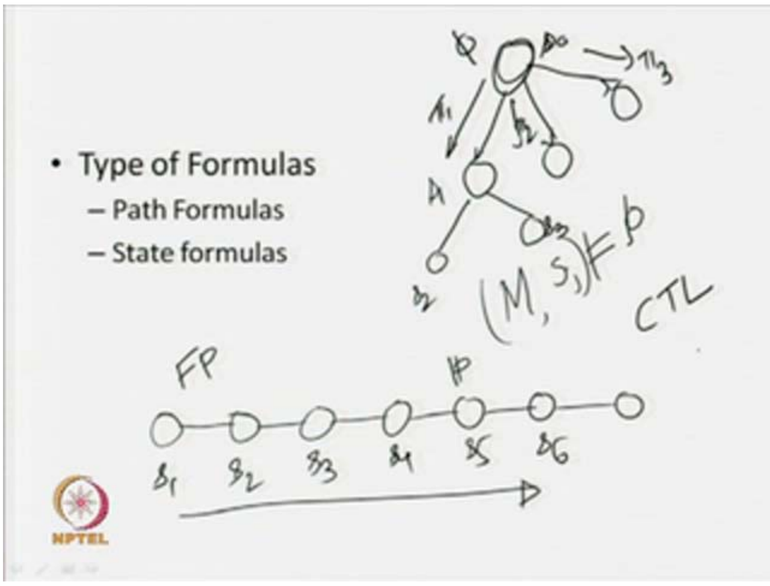
Design Verification and Test of Digital VLSI Designs
Prof. Dr. Santosh Biswas
Prof. Dr. Jatindra Kumar Deka
Indian Institute of Technology, Guwahati

Module - 4
Temporal Logic
Lecture - 3
Syntax and Semantics of CTL

So, in our last class I have introduce the notion of the temporal logic and I have talked about the temporal operators. Basically, we can categorize the temporal logic in two different ways; one is your past temporal logic and second one is a future temporal logic. And, in future temporal logic we can have or we have discuss about the notion of the four temporal operators; one is your next, second one is globally, third one is future and fourth one is your until operator, which is a binary operator. In this class, today I am going to introduce special class of temporal logic which is known as your CTL computation tree logics. So, I will say why, what is a what is a general notion of temporal logic and what is CTL. Now, after that we will go for syntax and semantics of CTL and after that I will give some example how to represent those things with CTL.

(Refer Slide Time: 01:39)



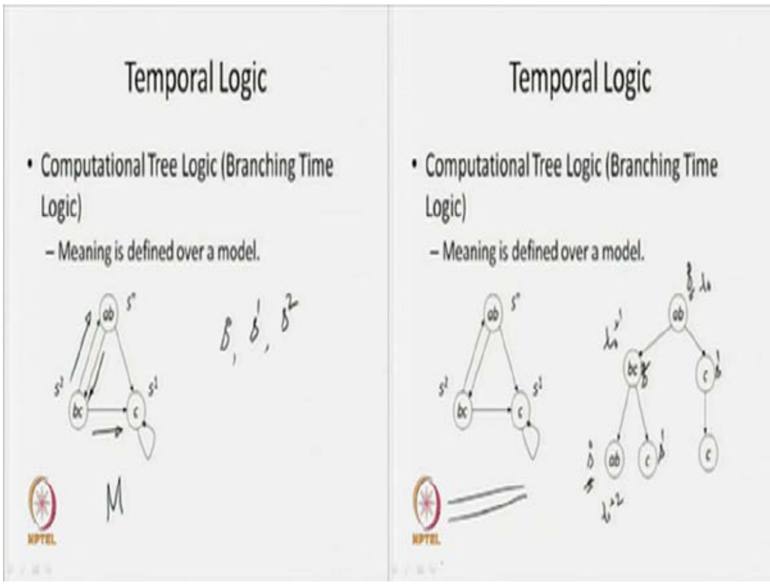
So, in temporal logic what happens? We have seen that we are going to give the meaning over temporal logic operator with respect to a model. So, you can have a model which is nothing but a sequence of states and we are going to keep the meaning of a temporal operator in a particular step.

So, we say that you, I am having a model M and I am having state say S j this particular combination model; a temporal operator phi, temporal formula phi. Now, we are having two notion one is call path formula and second one is call state formula. Now, just that in last class what we have discussed? That we are having some state this is a sequence of state like that; states are marks as S 1, S 2, S 3, S 4, S 5, S 6 like that. And, say that in S 5 we are having that atomic proposition P is 2; then, what happens? I can say therein state S 1 the formula F P is 2; in future P is 2. So, though we are talking about a state so on; it is related to this particular execution sequence. So, in case of linear temporal logic, what happens? We are having one path and we are going to reason about that particular path. But already I have mention that we are having the notion of your branching time; where time branch out in several direction. So, in that case we are going to look for all possible path or we are existed path.

So, in this case what will happen? We are going to talk about the state formula. Here, in notes general which are the this is in this particular path first that S 1 that F P holds. Now, in case of branching term what will happen, we just see that we are having a state as 0 depending on several condition and time may branch’s out in several different position. So, if I am coming to S 1 depending on the input combination or state of the system; it can proceed either S 2 or S 3.

Now, here we are going to talk about in state as 0. We have to see all possible combination; say this is a path pi 1, say this is path pi 2 and this is path pi 3. So, in that particular case we have to see the nature of this particular temporal formula in all those possible path and depending on that we are going to say the behavior of that particular formula phi in this particular state S 0. So, in this case we are going to say that it becomes a state formula. So, now we are going to talk about a particular class of temporal logic which is known as your CTL; Computational Tree Logic. So, in Computational Tree Logic, it is branching time logic. We have to look for all possible paths and when in a form a particular path we are going to look into all possible paths; then we are going to setup that particular formula is two in this state. So, we are going to talk about state formula in case of CTL.

(Refer Slide Time: 04:43)

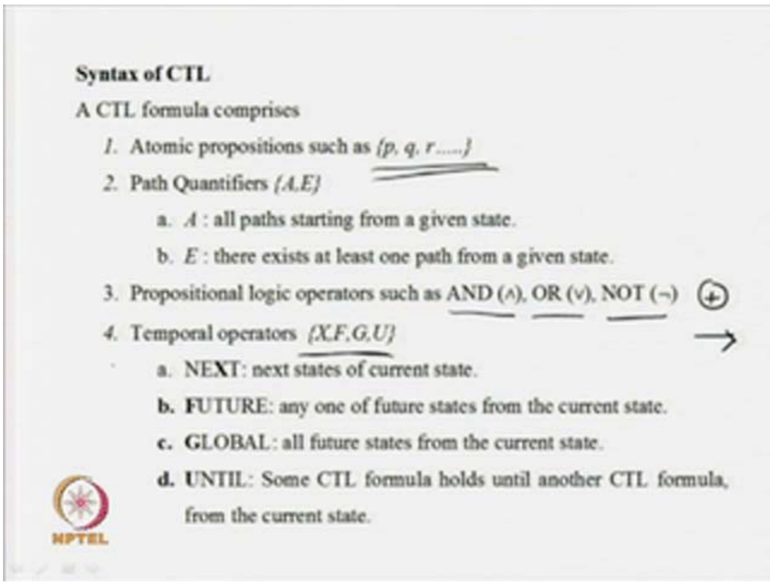


So, now you just select Computational Tree Logic it is a Branching Time Logic and we are going to define the meaning of this particular CTL or Computational Tree Logic with respect to a model. Now, just see in this particular example; say I am having this particular model M; it is having three states say S 0, S 1 and S 2. And, we are having some transition; just you consider this transition say S 0, it is going to S 2 by this particular transition. When we are having in your S 2, either we can go to S 3 or we can go back to S 0 again.

So, in this path we can have a transition like from S 0 to S 2, S 0 to S 2 like that. Now, in this case what happens; this is a graphic presentation. We can unfold this particular transition system and you can get a some sort of tree. Now, if unfolding of this particular situation you just see that in we are in this particular state S 0; depending on the scenario either we can go to S 2 or we can go to S 1. So, these are the next state behavior; so, this is your in S 0, we can come to S 1 or we can come to S 2. Now, when we are in S 2 again we are having two choices; either you can go to your state S 1 or you can go back to state S 0 again. So, behavior of this particular state S 0 and this particular state S 0 is almost same; all they are same basically apart from their timing stamps. If you can look into time stamps; in sometimes terms, t 0 we have got here then in t 1. I am going to be in this particular place, t 0 plus 1 and next time in terms in t 0 plus 2, we are in this particular place.

