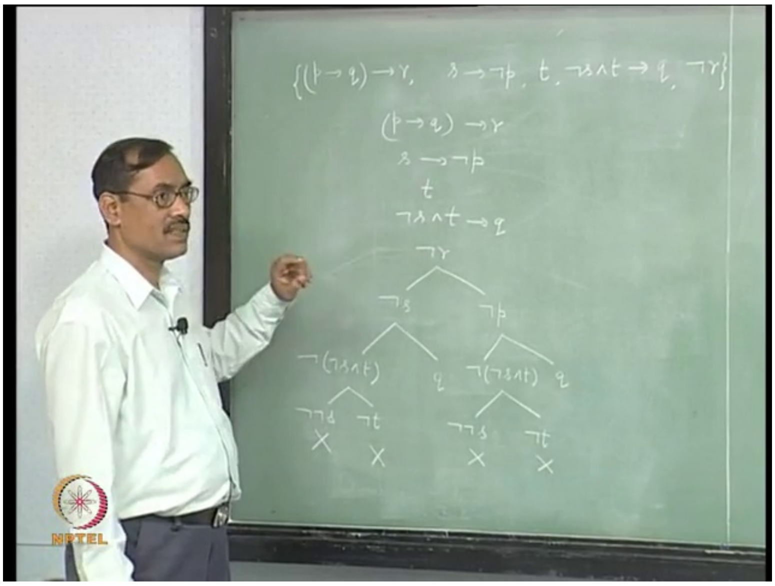
Mathematical Logic
Prof. Arindama Singh
Department of Mathematics
Indian Institute of Technology, Madras

Lecture - 20
Examples of Tableau Proofs

So, we were discussing analytic tableau, and then we have proved some set to be inconsistent, but the definition of inconsistency and consistency should be looked at exactly the way we have defined; that should be taken that way only. Like, you say that a set of proposition is inconsistent. Now, it is tableau inconsistency not PC inconsistency; so, it is told to be inconsistent, if there is a closed tableau for the set of propositions. All that you want is, you construct one such tableau which is closing. That is what you want to check, and for consistency just its negation, that you have to see whatever tableau you construct, it remains open; every tableau for the set of propositions remains open. If it is never closed, then you can say it is consistent. For inconsistency it will be then easier, right? For example, let us see one more.

(Refer Slide Time: 01:13)

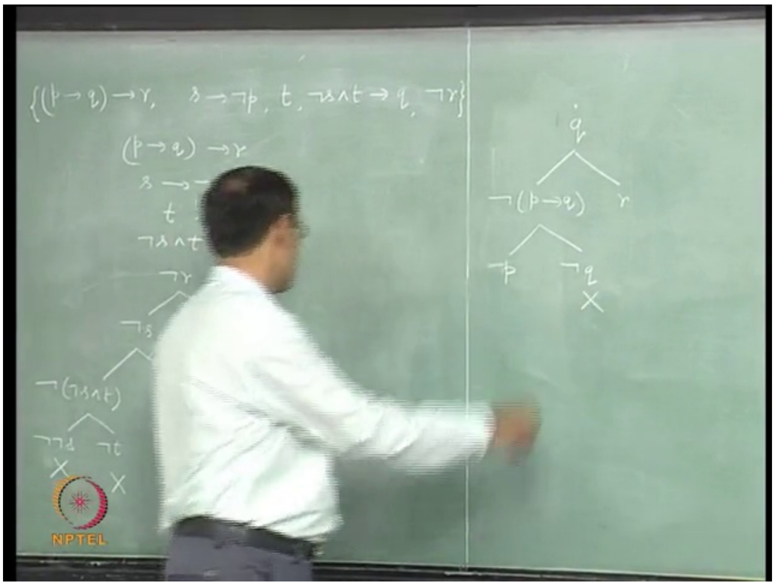


Say, this set is inconsistent. We will be constructing a tableau for this. Let us start with all of them at the root, then go on applying the tableau rules. Now, I see not r, not t, they are already literals and there are three others which are compound propositions on which tableau rule can be applied; and all of them are branching. Any one of them I can choose and proceed. Let me chose this one, and this is smaller. This gives two branches, one is not s, another is not p, is that okey? Because all that you remember is if it is in the form a implies b, it is equivalent to not a or b, because of the semantic trees. Then it will have two branches, one will be not a, another will be b. If you found a tableau rule, then you can reconstruct in that way.

