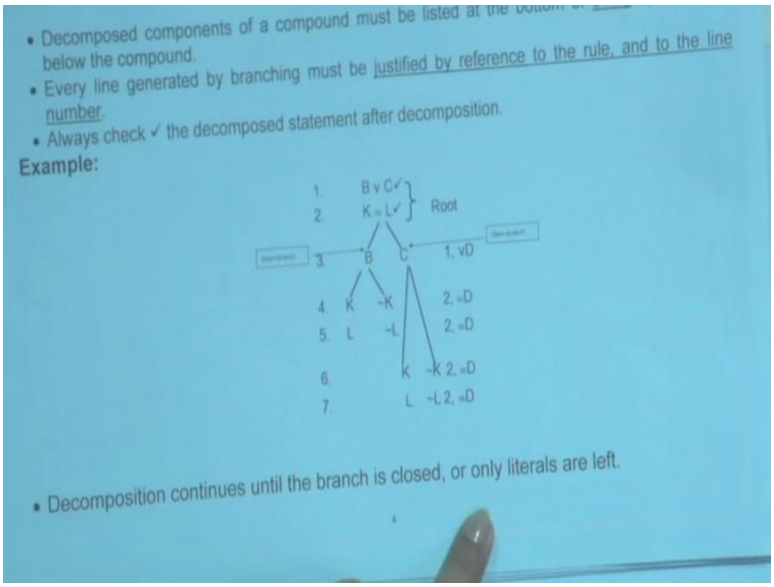
This is called checking as a ticking. So, whenever you decomposing a compound statement remember to put a tick why as a reminder that this has been already decompose do not forget to do that because you will soon see the that will be a helpful reminder for you because a root may have multiple candidates for decomposition right and as you are growing the tree as you are doing decomposition in the tree you may not remember whether there is skill something left in the root to be decomposed or not why if you leave something and still not decompose what will happen that is the results may not show the true situation and so on so forth. So, you need to that is the reminder that is the reminder that is the practical check from my side, but remember to check with a tick against every decomposed statement whenever you have done the decomposition.

let us take an example and I keep I suggest that you take out your own piece of paper and then you start doing this suppose I give you this root this root which has two statements proposition. So, b wedge see and k triple bar l right now we have to do the tree this is up to here as root then you see first thing if you if you go which one to decompose you do not know. So, you started with a line number one if you do that notice how the branches grow right from the root like. So, and you write the result of decomposition as b c and this is the justification given that line three was not given. So, you need to justify how do you justify line number one you got it by wedge decomposition this is not closed not closed these are still open branches I will show you one more and then please remember that at this point you are not done why because number two still to be decomposed right.

So, we need to have decomposition done here and we need to do it decomposition here also these are two branches no. So, only decomposing it once will not help why because both the branches are under this compound we just learnt that whatever has to be decomposed has to be done under every open branch. So, this is an open branch this an open branch. So, decomposition results should be repeated under and how do you do that and that is the point. So, this is my result four and five we generated by decomposing this

triple bar line two triple bar decomposition line two triple bar decomposition do we need to write this twice the answer is yes every line has to be justified. So, here is k l and there is not k not l and this is where you put a tick similarly when you are doing this decomposition remember to hold the other side. So, decomposition when one side of the branch is happening keeps the other branch on hold why because otherwise it is very confusing for you. So, procedurally I am saying you need to keep this out.

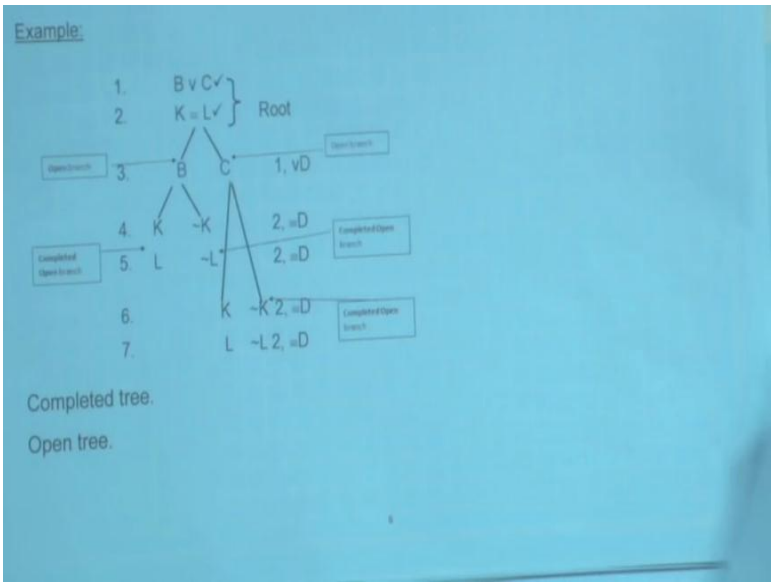
(Refer Slide Time: 27:54)



So, this is when you are doing it on the right hand side two line two still triple bar decomposition line two triple bar decomposition right and here is the result.

So are we done with the decomposition take a look you have reduced it all into the literals levels and all decomposition have been done only literals are left.

(Refer Slide Time: 28:18)



So, let us take a look into this and this would be our last slide for this module that this is the situation with this tree you did the tree and you found that this is an open branch this is an open branch. So, the results have to be entered under every open branch fine, but is it completed at this stage no when is it completed at this stage that is the completed open branch everything all decomposition, we have finished that is the completed open branch this is the completed open branch which makes this a completed tree there is at least one completed open branch and what this makes it also an open tree let me. So, this is where we are going to close it for this module and go though it a little bit because we have more exercises in the next module.

Thank you very much.