INDIAN INSTITUTE OF TECHNOLOGY GUWAHATHI

NPTEL

NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Prof. Jatindra Kr. Deka
Department of CSE
IIT Guwahati

Module IV: Temporal Logic

Lecture V: Equivalence between CTL Formulas

NPTEL Phase-II
Video course on

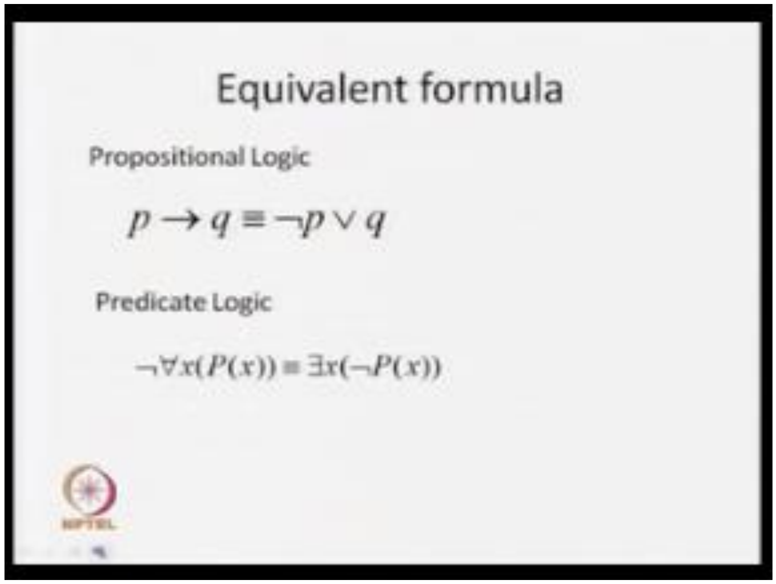
Design Verification and Test of
Digital VLSI Deigns

Dr. Santosh Biswas
Dr. Jatindra Kumar Deka
IIT Guwahati

Okay, till now we have seen about temporal logic and we have particularly discussed one kind of temporal logic which is called CTL computational tree logic. Till now we have seen what is the syntax of CTL and how to define the semantics. So while defining semantics we need a model and the meaning of those CTLs formula we define over a model which is basically known as your Kripke structure.

Now today we are going to see some equivalence between CTL formulas, because in logic we have equivalent formulas, in CTL also we are having some equivalent formulas.

(Refer Slide Time: 01:10)



Such as a technique as we just said that in proportional logic we are having equivalent formulas and we say that this P implies q is equivalent to knot of P or Q. So what does it means, it says that for same assignment approved values for P and Q both these formula will give me the same truth values.

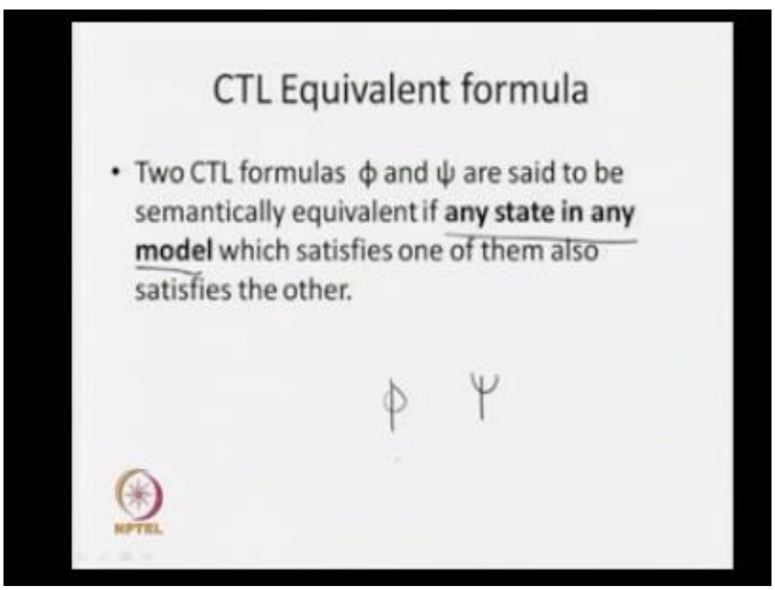
If one will give me truth value true then second one will also give me truth value true, if one would give me truth value false then second one will also give me truth valve false. So it should be true for all possible assignment for P and Q. Now here since we are talking about only two variables that means we can have that variables P and Q there are four different combination maybe both are yours 00 or I can say that true, true or I should say that instead of 0 I will say that this is false and 0 I will say that this is false.

So it is all same true, true and false and true and true these are the four possible combination we have. Now if you see this thing then for all possible combination both the formula will give us the same truth values. In this case w will say these are the equivalent formula in your proposition logic. Now similar notion of equivalent formula comes in our predicate logic also. In predicate logic we are having this particular quantifiers for all X and there exist X.