Now, what happens, nothing is closing really. We have only t, not r, then not s; another path is t, not r, not p. There are of course, other propositions in the path; then one more we can start with. Shall we start with this one? They will be same of course, let us try. This one gives again two branches, not of not s and t, another is q. But now if you go to the other path you wanted to expand it breadth first, then once you are using it, that again you have to add here.

Let us do that. Again this one gives two branches, not not s and not t, same way this, is that right? Now, this path closes because not s and not not s, that closes; this path also closes, because it is t, not t, same way, this, we have now two other open branches, open paths. I have q, not, s, not r, t and this. There, I have used s implies not v, this one also has been used, first proposition has not been used till now, right? You can still expand this path, right?

(Refer Slide Time: 04:23)



So, let us expand that path. I will take that path here, on that I have p implies q implies r. That will give rise to two branches, not of p implies q and r, this one, it does not close. No, there is no not s, not s is on this side. This is not closing, only this is closing not t. Yes, good. That also you have to expand again, let us see this. First one is q, which is open, there we have this, that gives rise to again not of p implies q, not implies rule. That will give two, one is p, another is not q. One is p another is not q, now this not q, closing, correct? There is q, there anything else is closing, it is here, this path we are taking there. We have t, not r, not s, q; there not q has closed, but about p, that path still remains open, p and not q, right? It is not different branch, they will be on the same branch, right? Both the things will be there, you can write p this way, and not q this way, or sometimes we do not write this vertical bar, you just leave it. Now you see that, that closes with q itself; there is no open path there. What about the other side? There is r, you have not r on the same path, right? So, that also closes. Now, what about this q?

Student: Same not r there.

The whole thing will be copied there because there you want to apply this, also the whole thing will be coming here, that is up to what you see here, that is also for the other q, right? We have to take another copy really on same page of paper, you have to do it exactly, right? That means, this set is inconsistent.

Student: Not not s is there.

Not not s is there. So you have another branch, there again you have to expand. Now, suppose you expand it, what happens there? Again you have to come to this branch. Again the same copy will be there. Once it is there, what do we get? It will close with.

Student: One end close with r.

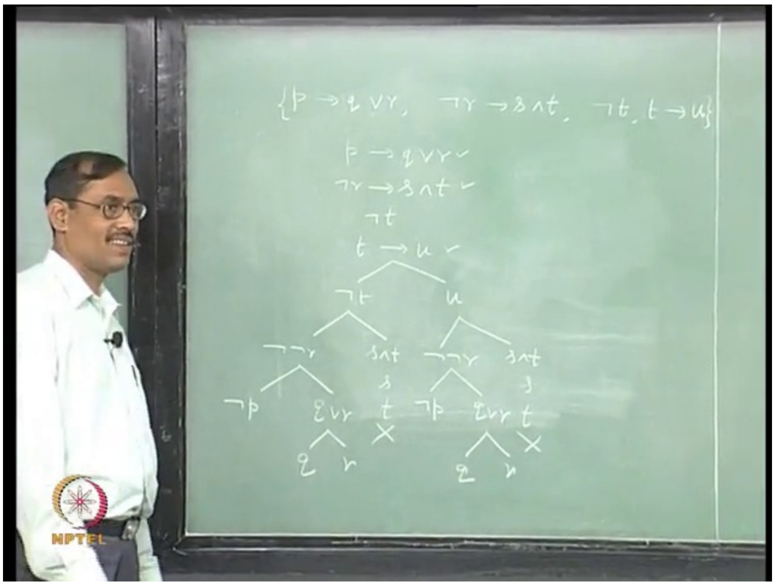
p will close with not p, right? What about the other one? That is r and that is not r, is it clear? The same copy will be here again. Now there, you have p, not q and r, p, not q, in the same path. So, p, not q, with that you have not not s, you have not p, that closes and the other one, you have r and you have not r here; that also closes, is that clear?

Now for consistency what should we do? Let us start, say, I take only this much, what should I do? We start the tableau now, but the tableau copy is there already, can you see there? Suppose, they do not have this, I have only 1, 2, 3 propositions. For those four propositions, I have already made the tableau here, the first proposition was never used till now. Therefore, this is the tableau, in this tableau what happens, I have a closed path, now there are three other open paths, fine? From these can you say that it is consistent or not? Yes, see I can tell now because all the compound propositions have been expanded; they have been used, right? Suppose, it is not that I have not done up to that, I just stay here. I just draw the tree, I do not expand, I keep up to this place. Now, I say that this tableau is also open, because this is also a tableau for the same propositions, same set of propositions.