So, apart from this particular time stamp all the behavior of this particular state S 0 and this particular state S 0 is same. Now, in case of Computational Tree Logic we are going to region about this particular formula over a ; that is why the name coming as your Computational Tree Logic. So, eventually if we are having a state transition system like that, we can unfold to it a graph and we can assign the meaning of CTL with respect to this particular tree. So, we are going to say this is your Computational Tree Logic. Now, we are going to see how we are going to define the CTL formulas.

(Refer Slide Time: 07:15)

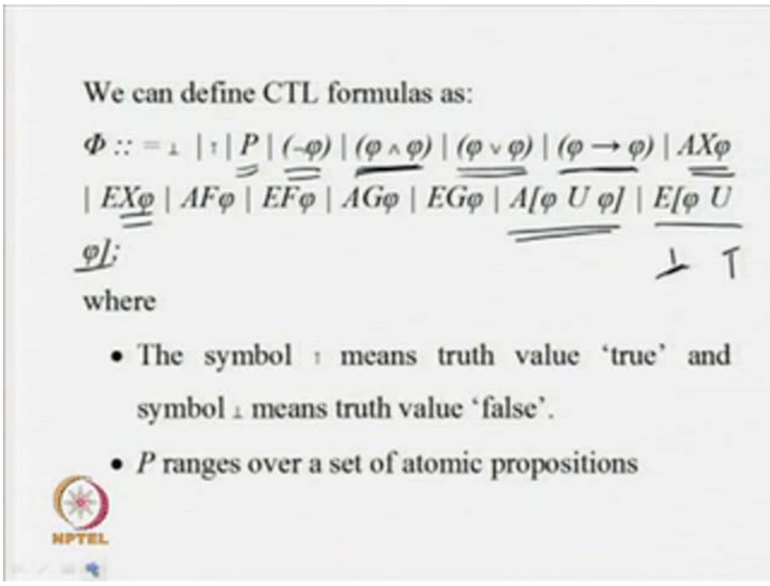


Now, first when we are going to look for a language or look for a logic, then we have to look for a syntax of that particular logic. So, what is the syntax of that particular logic CTL? So, when we are going to write a formula in CTL, then it involves several components; the first component is your atomic proposition. So, we are having a set of atomic proposition, these are represented as a say p q r s like that. So, this atomic proposition is going to take truth values either true or false, like our proposition logics. So, this is basically proposition that we have that truth values of this particular propositions are either true or false.

Next, we are having path quantifiers and the path quantifiers is basically represented A and E. A means that in all possible path; so, in a particular state if you go for all possible paths, then we will going to quantify this by this particular path quantifier A. Similarly, E is another path quantifier which as their exist a path. If you are interested for a particular path, then will look for this particular there exist quantifier? After that we are having all the propositional logic operators, that we have been propositional logic can be used in our temporal logic also. So, like that AND, OR, conjunction or disjunction, not negation or may be your explosive OR like that your implication; all those particular temporal propositional operators will be used in temporal logic also.

So, these are basically similar to your temporal propositional logic. Apart from that we are having temporal operator; we are going to use the temporal operator in our CTL. So, the temporal operators that we have discussed are used in this particular CTL also. So, these are the four temporal operators that we are having. X stands for your next step, F stands for in future, G stands for globally it holds and U stands for until operators. So, P until Q; so, it is a binary operator P remains to until Q become slow. So, these are the components that we are going to use while defining the CTL formulas.

(Refer Slide Time: 09:32)

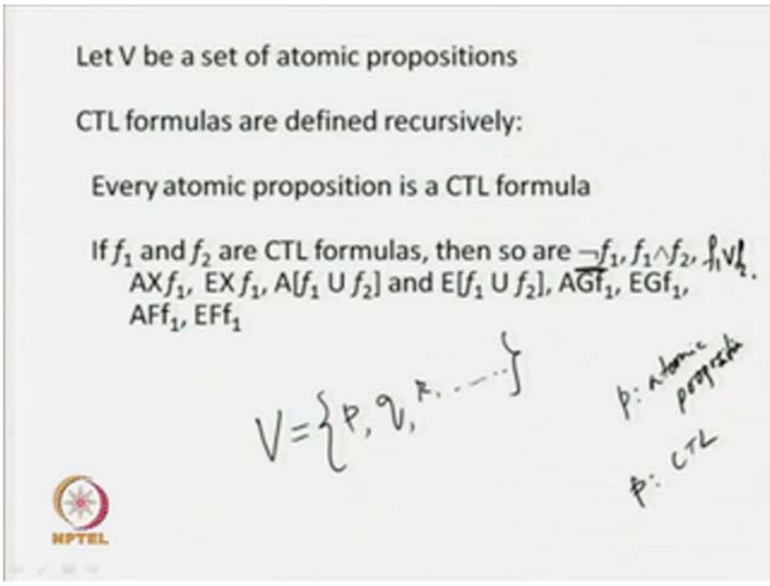


Now, in BNF notation we can define the CTL formulas like that. So, we are using two symbols; one is your bottom and second one is your top. So, this top and bottom are used as our CTL formulas; top stands for the truth value true and bottom means for the truth value false. So, we are having two truth values in our logic; one is your true and second one is your false. So, true is represented by symbol top and bottom is represented by the symbol false is represented by the symbol bottom. So, top and bottom are treated as your CTL formula. Then, next we are having P; all P will be treated as your CTL formula.

What is P? P is nothing but these are the proposition that we are to going use in our system. So, we are having some proposition; those propositions can take values either truth values, either true or false. All this proposition will be treated as your CTL formula.

Now, we can use the propositional connectives to form CTL formula. So, if phi is a CTL formula then not of phi will also be a CTL formula, phi and phi will also be a CTL formula, phi or phi will also be a CTL formula, phi impels phi will also be a CTL formula. So, all propositional connectives will be used to define CTL formulas from CTL formula. Apart from that we are having temporal operators. So, with these temporal operators are also going to form some CTL formula. So, the temporal operator next step. So, we are having next phi all temporal formula. If this temporal formula is specified by the path quantified A or path quantified E, then we are going to say that these are CTL formula.

(Refer Slide Time: 12:12)



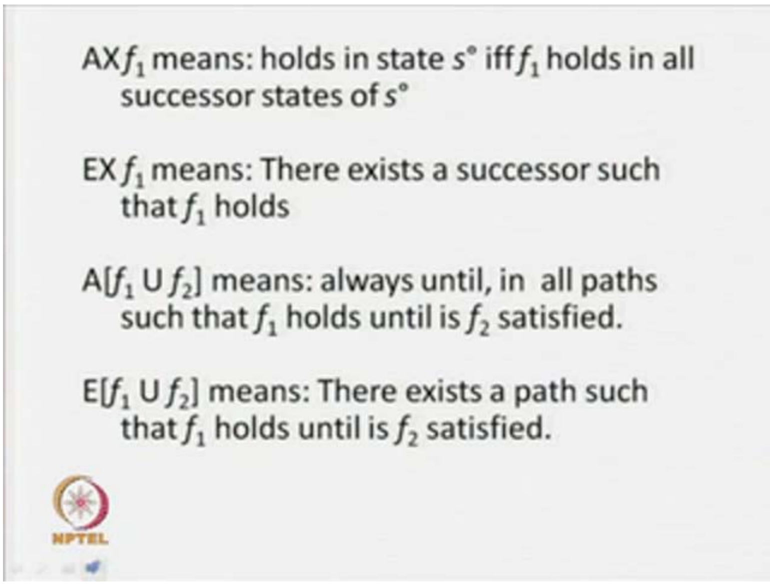
Similarly, for future operator we are going to have AF phi and EF phi; that means path quantify A and E we are going to get two CTL formula. Similarly, for globally also, G phi; so, we have going to have two CTL formula AG phi and EG phi. Next, we are going to have until operator which is a binary operator. So, if we are having two CTL formula phi and phi, then we can say that A phi until phi is also a CTL formula or E phi until phi will also be CTL formula. So, we can construct CTL formula using those particular rules.

If the formula confirms to these particular rules then we are going to consider those as our CTL formulas others will not be treated as our CTL formulas.