So one common or basic notion of equivalent formula here we are going to get knot of for all X Px is equivalent to there exist somewhere for which knot of Px is true. Okay, so basically Px is a predicate it says that for all X Px is true and the lesson says that we do not get any X for which the X is true. So that means we may have, w should get some X for which knot of Px is 0. So this is the notion of your equivalent formula in our predicate logic.

So like that now we have to see one will say the true formulas in CTLs are equivalent. So what is the notion of equivalent formulas in CTL, so you can see that two CTL formulas ϕ and ψ so we are going to consider two formulas ϕ and ψ. So the truth values of these two formulas ϕ and ψ will b defined with respect to models. So we are going to say that these two CTL formulas ϕ and ψ are said to be semantically equivalent if any state and any model.

(Refer Slide Time: 03:43)

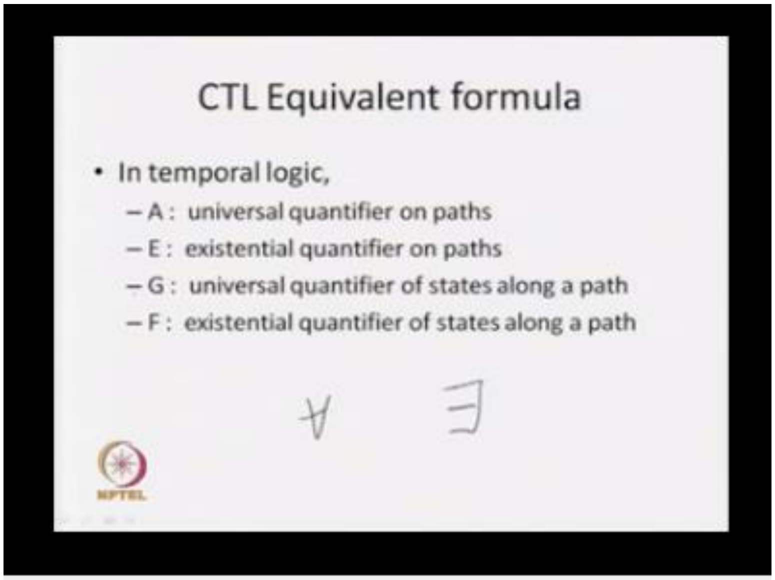


Now this is important any state in any model which satisfies one of them also satisfy the other. So that means if you are say that two CTL formulas are equivalent they must have the same truth values in all state of any model okay. So if we are going to consider any state if ϕ is true then ψ

must be true in the particular state of any mode. So if you do look for any models then both must be true in any state or both must be false in any state.

Then we are going to say that these two formulas are equivalent. Now we know the syntax about CTL formulas, we know the meaning or semantics of CTL formula now we will see what are the equivalents that we have in our CTL.

(Refer Slide Time: 04:31)



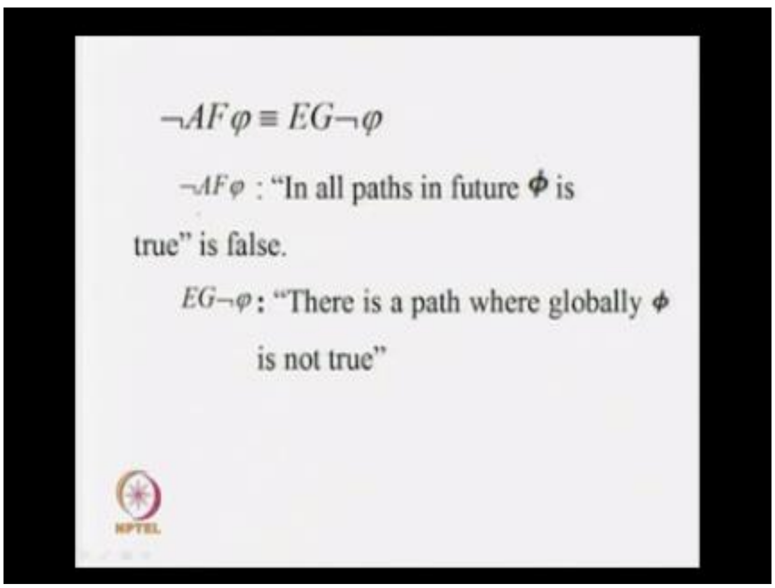
So in temporal logic basically you see that we are having some temporal operators and some path quantifiers. So it respective these are your CTL, so in CTL we are having A which is called universal quantifier on paths that means we are going to rejoin about all paths E existential quantifier on paths that means we are going to look for any paths, these are the two path quantifiers.