Student: But they have not opened up there.

I have not used them, but as a tableau, it says it is an open tableau up to this. Also, it is an open tableau, but here you are telling, we will decide that it is consistent, but up to this stage if I take, you will say it is not decidable. Here I have to go for some more steps, right? The reason is we have not used some of the premises, right? That amounts to telling that the tableau should be completed, is it right? If the tableau is completed, then only you can decide whether it is consistent or inconsistent.

(Refer Slide Time: 11:25)



The completed tableau if remains open then it has to be consistent, because essentially what happens for consistency you need that every tableau should remain open, not the completed tableau or any completed tableau, you want every tableau to remain open. Now, if you take a completed tableau then essentially it has a copy of all the tableaux because all the premises have been used. So, whatever there can be, expanded, either they will close or if they do not close then that open path will be still present in any other tableau, fine? That is the reason you say that it is consistent when some completed tableau remains open, you do not have to go for every. Now, see completed tableau becomes helpful, fine?

Let us see one more example. How to decide this, say, q or r? We will start with this, we do not know whether this is consistent or inconsistent. If you guess that it is inconsistent then you just go for one construction of the tableau, if you are not able to guess you will be going for the completed tableau, right? Where on every path a rule should have been applied on the compound propositions, that has to be checked, then it is a completed tableau, but you do not need that; even before that it closes. Then you stop there because a completed path is one which is either closed or if it is open then. Then all the compound propositions, on all the compound propositions rules have been applied, right? That is what we are going to do.

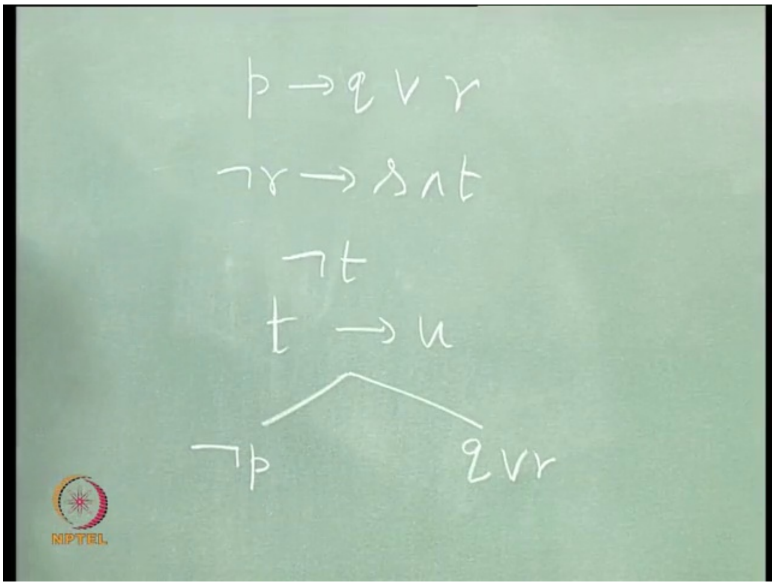
Let us start the tableau. You have to take all the propositions at the root, right? Now, I go for, say, branch out from this one. I get not t, u; we will take the other one after this, that gives me not not r and s and t here also, right? Then I can proceed say, s and t, I will stalk them, I have not yet checked where it is closing or not, is it closing? Yes, not t with t, it is closing. You do not have to expand this further, this also, I do not have to expand. There is one more which is to be used. Let me use it now. I should get not p and q or r again here, also not p, q or r. Now, not p is there you do not have a p there, from the path, and here also you have q, r here, again you have q, r, is it a completed tableau? In a completed tableau each path should be completed.

So, this path, it is closed, so it is completed; this path is open, but on every proposition occurring in that path, on every compound proposition, a tableau rule has been applied. Therefore, that is also completed, if it is any other, say this one. So, q or r is another compound proposition, it may not be the premise; it can be obtained from another premise. On that also, it has been applied, on all these, that is also applied, so that is also a completed path, but open path, right?

So, it is checked, that yes, it is completed; but it is open. Therefore the set of propositions is consistent. You can also make it very mechanical, the way we have done is we are just using which term is convenient for us. We thought this will be convenient; it is giving only two literals directly, so use it and so on, but when you do it by a machine, it is not able to find some heuristic, right? Of course, you can give those heuristic's, feed the algorithm with some heuristic like that. But let us have a crude algorithm which says you just start the tableau systematically; take some ordering of the propositions, whichever order you decide in the beginning itself, do not change it later.