Now, in by looking into this particular BNF notation, now we are going to see what the CTL formulas that we are having are. So, first we are going to say that we are having a set of atomic proposition. So, V is a set of atomic proposition; you can say those p q r etcetera are atomic proposition. So, they can take truth values either true or false. Now, we can define the CTL formula with the help of this atomic proposition. Now, every atomic proposition will be treated as a CTL formula. Because in BNF notation also we have seen that all atomic proposition will be treated as a CTL formula. So, if p is an atomic proposition; so, in this case we will say that p is also a CTL formula. Now, for all atomic proposition we are going to get CTL formula.

Now, you just consider that if we are having two CTL formula, f 1 and f 2; because recursively you are going to define it. So, if we are having two CTL formula f 1 and f 2, then we are going to say that not of f 1 is also CTL formula, f 1 and f 2 is also CTL formula. Similarly, we can say that f 1 or f 2 is also CTL formula. Then, AX f 1 is also CTL formula, EX f 1 is also CTL formula, A f 1 until f 2 is also CTL formula, E f 1 until f 2 will be also CTL formula. So, similarly, AG f 1, EG f 1, AF f 1 and EF f 1 will be CTL formula. So, this way we can construct our CTL formula.

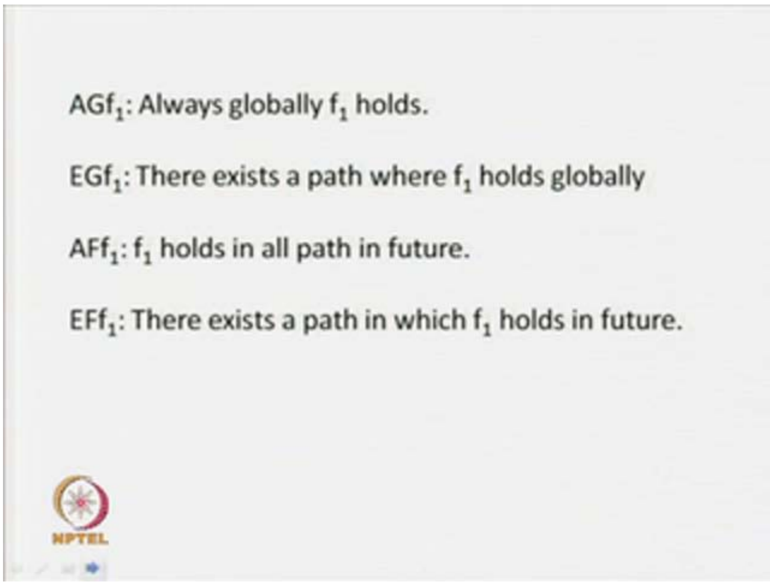
(Refer Slide Time: 14:00)



So, what would be meaning of this now? You just see that if you look in to this particular formulas will find that the meaning of those particular formulas. We propositional connective of very straightforward. Now, will see what is the meaning of those particular temporal operators? So, one is A X f 1; what does it means? That means, if we are going to look for a particular state it is having several possibilities in future in the in all those possibilities in next state; if f 1 holds then will said that A X f 1 holds at that particular point or particular state. That means, what we said that in a particular state A X f 1 holds if in all possible next state f 1 holds.

So, that is why this is says that holds in a state S 0; if f 1 holds in all successor states of your S 0. Similarly, E X f 1 it means the there exists a successor such that f 1 holds. So, it is having several possibilities in future but we have concern about a particular direction or in a particular future. So, if in a next state it holds then you can said that in the particular state EXf1 holds. Similarly, A f 1 until f 2 so, f 1 that until is a binary operator. So, we need to formulas f 1 and f 2 and that is quantify by path quantify at a in this it means that always until. That means, in all possible direction in all possible path f 1 must holds till f 2 holds. So, this is A f 1 until f 2. Similarly, E f 1 until E f 2. So, we are concern about a particular path or there exist a path but not any path any path we are looking into with one particular path. So, if we get such a path that f 1 holds until f 2 holds then we said that particular state E f 1 until f 2 holds.

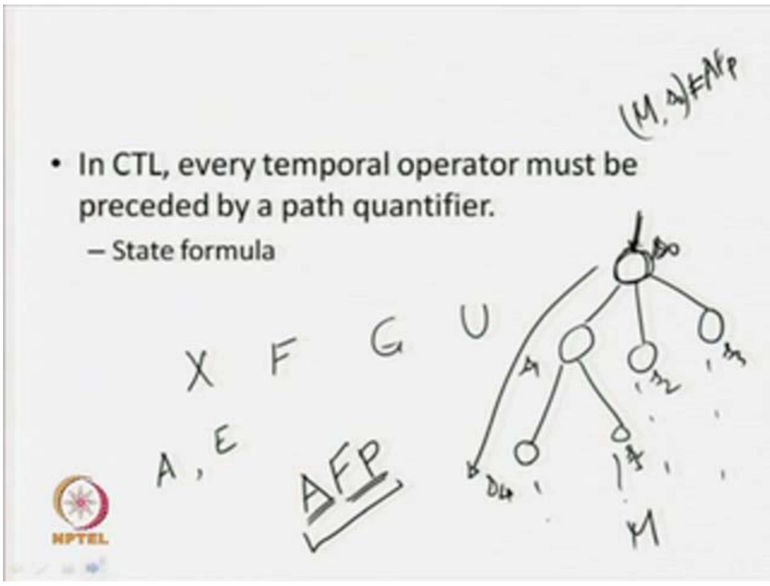
(Refer Slide Time 16:02)



A G f 1 this is always globally f 1 holds. So, if we are in a particular state wherever we go in all direction, in all path in exist that f 1 must holds then will said that A G f 1 holds. Similarly, E G f 1 this is constant to a particular path. So, we are having several possible place but we will look a particular path. And, in this particular path in all states f 1 holds then we said that in that state E G f 1 holds. Similarly, A F f 1 this capital F stands for a future operator or eventually operator.

So, that mean f 1 holds in all path in future. So, we should get future states where f 1 holds then we said that A F f 1 holds in those particular steps. So, you have to look for all paths. So, similarly, E F f 1 it looks for a particular path. So, if any path in future f 1 holds then we said a E F f 1 holds. Now, you just see that we are having these particular 4 temporal operators. And, these particular 4 temporal operators will be preceded by path quantity A and E. And, that means we are going to get a different combinations and we are going to get a difference temporal formulas. And, the meaning of those temporal formulas will be define over a model. Formally, you have going to define the semantics such I am giving a brief idea about it; how we have going to said that this particular formula is true.

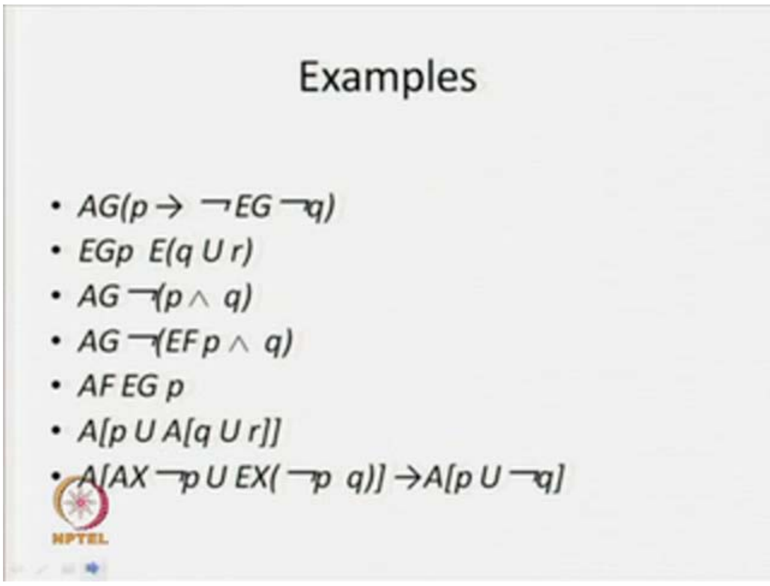
(Refer Slide Time 17:31)



Now, you just see that if you look into this particular syntax. Then, what will happen? You will find that in CTL every temporal operator must be preceded by a path quantifier. So, we are having say 4 temporal operator x future globally and until in CTL. This particular four temporal operator must be preceded by path quantifier A or E. Then, on you have to going to said that these are CTL formula. And, due to this particular quantification by path quantifier the CTL formulas are basically state formulas. We are going to define that truth value of CTL formula over a state. So, just simple like that I am going to said that if I am in a state S 0. So, it is having three possible polities in future this future say S 1, S 2, S 3. If you come to S 1 say I am having 2 different future possibilities. So, this is your S 4 and S 5 like that we can have future possibilities in all this direction.