Similarly we are having two temporal operator one is G and second one is F, what G says that this is the universal quantifier of states along a path. So basically we will say that G global it is true in all states in a path and F we say that we are going to get the states in future where some formula or some CTL formulas are true. So in that case we can say that F can be treated as our existential quantifier of state along a path.

And G is the universal quantifier of states along a path. Now you just see that, here we are having two quantifier A and E which is basically across path and F and G which are basically states along a path. So these two things can be again you can see that we are having that for all quantifier in predicate logic and there exist quantifier in our predicate logic and we know that they are having some relationship.

Okay with negation we are going to get the other. So similar notion should be available with our this temporal path quantifier as well as that state quantifier AE and FG. So for that we are going to get some equivalent formulas. So we are going to see those particular equivalent formula.

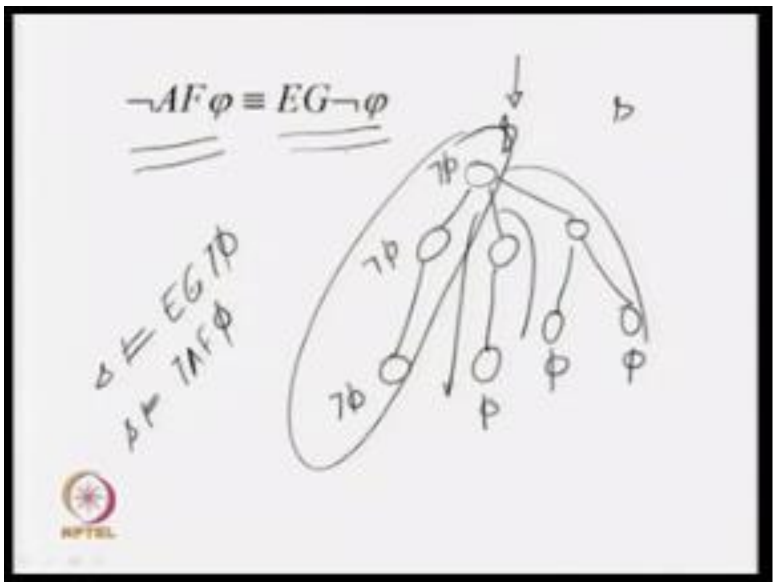
(Refer Slide Time: 06:11)



First equivalent formula that is coming over here is like that knot of AFϕ is equivalent to there exist a path globally knot ϕ just see what it means. Knot of Aϕ we know that Aϕ in all part in future ϕ is true. So if knot of this thing is this particulars that means is path, so in all path in future ϕ is true is basically false because we are having this particular indication symbol in front of this particular CTL formula.

Now second path, second formula it says that EG knot of ϕ here is the path where globally ϕ is not true, so we are going to get one path where globally ϕ is not true. So now see that you are saying that these two are equivalent how we are going to get the equivalent with the help of small example I am going to see that it says that in all path it future ϕ is true is false.

(Refer Slide Time: 07:09)



So basically just see that we are going to define the semantic of a model so we take some model something like that okay. So this is a simple model now we have to see that in all path in future ϕ holds so that means we should get some state in future where ϕ is true. And we are seeing that this is not true that what means in all paths we are not going to any future state where ϕ is true, less it is not true at this one of the path.

So basically if we say that if I am ϕ is true over here, ϕ is true over here, ϕ is true over here then in all this three paths you are going to say that AFϕ is true, but in this particular paths we are not going to get any state where ϕ is true. So in this particular case what I will say that AF Ø is not true in this particular state say S to FM we have start now this is equivalent to your saying that they are exits a part 12 globally¬ Ø so I say that this is in¬Ø.

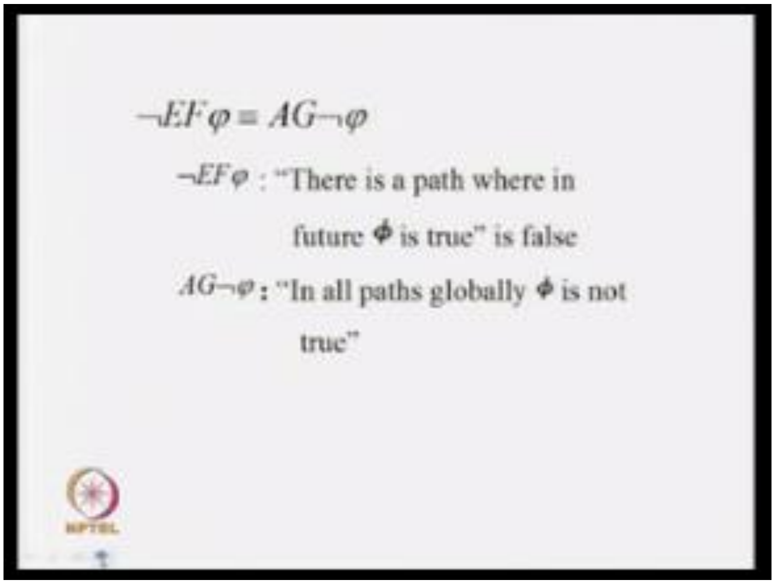
Basically you say that since¬Ø holds may be AF Ø may not peep through or the F Ø may not be true each what will happen in these order two state if Ø holds whatever then F Ø holds in this particular path so¬Ø must be true operator.