Follow that ordering, go on using the rules. But there is one thing which we have to modify; that for example, we have taken not p, q or r, this comes from the proposition p implies q or r. Now, immediately we take q and r here, those two paths we have taken. Imagine we have not used it at the last, but used at the beginning, right? I can start the tableau from there itself.

(Refer Slide Time: 15:56)



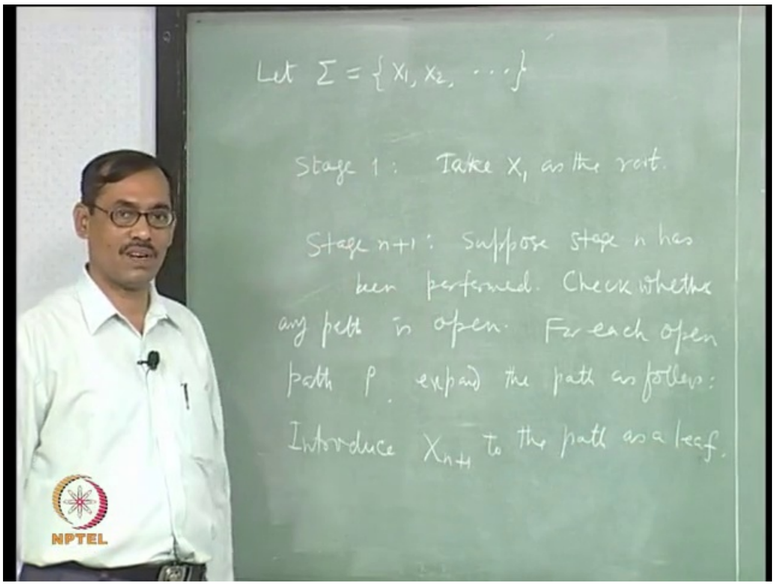
Say, I take a copy of that here, okay? Suppose, I use the first one; so there, I will have two branches, one is not p, another with q or r. Now what happens, you can apply a rule here directly instead of going further, using other premise. But in a systematic tableau we will not do that. We want to see that all the premises have been used slowly and then we will develop it path-wise. We look at a path, see, whatever is the compound proposition from the top, use a ruler on that, right?

That is what a systematic tableau is. Because what happens here in propositional logic, it does not matter, but later, suppose, you will have some opportunity to reuse the rules. The same compound proposition can be overused many times; it will give rise to different conclusions. Like, suppose you say for each x Px, in natural numbers. I would have taken taken P1, I would have taken P2, I would have taken P3 and this is infinite. If I proceed from Px, what to conclude, I say P1, P2, P3, ... where to stop? So, we will make it systematic so that it can be again extended to fast order logic later, that is the reason. What we do here, we will not apply a rule on the recently got premises, or whatever has been obtained from the rules. We will start again looking at the path, from the top of the path you go on finding the compound proposition, apply the rules, follow it systematically. Let us see how does it look like.

There is one more difficulty. Suppose, your set of propositions is infinite, then what will you do? There will be problem in applying the rules, I am thinking that the tableau will finish there, but it is possible that the tableau still will become finite. If it is closing, everything is closed, after that there are only redundant propositions. I have p, I have not p, after that everything is redundant, right? It will simply close, so similar thing can also happen even if the set of premises is infinite, okay?

Let us think a bit. How to give the systematic tableau for the infinite sets directly? Anyway it will be a countable set because the set of propositions is countable. So, whatever set you take, any set of propositions is a subset of the set of all propositions, it will be countable. In that you can have an ordering. Let us start with a set of propositions sigma.

(Refer Slide Time: 18:50)