Now, when we talk about CTL formula; that truth values of the CTL formula will be define over this particulars state. So, it is a state formula. So, any temporal formula says F P. What is it says that? In future p holds. So, this is a temporal operators, temporal formula and basically this temporal formula defines over a path with respect to a starting state.

(Refer Slide Time 20:06)



But one we are going to talk about CTL that means I can say that it will be A F P. So, this is the temporal formula f is preceded by this particular path quantifier. So, we are going to have this particular formula which is the of formula. And, the truth value of this particular CTL formula will define over this particular state or any state. So, if we talk about this S 0 set this is my model M. So, model M instead S 0 whether it models your A F P . That means, this CTL formula truth value of this CTL formula is the final word this particular state. So, CTL are yours states formula. And, what we can say which one is a CTL formula; if every temporal operator is preceded by a path quantifier then we said that this your CTL formula.

Now, after looking for the syntax of the CTL formula; let us see some formulas and let us say whether these are CTL formulas or not. So, you consider about this particular past formula. Now, what would happen? You can say that in this particular case we are having a set of atomic proposition p say p q r; these are setup atomic proposition. These are having the truth value whether true and false. Now, every see atomic preposition will be treated as a CTL formula.

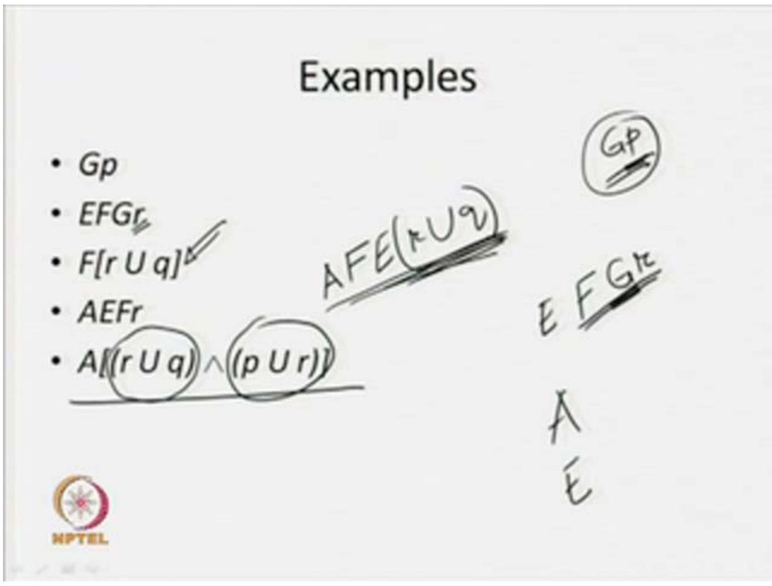
So, in this particular case I can say that q is a CTL formula; so, naught of q is also CTL formula. Similarly, p is a atomic preposition so, p is also a CTL formula. Now, G is a temporal operator it is presided by E. So, E G naught of q is also a CTL formula. So, if we are having as CTL formula; then what will happen? Naught of phi is also a CTL formula. So, in that particular case knot of E G phi is also a CTL formula. So, p is a CTL formula and naught of E G naught q is also CTL formula. These 2 CTL formulas are connected with the help of these particular implication operations. So, this whole set is going to give a CTL formula.

Now, this CTL formula is again quantify or again having a temporal operator of calls G. So, whether globally these holds are not again; this particular globally temporal operate is preceded by this path quantifier s. So, that means this whole formula is a CTL formula. Now, similarly, we can analyze each and every formula and you will find that these are a formula. So, here in the second example you just see that this is your q until r; this is until operator q and r are CTL formula. So, E q until r is also a CTL formula. Similarly, E G p so, p is a CTL formula. So, E G p is a CTL formula.

Now, if we look into it E G p, E q r then we will find a these particular operator E is not is a CTL formula but it is not directly connected to E because after p we are having E. So, in that particular case it is not a CTL formula; but we can make it CTL formula by putting some binary connectives or say proportional connective. If I said that this is odd then it becomes a CTL formula. So, E G p is CTL formula and E q and p are another CTL formula. Because this E until U temporal operator is preceded by this particular E quantify, path quantify. Similarly, G is preceded by your E path quantifier. So, this whole formula becomes a CTL formula; like that you can analyze other one also. Say the fourth one I am going to say p is a atomic preposition.

So it is CTL formula; f is a temporal operator and it is preceded by path quantifier. So, E F P is also CTL formula, q is also is a CTL formula because it is a atomic preposition so they are connected by this particular conjunct. So, this whole thing is also CTL formula. And, already I have said that if phi is a CTL formula then naught of phi is also a CTL formula. So, naught of E F P and q is also a CTL formula. And, this CTL formula is again having up temporal operators G which is preceded by A that means in all path globally weather these holds are not. So, this whole formula is also a CTL formula. So, like that we can say each and every formula and will find that these are formula.

(Refer Slide Time 24:06)



Now, look for this another set of formulas the formula is G P. So, in case of your G p ; p is a atomic formula. So, it is a CTL formula G is a temporal operators. So, G p is a temporal formula but this temporal formula is not preceded by any path quantifier A and E. Since, it is not preceded by any path quantifier. So, this temporal formula is not a formula. So, we are not going to said a this is a CTL formula.

Similarly, second example r is a CTL formula; since it is a atomic preposition. So, F G is a temporal operator. So, Gr is a temporal formula FG are now F is a temporal operator, G is a temporal operator and r is a a CTL formula because it is a atomic preposition. So, this particular temporal operators Gis not preceded by any path quantifier directly you are getting apps. So, these F G r is a temporal formula no doubt but it is not a CTL formula. Since, it is not a CTL formula so; E F G is also not a CTL formula. So, to have or we said that it will be a CTL formula provided its temporal operators is preceded by your path quantifier. And, with the help of this path quantifier we are making or we will said the CTL formulas are state formula. We are going to talk about the truth values of a CTL formula over a state.

Now, similarly this next formula if you see that r until q. So, until is a yours temporal operators r until q is a temporal formula because is a until operators. So, these temporal operators are not preceded by any path quantifier. So, it is not a CTL formula but if we said that E r until q then it becomes a CTL formula because this temporal operator is your proceed by u path quantifier.

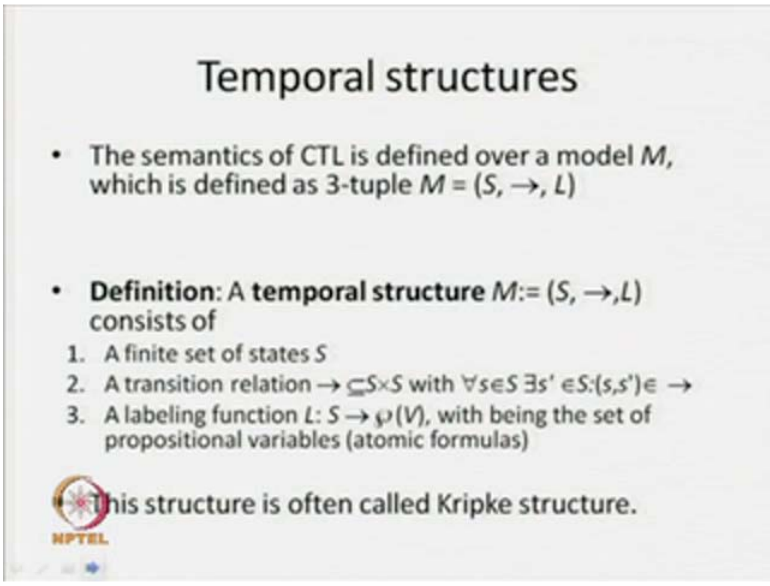
So, F r until q is not a temporal formula but we are saying that this is a temporal formula. But since it is a temporal formula so, I can say that F and this temporal formula. If I write this thing then it will become a temporal formula but it is not a CTL formula now. Because this particular temporal operator F is not preceded by any path quantifier but I can make it a temporal formal CTL formula by placing a path quantifier in front of it. So, this whole formula becomes CTL formula acts as this particular third formula is not a CTL formula; but it is a temporal formula. But if I put path quantifier in front of every temporal operator then it becomes a CTL formula.

So with similar argument we can said this equation 4 or say it is some 5 are also not a CTL formula use you can find it this may be a CTL formula. Because it is preceded by your this particular path quantifier A. But if you look component words first look into this particular components p until r. This is a temporal formula but it is not CTL because this particular until is not preceded by any temporal path quantifier. Similarly, r until q is also temporal formula but this is not a CTL formula because this until operator is not preceded by any path quantifier. So, that so, this since these 2 components are not a CTL formula. So, they are conjunction will not also be a CTL formula. So, in that case we will said a this is not a CTL formula.