So similarly in S0 also not¬Ø must be true so what it say that basically we are going to get one particular path where globally¬Ø is true so since in this particular state diagram in state S I say that these models they are exit a path globally¬ Ø holds similarly you will find that in this particular state S¬of AF Ø holds basically we are getting one path where in further Ø is not true so¬of AF Ø is true in this particular state so like that in this particular state these two formula are equivalent both are true.

So like that we will say now if you are going to make up any model if you will find that EG¬Ø is true in a particular state always you are going to find that ¬of AF Ø is true so that is why you are say that these two are equivalent you just see that path quantifier A which your universal quantifier it become the existential quantifier across path and the existential among states in a path becomes the universal quantifier along that particular path so they are dual of each other so in this case I am going to say that these two formula are equivalent.

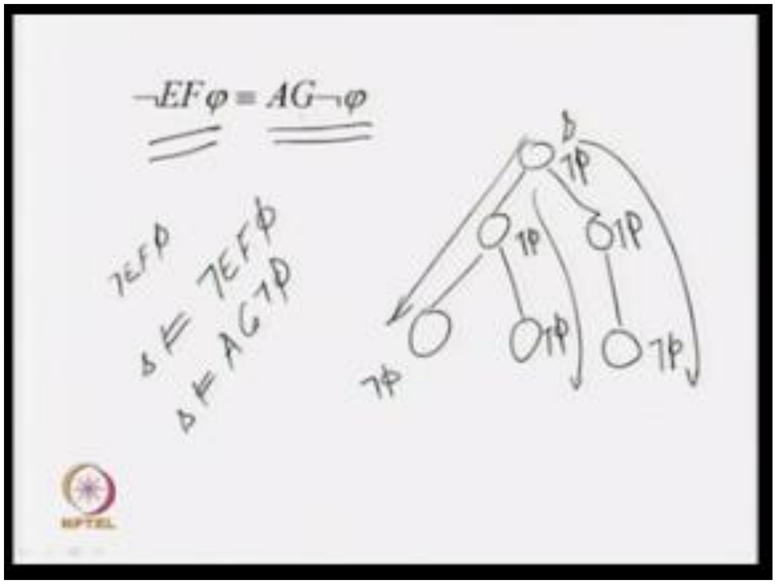
So this is the first equivalent formula that we are discussing now say what are the other equivalent formula we are having.

(Refer Slide Time: 10:07)



So this another equivalent formula we are having they are exist a path in further Ø holds and negation of this is true and this equivalent true in all path globally¬of Ø holds so what it say that EF Ø so DI is path where in further Ø is true and negation change that this is false basically and AG¬ of Ø it say that in all globally Ø is not true and we are going to say that these two are equivalent of course in that you can get the internal feeling or intermediately you consider that S it is true because in we are does not exist any path where Ø is true and it is happen that in all path globally¬Ø must be true.

(Refer Slide Time: 10:58)



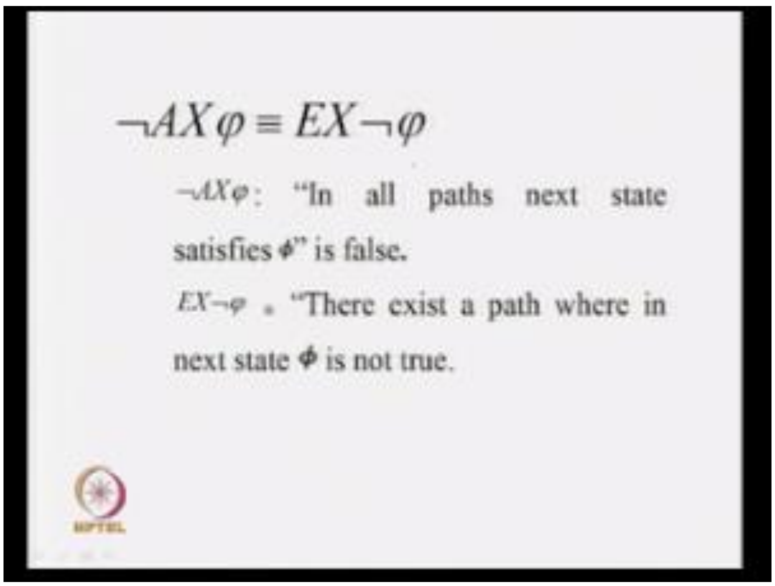
Again just come back to look into some model because we are going through have defined the meaning of this CTL formula of a model so you just see that we are going to say that there exist a path in further Ø is true and the negation of this one is true in this particular state and AG globally¬of Ø so in all path globally¬Ø say even I am going to say that¬ Ø is true in those particular state okay.