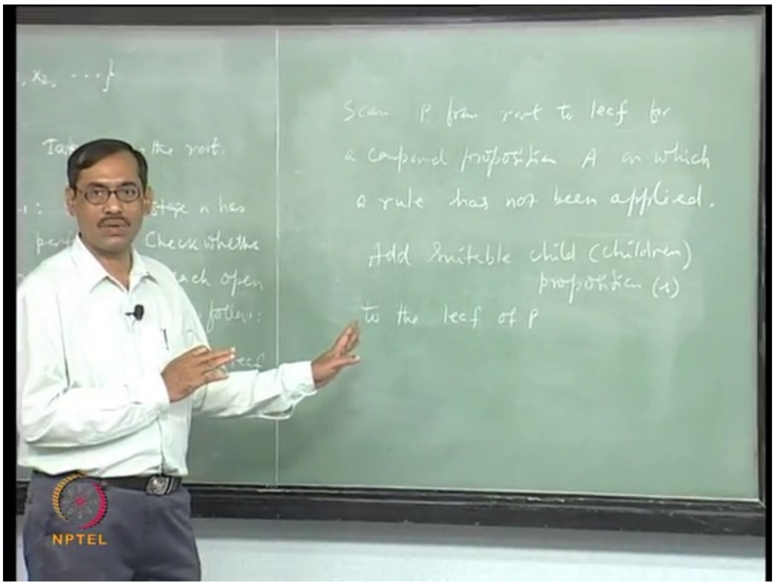


Let us write sigma equal to X1, X2, and so on. You are thinking as, this as an ordered set. Now, the way it has been written, that is its ordering, right? X1 is the first one, X2 is the second one, X3 is the third one, and so on. What we do, we will define the generation of systematic tableau in stages, because there are infinite things. Each stage will be defined inductively, after one stage, fine? First stage is this: in stage 1, what you do, see the problem is we cannot take everything at the root, the way we are doing earlier. We have to start with one; that is why the stages, right? So, introduce X1 or take it, take X1 on the root, X1 as the root, we start with that. Then after stage n what we will do? In each stage we are introducing one, then we are expanding, starting from the top. So, in stage n plus 1, we assume that Xn has been introduced already or stage n has already been performed. Then we have, we are going to do in the next stage, that is what we are concerned, right? So, suppose stage n has been performed, fine? Now, in stage n plus 1, we introduce, what we want to introduce really? Suppose, you want to introduce X n plus 1, where will you introduce it?

Student: On the open branches.

On the open branches. If some path is already closed, there is no need to introduce there, right? Is that so? That is what we are going to do, fine? Then, once stage n has been performed, first you have to check after this, whether any path closes. If closes, you do not have to do anything, so first check that. Check whether any path is open. That is the first thing to do. Then for each open path you have to do something. For each open path rho, let us call it rho, expand the path as follows. How do you expand? First we have to introduce Xn plus 1, right? Introduce or just add, whatever you write, X n plus 1 to the path as a leaf of course, right? Your introduction will be on the leaf level always, not from the root, that is what we are following. So, introduce at the leaf itself, X n plus 1 becomes the new leaf. Now, it is the child to the earlier leaf, whatever leaf was there on the root, right?

(Refer Slide Time: 22:43)

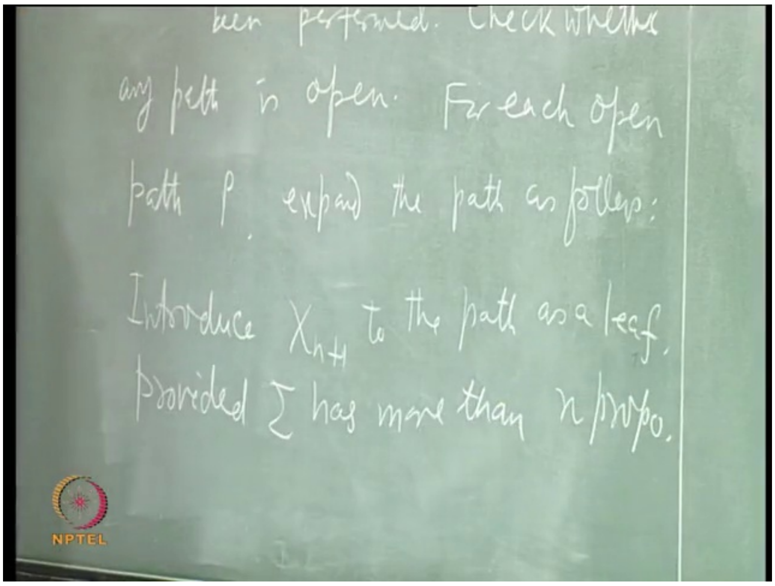


Then what we do, scan rho from root to leaf for a compound proposition, say A, on which a rule has not been applied. Suppose, that is the first, so you are scanning from the top, root to the leaf, top down on that path only rho, not other paths, other paths we are forgetting. Now, you are trying to expand only that path, and each path we will have to expand at the end, right? So, on that path from the root when you proceed, you find one compound proposition on which a rule has not been applied. Then apply the rule to get the child or children whatever it is, if it just stacking rule, then there will be one child or two, may be on the same path also, right? But if it is a branching rule, then there will be two branches out from the leaf. So, add those child or children, whatever it is, add suitable child or children, or you can write propositions. Once you write suitable, that takes care of propositions to the leaf of rho. Now, it is really for Xn. Xn is at the root, now leaf X n plus 1, write. So, leaf of rho, this is how you expand. One step of expansion is over. This is what we have to perform at stage n plus 1.