So, now we have seen the syntax of our CTL Computational Tree Logic. We can use atomic preposition all atomic preposition will be treated as our CTL formulas. CTL formulas can be connected by your prepositional connective like An, OR, XOR implication etcetera and those will become form CTL formula Again, if we are having any CTL formula they can we construct another CTL formula by temporal operators; but each and every temporal operators mass be precede by a path quantifier. Then, only we will said a this is a CTL formula; without pressuring by a path quantifier we are not going to get CTL formula but will may get temporal formula. So, CTL is a sub class or may temporal formula. So, with path quantifier we are making CTL formulas. Secondly when we are going to say that path quantifier; basically we are going to say that path quantifier a S say that in all possible path form a particular step S or E says that they are exist path from a particular step S.

So, in that particular case this CTL is suited as our state formulas. We are going to look for the truth values of a CTL formula which respects to states; but which should have in either all possible computational path or there exist a computational path. This is the meaning of ANE. So, with the help of path quantifier ANE we are making them the state formula.

(Refer Slide Time 29:32)



Now, we have to look for the semantics on meaning of these particular CTL formulas. Already, we have mentioned that if we are going to look for a truth values of a temporal formula that is always define over a model. So, for CTL also we have to define the meaning of CTL formulas over a model. So, in this particular case we have going to have a model M call M and this model is having 3 components. So, this model M is define as

S this arrow and l. So, we are going to have a model which is a 3 tuples; basically we having 3 basic components. As we are talking about one arrow and we are talking about L. With the help of this we are going to define a model and in this particular model we are going to define a meaning of our CTL formula

Now, what are the 3 components? Just see the formal definition of this particular temporal structure. So, we said that a temporal structure M is having a 3 tuple where we having a 3 component S arrow can help with S is a set of finite states; that means we are having states in this states is a finite state. So, we are having finite number of steps in our states. So, we are representing F by s. The second one that arrow that we are representing it is a transition relation.

And, transition relation is nothing but a subset of Cartesian product of S n S. That means, we are having a transition from one set to the other set. So, that is why what we have saying that it is a sub set of this Cartesian product S and S. So, Cartesian product S and S basically give need a components like that S 0 and S 1; say if S 0 and S 1 say belongs to here say s then S 0, S 1 will be a component in our this Cartesian product.

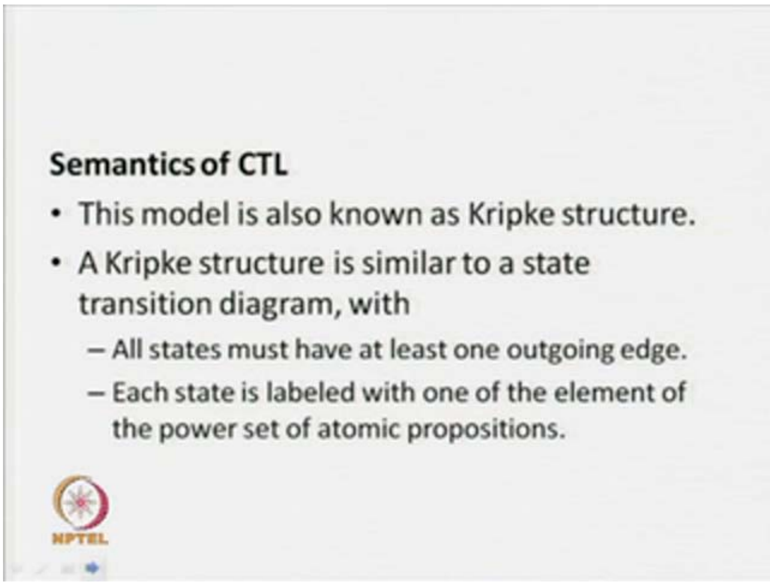
What does it means that? That means we are having a transition form S 0 to S 1. So, the second component is a transition relation; which is subset of the Cartesian product of S and S with a particular restriction it says that or S belongs to S. There are existence is such that s as this belongs to this particular transition relation. Now, what does it means? So, if we are having a set formed at particular set state we should have a transition to some other states. That means, it says that transition relation is complex. So, this is one restriction the CTL structure that we are defining over here; now transition relation must be complete.

That means, if you pickup any random any state as it must have a success or state. If some states are not having success or state then we are not going to treat these things as a CTL structure. Along with that we are having a labeling function which is known as basically L definable it is nothing but s arrow rho of v. Basically, what is v here? V is basically this is set of atomic proposition and rho be basically says that power set of this particular set of atomic proposition. So, will keep on a example.

So, the states will be label by the set of atomic proposition and what is saying that set of proposition variable and is a atomic formulas or atomic variable. So, what basically this labeling function says? It says that this particular atomic proposition is true in these particular states. So, we are having a labeling function because we will know since it is the atomic proposition. We will know the truth values of those particular atomic proposition either they will be true or they will be false.

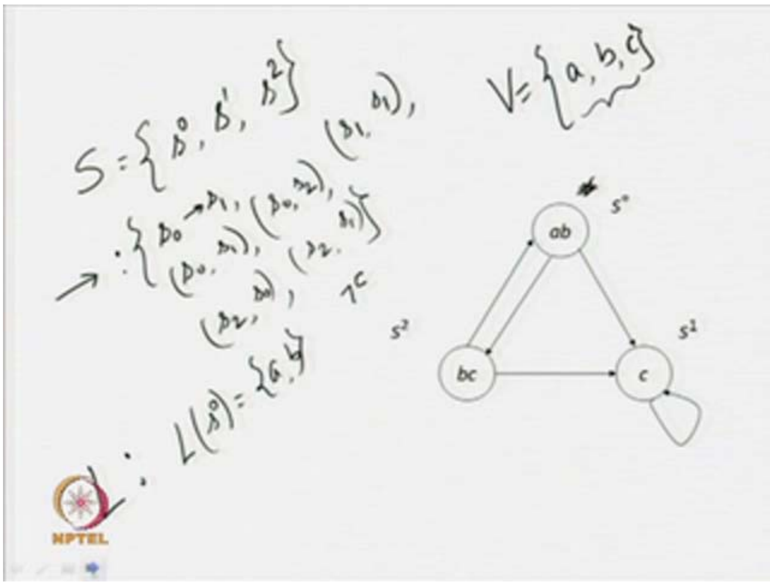
So, if a particular atomic proposition is true in a particular state. We label this particular state by that particular atomic proposition. And, this structure is often called Kripke structure. So, basically the semantics of your CTL formula is define over Kripke structure; we use the term Kripke structure. So, this is a Kripke structure which is having this 3 component as transition relation as and L. And, what is the basic requirement? One is your transition relation must be complete for every state there should be a success or state.

(Refer Slide Time 33:58)



So, we are going to define it over a Kripke structure. And, if you look into the behavior of the Kripke structure we will find that it is nothing but very much similar to your state transition diagram; it is a state transition diagram. And, in this particular state transition diagram we are having a 2 restriction; one is the all states must have at least one outgoing edge that means every state must have a success of state. And, secondly we are having a labeling function the state will be label by the atomic proposition. Why this particular atomic proposition is true? We will just see one example then it will be clear.

(Refer Slide Time 34:38)



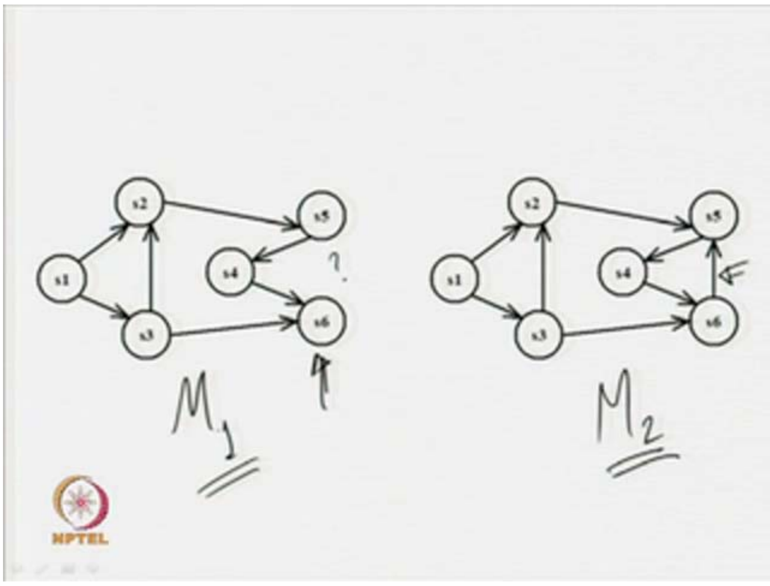
Now, you are see that come back to this particular transition diagram. So, it is a finite state transition diagram because we are having the state S. What are this particular states? We can said a this is your S 0, S 1 and S 2. So, we are having 3 states along with that we are having the set of atomic proposition v. Here, we are having the state of atomic proposition or that atomic proposition or the atomic proposition that we have a; a b and c. This atomic proposition gets 2 equalizer either true or false; along with that we are having this particular transition functions. So, here what is the transition function that we have? We are having a transition from as S 0 to S 1. So, we are having S 0 to S 1. This is a transition on the notation I can say that since I am saying details subset of Cartesian product I can say S 0, S 1 also I can write like that.