That means in this path¬of Ø true in this path¬of Ø true nut if I am going to say that Ø is true over here then what will happen we are going to get there are exits a path in further Ø is true in this particular case that means¬of there exists a path in further Ø is¬two along this path to become to have this particular formula through over we will see that¬of Ø over here then only we will say that in this particular path also we are not getting sate where in further Ø is true.

So in this particular state S I can say that¬of EF Ø is true because there does not exist any state where in further Ø is true okay so this by looking into this particular model we have found that again ion S model is your AG¬Ø because¬Ø is true in all those path so if you consider these two formula and you take any state of any model you construct any model you look for nay state if 1 is true in particular state of that model we will find that the other one is also true in that particular model so also I will say that these two are equivalent okay.

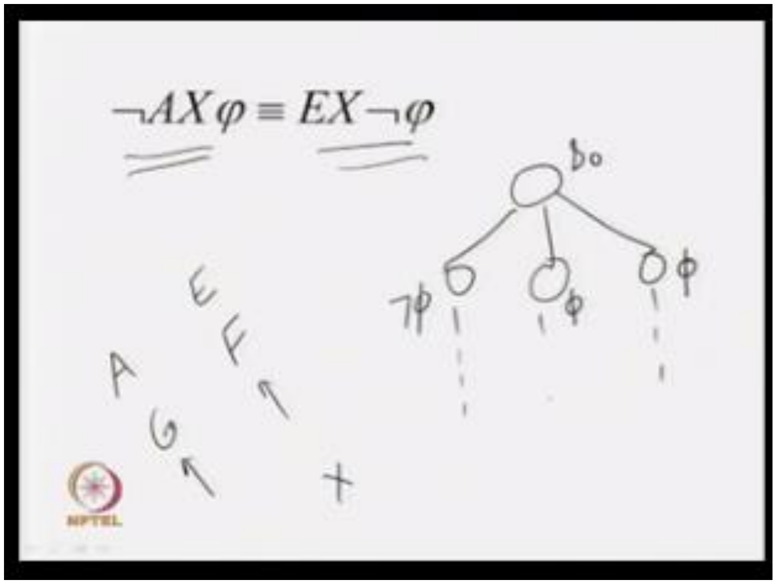
So again you will say that they are having a relationship between the existence quantifier along path and it is universal quantifiers along that was a path and this is your existential quantifier of state and universal quantifier of state along a path so they are having relationship.

(Refer Slide Time: 13:15)



Now another one in case related to your next sate operator so what is the next equivalent formula we are getting that⌝x Ø is equivalent to EX⌝ Ø so first formula say that in all paths next step satisfies Ø is false that is means in all path that Ø is not true in next state so that means second formula say that they are exits a path say in next state⌝Ø is true that means Ø is not true and we find that these two are equivalent again just look in to particular.

(Refer Slide Time: 13:51)

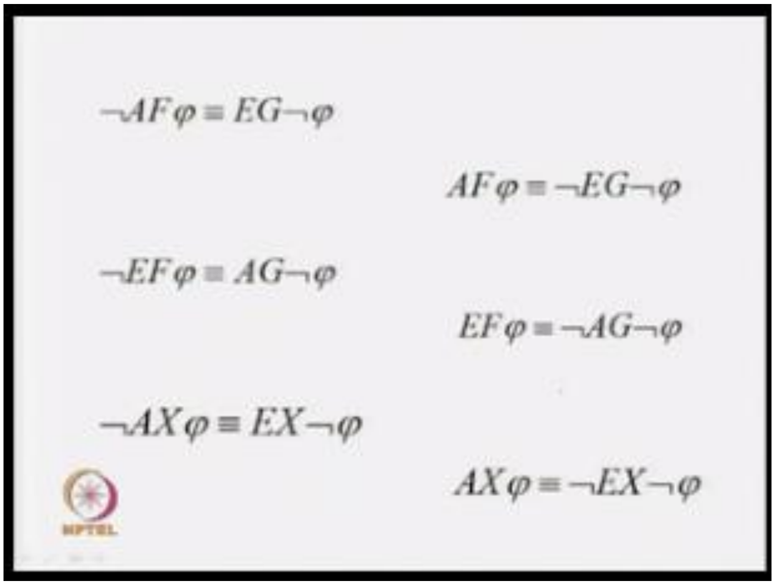


Model say if I am having a model something like that so in the particular state so in all path next state Ø is true and indeed this is not true in this particular state S0 then I can say that Ø is true in these two states but Ø is not true over here so in all path in next state Ø is not true and this equivalent to you can say that we are going to get there exist a path in next state ¬Ø is true again.