But there is one condition. Condition is suppose sigma has only n number of propositions, then you cannot find Xn plus 1, right? So, that is the condition, if there is at least Xn plus 1, then use it, for infinite only we are thinking. For finite case, there is a possibility that at some stage, all the propositions are over. In stage n plus 1, all that you have to do is, look for the compound proposition and go on expanding it, see if it cannot be further expanded, stop. Yes?

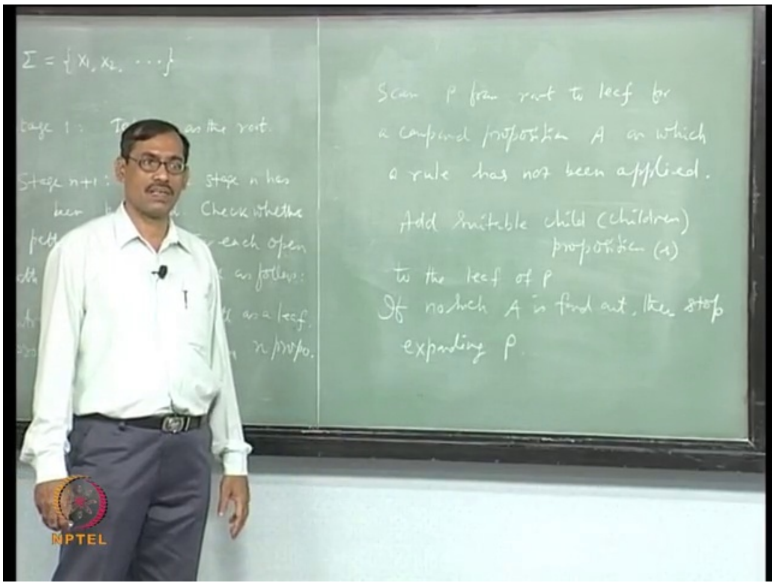
Student: Are we not expanding totally?

(Refer Slide Time: 26:19)



We are not expanding totally. suppose you come to know p implies q or r. So, at one stage it will give not p, q or r, q or r remains, you are not expanding at that stage, only one from the top. So, it may be expanded in the next stage or may be later, I do not know when, right? Again, I have to come from the top, I get this one first; then that is why it is systematic. We are not worried about whether it is finished or not. We should add something here, introduce Xn plus 1 to the path as a leaf provided sigma has more than n propositions; if it does not have, then stop there. Now, let us look at this place, so there is, when this algorithm will stop, then you cannot further expand it, right?

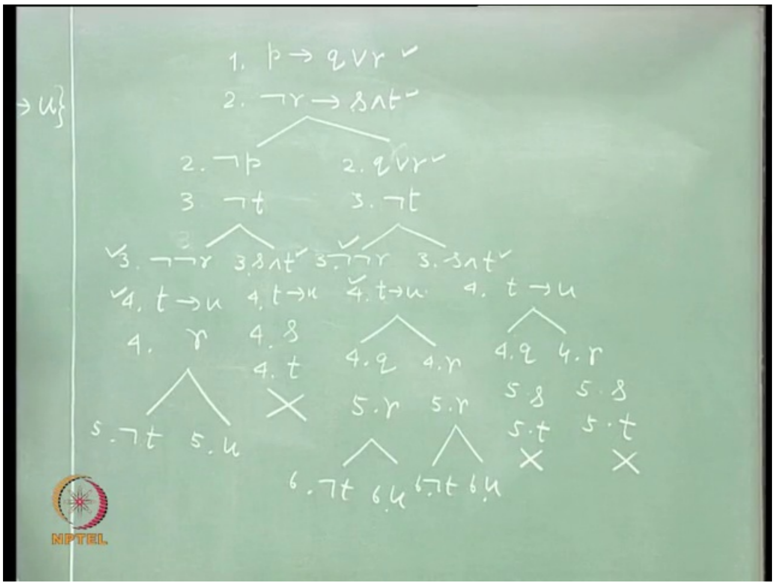
(Refer Slide Time: 26:46)