Again from S 0 to S 2 I am having another transition. So, I can say that that transition set up S 0 to S 2. Now, when I come to S 1 we will find that we are having a transition from S 1 to S 1. So, S 1 to S 1 we are having another transition. Now, only one transition we have. And, similarly, when we come to it is S 2 then we find that? From S 2 we are having a transition to S 0 and from and again another transition we are having form S 2 to S 1.

So, this set of transition relation basically conference this particular 5 transition 1, 2,3,4,5. Now, it should satisfy one condition that this particular transition relation must be complete. That means, every state must have a successor state. Now, if you said that a if you look into state S 0. We will find that it is having at least 2 success states. If you look into S 2 again it is having 2 successor states. If you look into S 1 it is having a self look that means we are having a S 1 as its own successor that means it is complete. Now, along with that should have a labeling function L. So, what is this labeling function? It says that we are going to label each and every state by the power set of this particular set of atomic variables. So, in a particular set if the atomic variable is true then we are going to labeling by this particular atomic process. So, here it says that L of your S 0 is basically nothing but a and b. So, what it says in when we are coming to this particular state S 0 then this atomic proposition a and b is true.

And, indirectly it says that in this particular S 0 c is not true. That means, you can say that not sub c that means we can say again label if by naught of c also it says that; that means naught of c is not true in this particular state. But in conjunction what we say that if it is not labeled it any atomic proposition. We say that these atomic propositions are false in this particular state.

(Refer Slide Time 38:22)

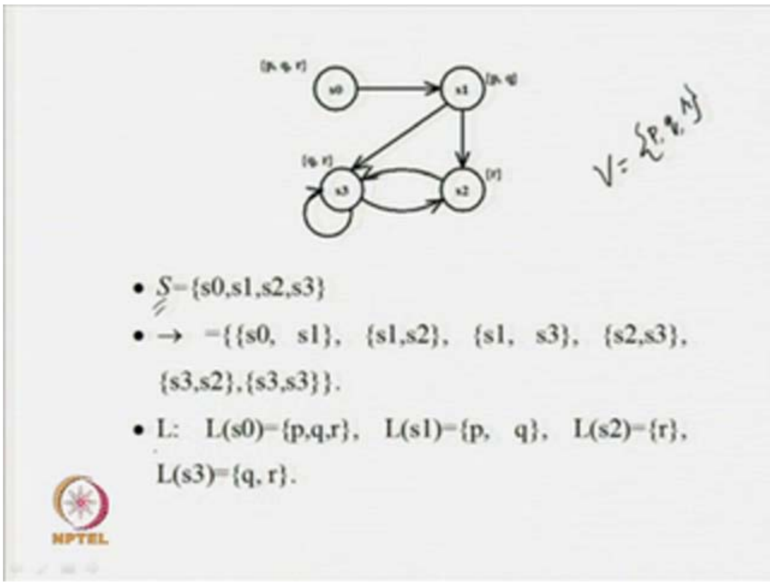


Similarly, these particular state S 2 it is bracket b and c that means a atomic position b and c are true in this particular state S 2; but atomic proposition s falls over here. And, in S 1 it is labeled it see so; it says the atomic proposition c is true in this particular state S 1 but atomic proposition b and c are falls over here. So, if atomic proposition is falls is never marked it but internal meaning is the these atomic proposition falls in this particular state. Because atomic proposition can have 2 tools equalizer either true and false. If it is true we are labeling it; if we are not labeling it says the this atomic power will this false in this particular state.

Now, just look into this particular model. So, I am saying this is M 1and this is M 2. What are these 2 models are Kripke structure? So, what are having that similar structure it is having similar step. And, we are to looking about the transition only labeling function we are not showing it over here. Because we are not talking about this set of atomic proposition but if you look into this 2 models M 1 and M 2 whether they are Kripke structure.

If you observed it you will find that both are a identical but in M 2 we are having 1 extra arrow, extra transition. These transitions are not present in your M 1 whether this is going to contribute something over here. If you observe it then in M 1 is we find a this state S 6; it is not having any successors state these are say that this is no outgoing S from this particular step S 6. So, since it is not having any successor that means this transition relation is not complete. So, you are not going to treat this particular m is a Kripke structure. So, M 1 is not a Kripke structure.

(Refer Slide Time 40:04)

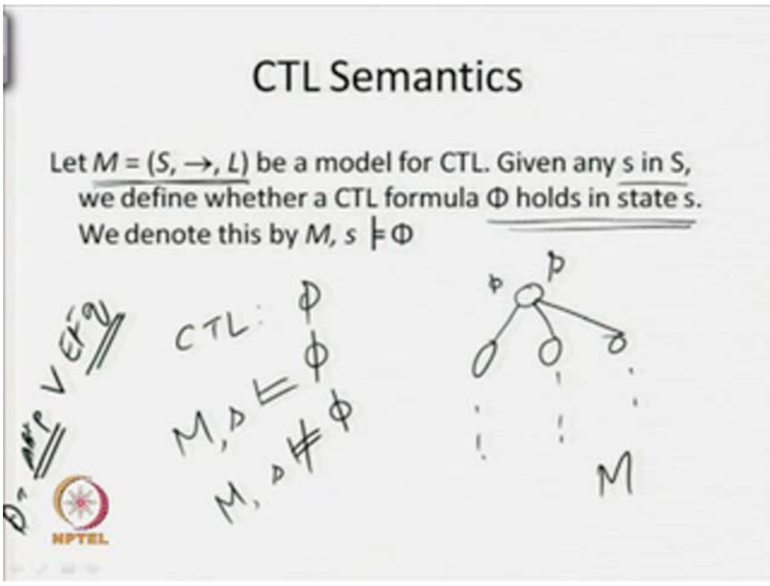


Similarly, M 2 now if you look into it in all the step in find at we are having transition or we are having successor state. So, M 2 is a Kripke structure. So, we have to careful about it; it is not like that in finite state model. We can define a meaning of CTL formula it must be Kripke structure. And, what is the basic requirement of Kripke structure? That transition relation must be complete for each and every state we must have a successors state. So, here M 2 is a Kripke structure; M 1 is not a Kripke structure.

Now, look into another example say already I have explain this things. Now, by observing it also you will find that this is a Kripke structure because if you look into all the states. All states are having you the transition or that mean you are having a successor. Now, what are the 3 basic components? We having set of state so, this is a S 0, S 1, S 2, S 3 this is a set of state. And, the transition relation how many transition we are having? 1, 2, 3,4,5,6. So, we having the 6 components these are transition relation. And, this is the labeling function you such say that we are having this label over here. That means, you are having a set of atomic proposition and the atomic proposition that we are having here is your p q and r. So, S 0 is labeled it p q r it says the all p, q, r is 2 over here. So, label of S 0 is p q r in S 1 q and p and q is 2. So, label of S 1 is p and q l label of S 2 is only r because r is true and level of S 3 is q and r.

So, I think with this particular explanation I think the notion of Kripke structure is clear about it because it is having 3 components S the set of states conjunction relation. And, the labeling function and what is the labeling function? Already, I have explained.