Now if you are going to look any model and look for any states of any model you will find that if 1 is true in a particular state then other will also be true in this particular state so that is why we are going to say that these are equivalent formula now what we have seen now that we are getting that path quantifier A and E and your that state quantifier we can say that state temporal operator this is basically globally and you are in further

So this two can be treated as your universal quantifier and these are basically existence or quantifier and by looking into this formula we will find that 1 is that dual of matter so if A and G are there can be represented by E and F but in next state it is dual of itself because next is going to represent by next of water only but A becomes new when we are going to take the recursion of this particular formula so that is these are the equivalent formula that we are getting in your CTL formula.

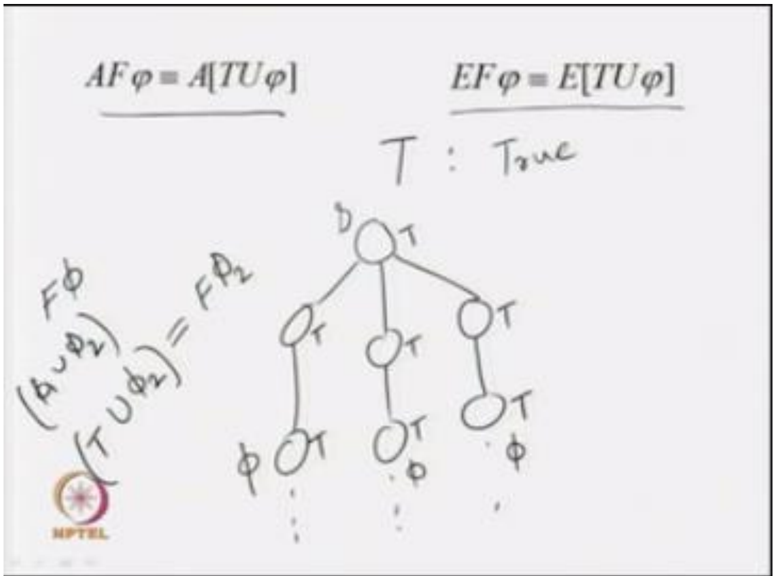
(Refer Slide Time: 15:37)



So in natural we can summaries it like that this is once equivalent that we have got ¬of AFØ equivalent EG¬of Ø so if you take the negation of these two formula then you will find that AFØ will be equivalent to ¬ of EG¬of Ø that means that formula AF can be replaced or expressed by EG similar second one that we have seen¬of EF Ø will be equivalent to AG¬Ø so if you take the negation in both side we will say that EF Ø EF Ø will be equivalent to¬AG¬of Ø and third one which is related to your next operator so¬of AX Ø will be equivalent to EX¬ of .

So in that case we cloud take the negation in both side we will say that AX ø will be equivalent to¬EX¬of Ø so basically we say that A and F can be replaced by E and G or E and F can be replaced by A and G so E will be replaced by A and F will be replaced by G so this is the things that we have seen till now.

(Refer Slide Time: 16:48)



So these are basically we are going to talk about the operator globally further and next state we are having another temporal operator which is your until okay now we will see whether some of this particular operator can be expressed with the help of until operator or not until is your binary operator which involves to a CTL formula and others F C and X are uninor operator now you just see that we are having equivalent with the respect to your until operator one we are going to set at AF Ø is equivalent to A T until Ø or EF Ø will be equivalent to E T until Ø.

Here that symbol T that I have replaced it basically it is that top it says that the truth value true now well you say that when we have we defined a symmetric of your CTL formula what we have seen that all state will be me marked by this two symbol to and none another state will be level by truth symbol false so what does it means true is true every where this is the motion that we have while depending our symmetric so if we take any model say these are the model we having also computation three we have.