If no such k is found out, then stop, stop expanding rho. Now for each path we are doing that; this is only inductively defined. It does not say that at any stage it will stop, it may not stop here, because sigma is infinite and it is consistent, and it will not stop, it will go on forever. It was in p0, p1, p2, p3, all propositional variables, right? Now, you will introduce at each one stage one proposition continue, no expansion is required, still it goes on infinite, right? One path only, it will all stalk and it is continuing infinitely; there is an infinite path, is that clear? That is the only thing possible there. Now, let us see here a question. What happens here, how to develop a systematic tabular for this set? Suppose you take these propositions in that order, that is may sigma, as they are written, that is my ordering. You start with the first proposition which is p implies q or r, that finishes my stage 1. Let me write here my stage number just to keep my documentation. In stage 1, I have introduced this. Now, I do, in stage

2, I have to introduce second one if there is a possibility. Suppose, nothing is there, so again it will go to stage 2, I have to write stage 2, apply the rule, there is nothing to introduce, fine?

(Refer Slide Time: 28:07)



If there is something, I have to introduce there. So, introduce in stage 2, not r implies s and t, but stage 2 is not over. I have just introduced because there is one. Then what you have to do? For each open path I have to do something. This is the only path that is open. Now, you should go back to root, from root I have to scan which one is a compound proposition, on which one a rule has not been applied. So, I find first proposition which is my A. Now, it is a dynamic variable on that, so on my A, that I apply the rule. I get not p, q or r, both of them have been obtained in stage 2, there ends stage 2. Nothing more. It does not say you do it again.

Student: Oh, okey, but the loop?

It is not a loop, it is telling that, that is all. So, suppose that stage has been performed, I have to go back again, follow the procedure, the loop will be on its end, directly not there itself, there is no sub loop there, okay?

So, second stage is over, now I come to third stage, in third stage. In third stage what will happen, there is another we have to introduce that, so everywhere, it is not over. I have to again scan on each open path on which, a compound proposition on which a rule has not been applied. So, first one I have already applied in every path, I am doing it breadth first. So, that will be easier to look, easier to see it at least, then second one I have not applied on this path, it is done really path-wise.

So, on this path this has not been done, now I apply it. That is, again in the third stage, that is again in the third stage. It will have not not r, s and t. All these are done in third stage; here also there is another open path, same proposition is taken here. s and t, they are going on, third stage, third stage is over. That is the only thing I have to do. Anything is closing, that has to be checked. All these, nothing is closing till now; so you go to fourth stage.

Fourth stage, there is the proposition. I have to introduce in fourth stage. That gives t implies u, t implies u, t implies u, t implies u. Fourth stage starts with that. Then what we have to do? I have to look at each open path. One open path is here, the left most, I am taking, where? I have to find, this compound proposition, a rule has been used. So, literal, this is not a literal, not. Everything will be obvious for us, but not for the algorithm.

So, on the fourth stage I have to write r, this is done. We have not not rule, the rule can be applied. I have to apply it and I forget about that, for that path, in that stage. I come to next path, where I have to find similar. So, here again I get s and t, so that is again fourth stage, s, t. That is also done; so different propositions you might get in different paths, that does not matter.

Next open path I take there, q or r, that is also in fourth stage, q, r and this path, same thing, q or r. That is again fourth stage, as q then. Next, I go to fifth stage. There is nothing more to be introduced. Now, you have to first check whether anything is closing, out of these, nothing is closing. Looking, yeah, this is closing, t and not t, right? That is closed, next about this. I have q, not t, nothing is closing, r nothing is closing, right? Next path, you get, q, r, nothing is closed, they remain. Next, I have to check each path, whichever is open; usually the left most is taken by the algorithm, it says you start from the left most path and then continue.

Then? I have to take this open path, find out whether I have not applied a rule, apply it, that is the fifth stage. This is the proposition on which I have to apply a rule: not t, q, that is all about the fifth stage. Closed path, I do not worry. Open path. So, these two have been applied not t, not not r. I have to write r; that is all about the fifth stage.

Then here again same thing, fifth stage r, it just does blindly, no intelligence. Next one is again s and t. Fifth stage will be s, fifth stage will be t. And here also same thing s, t, that is about the fifth stage, over. Now, we have to again check, whether something is closing or not. This one is not closing. u, r, not, not closing, that is also not closing, r, this is also not closing s, t, q, not, t, closing; t, not t closing; right? So, fifth stage is over.

Sixth stage I have started when you are finding out which paths are open or closed. Now, in the sixth stage I have to go for the left most path, I see everything has been used, there is no other compound, right? Next path, everything is here, next path there is one, not used yet. So, use it, that is sixth stage. Here also same thing, sixth stage sixth stage is over. Now, seventh stage starts. This does not close this, does not close, nothing is closing nothing is expanded, over; it closes. That is also another way of telling the stepping criteria, if you get some open paths as the last stage you can stop because in the next stage only you can verify the whether some path is closing or not.