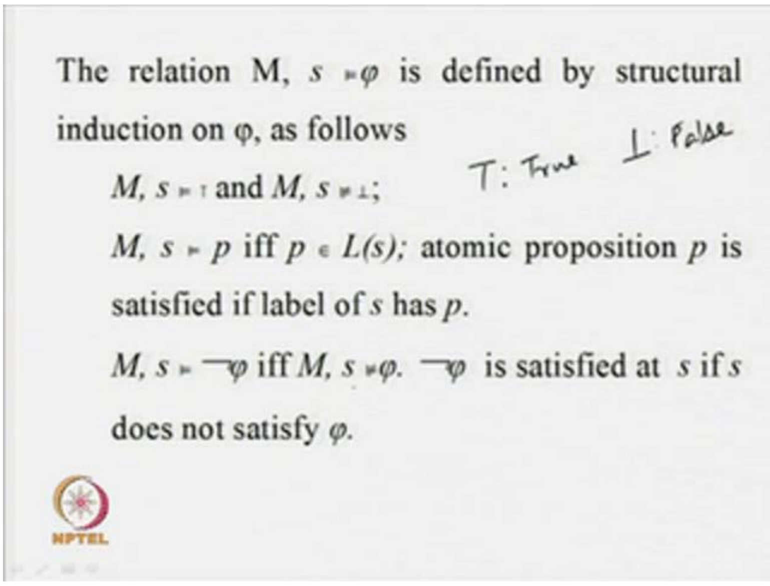
(Refer Slide Time 41:33)



Now, we are going to define the semantics of a CTL formula. So, it is like a already temporal formula I have said. Now, in CTL formula also we said at we have going to consider one particular model M which is having these components set of states conjunction relation L be a model of CTL. Now, given any state S of S. Now, you consider any state S of S we define whether CTL formula phi holds in this particular state or not. And, we donate we denoted by M s phi. That means, if I am having some sort of your model M. And, we are going to consider one particular state as we are having a CTL formula say phi will say that M s models phi is says that this particular CTL formula phi holds this particular state S.

And, similarly we use this particular symbol also M s does not model phi. In this particular case what it says that? You will state as your formula phi does not holds. So, basically we are going to set up phi holds in a particular state, phi does not hold in a particular state. So, in this particular state if phi holds in a particular state sometime you can have a label of phi over here? And, the meaning of those particular CTL formula will be define on the structure of this particular CTL formula; because if we are going to talk particular CTL formula it may have sub formulas also.

(Refer Slide Time 43:56)



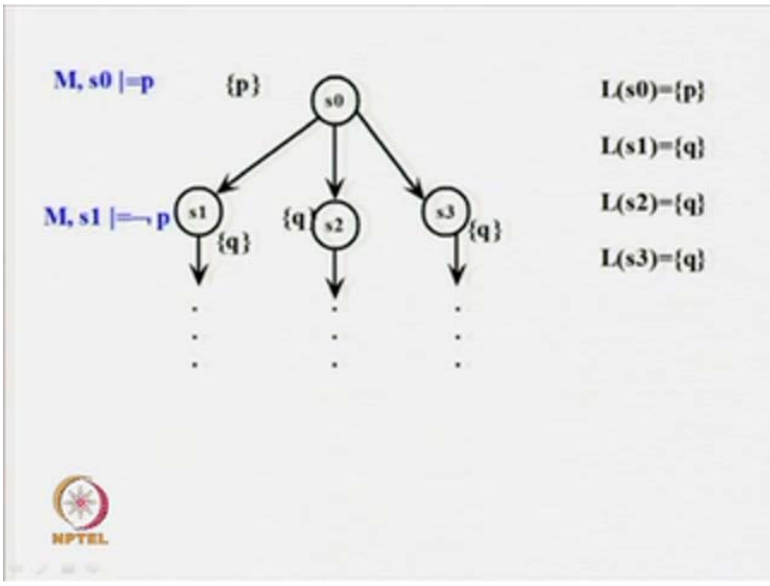
So, basically indirectly we have going to define a meaning on this particular structure of CTL formula. Consider one particular formula say A F P or E F q. That means, this is one CTL formula, this is another CTL formula. First we should have the label this particular CTL formula in a state. Then, if E F q holds in state S then we can say that a F P or E F q also holds in this particular states because either of this one component is to in this fellows. So, if we look for one particular formula or we have to look for this particular component was I am after it is going to look for the entire formula.

Now, that is so I am saying the relation this M s model phi is defined by structural induction of phi as follows. So, in general what happens? We have going to said that in every state it will be mark by this particular tool fellow true. And, it will not mark by the tool fellow false because this top represent the tool fellow true and bottom represent the tool fellow false. So, this is basically top is your true and bottom is your false. So, by default every step will be mark with a tool fellow true and none of them will be mark with false. That means, it says that incredibly you can say that true is true in all state and false is false in each and every state.

Another one we are going to said a second rule M s models p. If p is the member of this particular labeling function of this particular state. That means, we are going to talk about the truth values of atomic proposition. If the state is labeled with the help of this particular atomic preposition then we said a these particular atomic preposition is true .

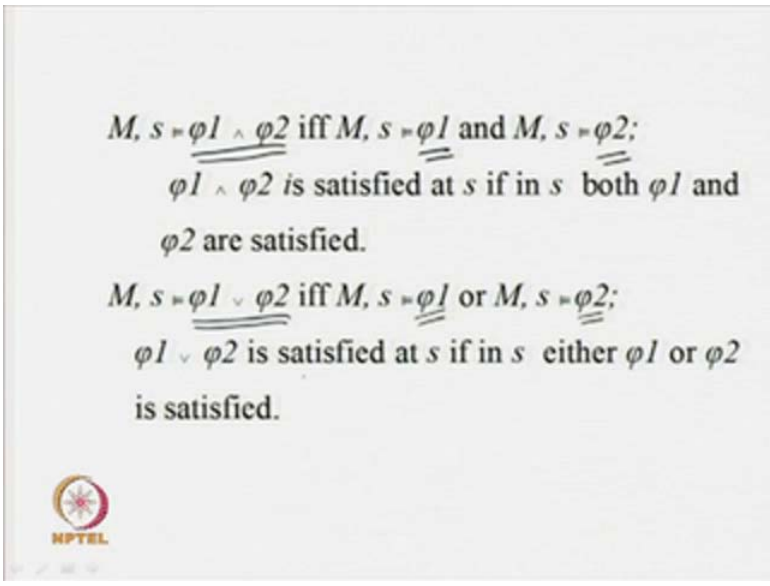
Similar, if we said the a particle state s of a model M is models naught a phi. That means, so, the negation of this phi is true over here. Basically, it means that if does not this particular state as a model M does not models this particular of phi. So, if this does not model phi you can marked with naught of phi also. So, already I have mentioned that basically negation of atomic proposition we are not going to marked by default is here. But in some formula we have to marked if it is CTL combination of CTL operators because it is will come as a sub formula of another formula. So, we need the have the truth values of this particular negation. So, if it is not 2 it your naught of phi it means that phi is not hold in this particular state. So, these are the 3 basic components that we are having.

(Refer Slide Time 45:55)



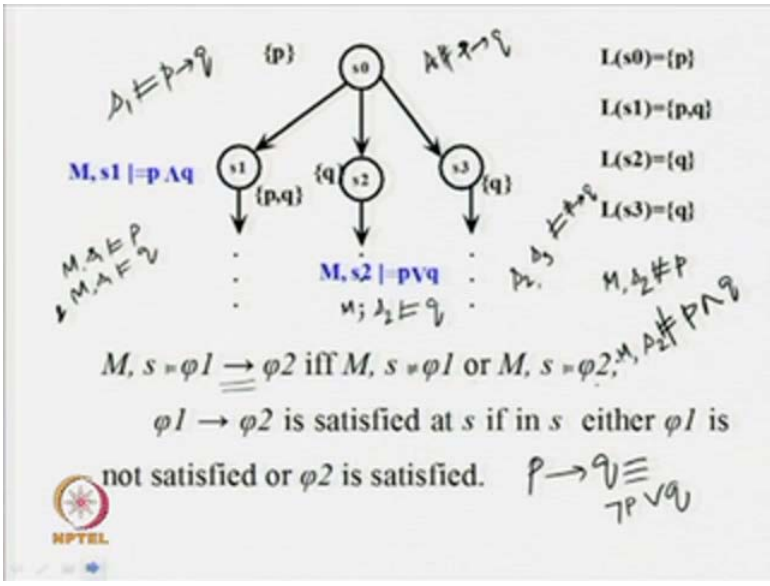
So here you just see that we are having an model. So, it is a 3 structure we have one folder model. So, S 0, S 1, S 2, S 3. Now, if you look into this is a labeling function that we are having p, q, q, q. So, M of S 0 models p because it is labeled it be we consider it is models p. Similarly, I can say that m of S 1 models not of p because it is not marked of this particular atomic proposition. So, in state S 1 it models naught of p .