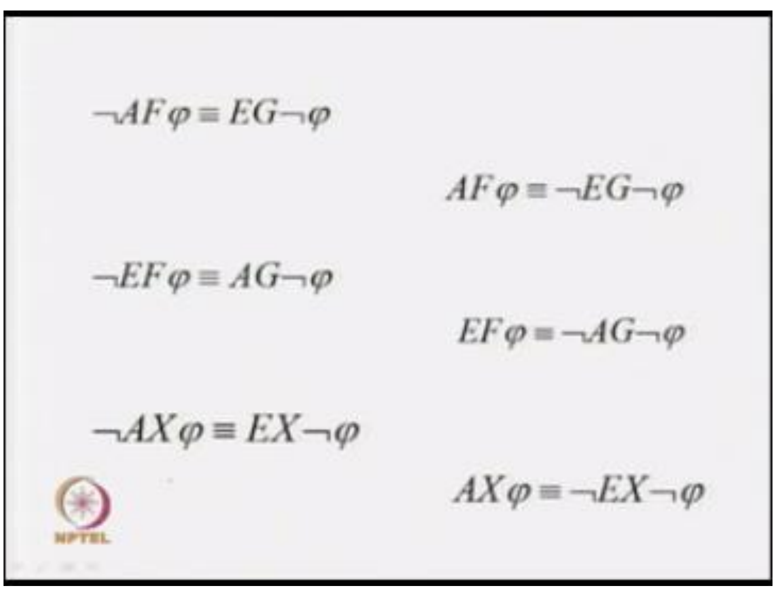
Now if we look into this particular model that means 2 is to everywhere that means all the states will be model by this particular truth symbol 2 so that means it will be level by these things we will say that true is true in all state now what is the further operator F Ø it says that in a path we should get some states in further where Ø is true so that means if you consider this particular path that Ø if Ø is true over here then we will say that F Ø is true in this particular path.

So similar if it is true in all those particular path then we will say that AØ is true in this particular state S and what is the until operator if we say that Ø1 until Ø 2 it says that Ø1 remains to until Ø2 becomes true okay so in this particular case now I replace this Ø 1 by say T top or the truth symbol 2 and until Ø 2 so now I am going to have these things that means 2 is to everywhere so true until Ø so this is nothing but we are going to look for some state in further where Ø 2 is true.

So this is basically nothing but we can say that F Ø2 so that is why we are saying that the further operator can be replaced with the help of your until operator so AF Ø will be equivalent to A true until Ø and EF Ø will be equivalent to E true until Ø so it is there exist a path in φ of ϕ to we will say that we exist a part to remains to until ϕ becomes true, okay.

So we will consider these two as our equivalent formula, and similarly Aϕ will be equivalent to A1 until ϕ okay, so now you said that the temporal operator F can be expressed with the help of temporal operator until so these are another two equivalent formula that we have.

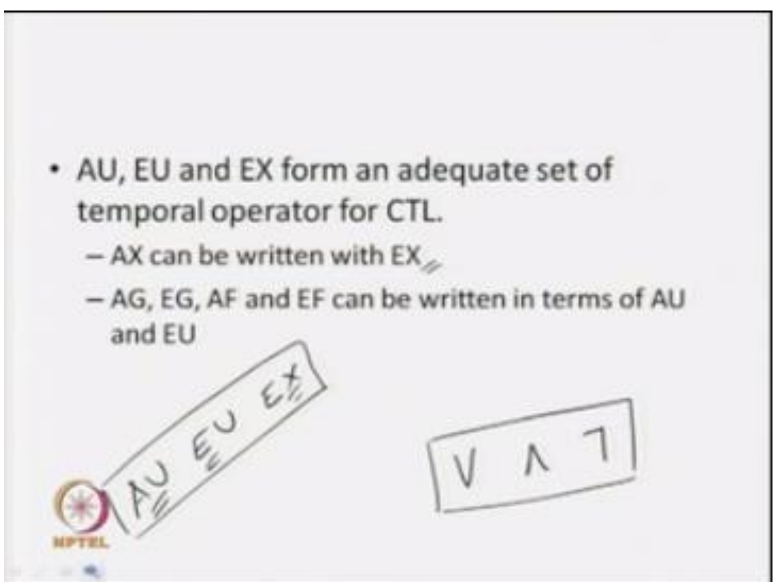
(Refer Slide Time: 20:31)



So all that is we have seen the equivalent formula which are related to AF is G and AX EX and now we are getting two equivalent formula which are L until an φ operator. Now if you look into CTL computational three logic we are having four temporal operators as we have discussed four temporal operator and all those temporal operator will be presided by part quantifier either A or E so in this case we will see that we are getting where we are having all together take define combination.

Now we need method for all those particular eight different combinations all a define CTL operator to check the truth values of a formula in a particular step. But since we have those particular equivalent it may happen that we inner to loop for the procedure for all eight combinations or all eight operators CTL operators we can come to your restrictor set and after the other operators will be expressed with the help of those particular restrictor. So in that particular case what we can say that.