Now, you can compare this systematic tableau with the one you have already constructed earlier, this one, right? Systematic tableau will be bigger that is clear; does not matter. But there is a copy of this tableau inside it. Somewhere, may not be in the same form, same order, something else might be inside, because they are closed early. They can expand, that is fine, But then the copy is there.

So, systematic tableau is necessarily a completed tableau, right? It has to be completed, but that, you are guessing only finite case. If it is infinite case, how do you justify? That a systematic tableau is necessarily a completed tableau? You take any systematic tableau, it has to be completed. Why is it so? Well, if it is not completed then what happens, there exists at least one path which is not completed. It is not completed path, means what? So that compound proposition has been obtained by the tableau, either that there it is originally in sigma; if it is not originally in sigma, then it would have been introduced in some stage, right? It has been obtained. If it has been obtained, then from the root to that place whatever is the number say m, then it will be used in stage m, right? So, always you are concerned with the finite place even though tableau looks infinite, you can really get away with finite place, fine? Now, systematic tableau is completed. Once it is completed, what you have to do for consistency? You just go for the systematic tableau and then decide. If there exists an open path, then it is consistent, only thing for infinite, what to do?

Student: It may not have the systematic tableau.

Why not?

Student: Because, you can never complete the systematic tableau.

Student: Check only if it is inconsistent.

Student: You cannot check for consistency.

Student: If it is a consistent ...

If it is inconsistent, then it will close, everything will close in a finite stage; it is verified. Otherwise, it may go for infinite. If it goes for infinite, can you say it is inconsistent or it is consistent?

Student: No, there might be a future proposition.

Student: Basically we do not know whether it will stop or not.

Student: Yeah, you cannot say.

No, it need not stop, it is infinite; you are telling it is infinite. So, there is an infinite path. Let us say, in the tableau there is an infinite path, in the systematic tableau. Now, does it mean it has to be open or it is closed or it can be anything? If it is closed, it will close at some finite stage. Once it is closed, it has to be finite. Why is it so?

Student: It can close, but it cannot be complete.

Student: Are we considering infinite number of literals or finite number of literals?

There can be infinite number of literals also; as we told there can be a set with p0, p1, p2, p3, and so on, right? Suppose, you say one path is closing, there is a path which is closed, is it necessarily finite? Is it a finite path, is finite; isn't it? Yes? The path is finite or not? If a path is closed, is it necessarily finite?

Student: Yes.

Student: Yes.

Yes, then there is one infinite path, it has to be open.

Student: You cannot determine whether that path is infinite or not.

It is a separate matter.

Student: If it is given infinite, then it should be open.

Is that so? Yeah? So you are not able to decide whether it remains infinite or it remains finite. But it is true that if it is closed it has to be finite. If it is infinite it has to be open, right? There can be finite open paths. That is okey. If the set of propositions is finite in the beginning, yes, fine? So, what we see here is, in the systematic tableau we have to be concerned about the finiteness, right? Finiteness closed, finiteness open. There is something which we may need, because finiteness is not easy to look in the tableau itself, finiteness will give you something more there. We will see what does it mean.

Let us take any general tree. What happens there; see all these tableaux are binary trees, right? You have the binary tree, here in the binary tree you are telling whether it goes to infinite or not, I do not know, right? It is an infinite binary tree and I do not know what happens, whether one path remains open or not.

If it is not a binary tree, you can take any general tree, and you know there, it is finitely generated; that every node has only finite number of children, how many, we do not know. Can you tell, in that case something? I am, slightly, asking you a difficult question. But it may be easier to see. You have the binary tree here, it means you take any node, it has less than or equal to two children, right? May be one only, less than equal to...

Now, consider one finitely generated tree, which means you take any node there, it will have a finite number of children, not infinite number of children, right, as usual; but that finite number, I am not giving what it is. There might not be any bound, it can grow, like, from level 1 each one will have one child, level 2, each one will have two children, level 3, each one can have three children, and so on, but this is also finitely generated, is that clear? Possible.

Student: Random, random.

It is random, I am giving an example, that you cannot say that there is a maximum number of children of any one, right? That is possible. Then in that case what can you say about paths? Suppose a tree is finite or infinite. I give something, say, the tree is infinite, does there exist an infinite path or not? Why is it so?