(Refer Slide Time 46:29)



So, similarly now you can look for that true fellows of your this conjunction phi 1, phi 2; whether in a model m in a particular state s whether it model phi 1 and phi 2 over here. If holds provide at phi 1 and phi 2 independently holds particular states. So, you said that phi 1 and phi 2 satisfied in this particular state; as provide at these state models phi 1and models phi 2. Similarly, we can look for a the junction phi 1 or phi 2. So, this since this is distinction. So, what will happen? We said that in particular state s phi 1 of phi 2 holds provide at either phi 1 holds in s of phi 2 holds in s. So, these are the basic operators that we are having. So, this is your phi 1or phi 2.

(Refer Slide Time 47:36)



So, now you are look into this particular again that model that we are having S 0, S 1, S 2, S 3. And, we are having these are a trees like structure and these are a labeling function that we have. Now, if you concern about this particular state S 1 will find that M S 1 models p and q because this S 1 is labeled with p and q. So, basically we can say that M S 1 models p and M S 1 models q see both p and q models in state S 1 we can say that in S 1 p and q models.

So, that means p and q holds in this particular state S 1. If you come to this particular state S 2; you will find that it is labeled at q that means atomic proposition q is over here. So, you can say that in M S 2 models p or q because you just see that here either p or q suite holds in this particular state. So, here we will find that M S 2 models q; since M S 2 models q. So, we are going to said that M S 2 models p or q either p or q but now if you look into that state S 2. You will find that M S 2 does not models p because it is not labeled to it your p; since p S 2 is not labeled it p. That means, is not in this particular labeling function that means p is falls the atomic proposition p is falls in this particular state S 2.

Since, p is falls so, it does not model p. So, if you look into this combination say p and q in this particular state S 2 of model M. Then, what will find that? M S 2 does not model p and q because if models q but it does not model p. So, that is why you have saying that m S 2 does not model q; just you say the meaning over here. You will say that s models q 1 and q 2 provide at M s model phi 1and M s model phi 2. So, both must be to over here. So, since in this particular state it does not model p. So, you can say that M S 2 does not model p and q. So, similarly now we come to this particular connective say implication phi 1 implies phi 2 whether in a particular state M model M in state s whether in state s phi 1 implies model phi 2 or not. So, in this particular case what will happen? We will say that this state s models this phi 1 implication phi 2; if M s in state it does not model phi 1 or M s model s phi 2.

So, what we are going to say that if whether model s phi 1 does not model phi 1or does not model s phi 2. So, basically we say that phi 1 implies phi 2 satisfy whether as if n either phi 1 is not satisfied a phi 2 is satisfied. So, either of these 2 conditions. So, basically we know that if you are going to talk about your some formula say p implies q this is equivalent to naught of p or q.

So, we know that these particular implication sign is equivalent to naught of p or q. So, in this particular case we said that if p then q. So, if p is true then q must be true if p is fault we are silent. So, in this particular case if you looked at p is true over here but it is not marked with q. So, this particular S 0 does not model p implies q but in this 2 state S 0 and S 3; we will find at it is marked with q but it is not marked with p. So, it is p is falls over here but will said that if p than q. If p is falls here these for a about it. So, in this particular set S 2 and S 3 there model here p implies q. So, here you can find that S 2 and S 3 both models here p implies q because q is to over p is fall at s does not models q 1 or s models q 2.

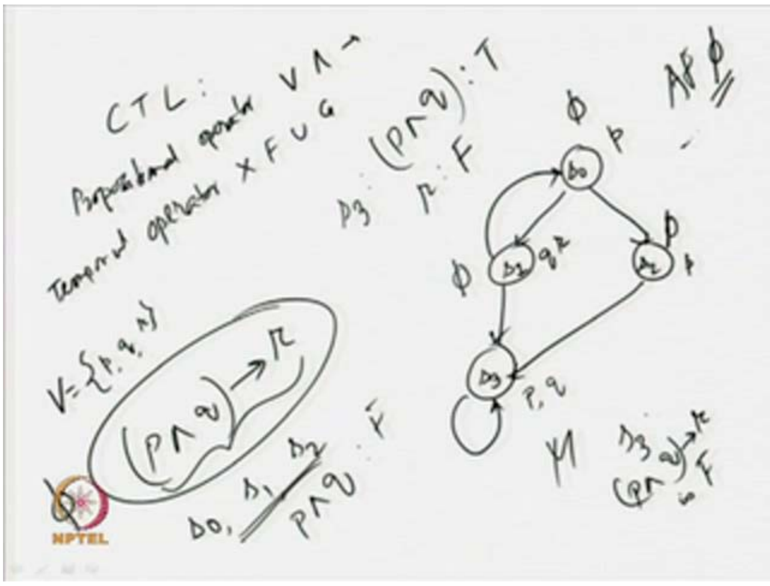
If you come to this particular one both p and q . That means, in this all condition phi 2 is true over here. We can say that in S 1 also if models p implies q. So, this is the way that we are going to look further 2 values of our CTL formulas. Now, till now we see that we are discussing about CTL formula; we have define the syntax of our CTL formulas.

Now, we are going to define the meaning of our CTL formulas. And, what are the operators that we have in our CTL formulas. Basically we are having all the proportion connectives AND, OR, NOT, XOR implication etcetera. And. along with that we have a temporal operator and if we are using temporal operator; those temporal operators must be preceded by our path quantifier. Then, we are going to have CTL formulas otherwise we are going to have temporal formulas only. And, when we are going to talk about CTL formula that 2 fellows of define over a state that means these are basically state formulas.

Now, when we are going to define the semantics. We are saying that CTL formula some meaning of CTL formula will be defined over Kripke structure. We have seen what is Kripke structure? It is similar to finite state but we are having 2 additional components. The common one is the transition function must be complete. And, second one we are having the labeling function it is state should be labeling by the state of atomic proposition that are true in this particular states.

Now, we have seen how to define the meaning. So, in this lecture we have talk about the meaning or how to assign the truth values temporal for CTL formula with respect to our this particular propositional connectives. So, we have seen AND, OR, negation and your distinct implication and like that. Now, you have to see the how to going to assign the truth values of our temporal operators? So, next class we are going to discuss about this things. So, up till now we have talked about the truth values of our this particular propositional operators only. So, next class we are going to talk about our something called what we called say we are going to look for all the temporal operators.

(Refer Slide Time 55:07)



Now, you just see that now what happens? We are talking about said CTL formula it is having propositional operator like that OR, AN, implication. And, temporal operator like next stepped, future until and your globally. Now, today you have seen the meaning of those particular prepositional operators only. So, next class we are going to talk about this particular how to define a meaning of this particular temporal operator?

Now, you just take a simple model just say the this is my 1.(No Audio From 55:50 to 56:01) So, whether this is a Kripke structure if you look into it you will find that we are having these particular 4 state S 0, S 1, S 2, and S 3. Now, if you look into your transactions relation you will find that every steps are having a success of steps. So, that is why the transaction relation is complete. So, we can say that this model M is a Kripke structure. Now, we just said that we are talking about the state of atomic preposition state p, q, r. Now, the labeling function we can say that p, q, r; p, p, q. Now, if I give as a temporal formula p and q implies say R. So, if you look into this particular temporal formula whether this formula is true. In this particular in which are the states this particular formula is true because we will find that every where this particular component in state S 0, S 1 and S 2; this component p and q is your false. You said that p is true but q is false, q is true but p is false, p is true but q is false. So, p and q is false in this particular 3 states. So, since this is false so, p and q implies r will be true in this particular 3 states. Now, when you come to this particular state S 3. Then, we will find that p and q is true over here but what is the state of your R. You will find that R is false so, true implies false. So, in this particular case we will find that in state S 3 the given formula p and q until R is false basically because we so see that p and q is true but r is false. So, true implies false which is basically false.

Now, what will happen? Now, if we are having these particular 3,4 states they are label with this atomic proposition. Now, when you going to look for this particular temporal formula. Then, we will find that this particular formula say if I said that this is your phi; then what we can say that this phi is true in state S 1, S 2, S 0, S 1 and S 2 but this not true over here. So, like that what we can say that? Now, we know this particular component is true now, If I said that going to say that A F phi; that means we know the level of this particular phi it is level in this particular step. Now, we can look these particular truth values of this particular formula. So, prior to that we are going to see how we are going to define the meaning of f. So, in next class we are going to talk about or we will going to discuss about how to define the meaning of this temporal operators along with path quantifier in our Kripke structure.

Thank you.