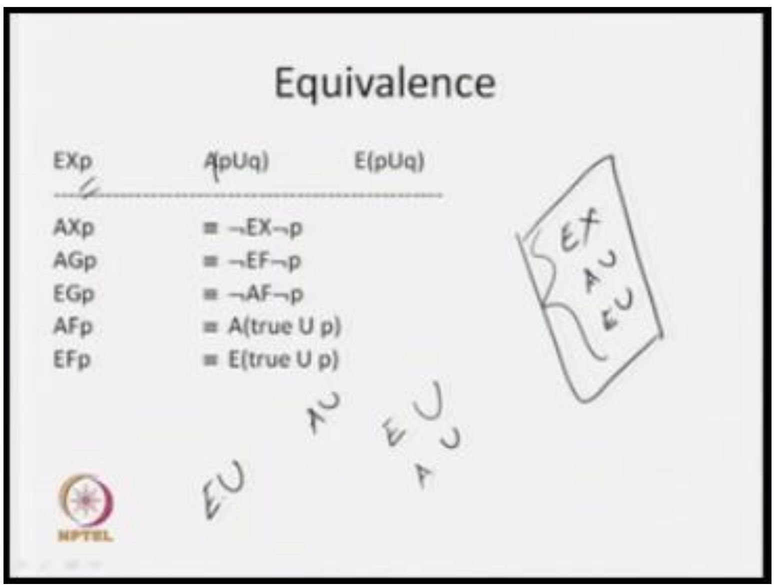
(Refer Slide Time: 21:42)



AU, EU, EX form an adequate set of temporal operators for CTLs like your propositional logic also classical logic what I have said that these junction conjunction and negotiation form say do you have save the operator because with the help of these three operator we can express another operator, like that in your CTL also now what we are going to say that AU in all part until operator EU in all part until operator and EX this three forms an adequate set out your CTL operators.

Because other operators can be expressed with the help of these three operators like that say AX can be written with EX, so I am having EX very well I can go for AX. Similarly AG EG AF, EF can be written with the rein terms of your either AU or EU, okay. So basically that is why we saying that these three are sufficient off to express all the eight combinations that we have discussed till now.

(Refer Slide Time: 22:52)



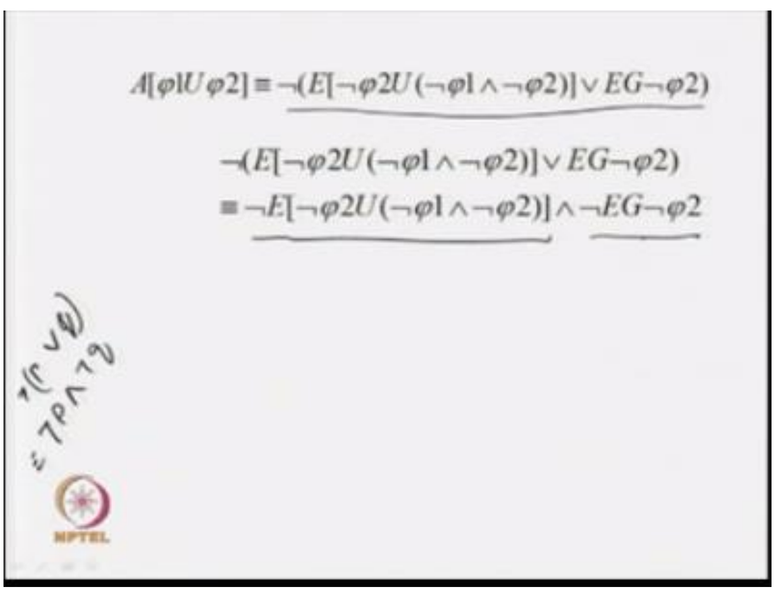
So how this is simple example that I am saying that I am going to take only Exp, Ap until q and EpUq so this is the adequate set of operator EX, AU and EU now since I am having total eight combinations so while I will have, I am having this another five combination over here so AXp will be nothing but that we have already seen the equivalent so you can say that this is not of EX, not of p, AGp will be equivalent not of EF not of p okay.

So AX is represent by EX, EGp is nothing but not of A F not of p so that G operator is replaced by F again this F will be replaced by your until operator say AF is your A true until p and EFp is equal to E true until p, so eventually all those particular eight can be expressed with the help of this three operator. So you can say that these are the adequate set of operators that we have and with the help of these tree operator we can express other operators also.

But this is not the only adequate set of operators for CTL formula we have some other adequate set also, but how we are going to get it we have use just see that with the help of this things I need what AU and EU so in until operator I need both the combination EU and AU so that is why we need these two things along with that we are having this particular either EX or AX we are going to take.

But somehow if you can see that EU can be expressed in terms of EU or AU can be expressed in terms of EU then what will happen in that case I can just omit one of this two combination then if we can omit one this two combination then we are going to get sum or the adequate set of CTL operator, okay. Now what it is possible or not let us see.

(Refer Slide Time: 25:04)



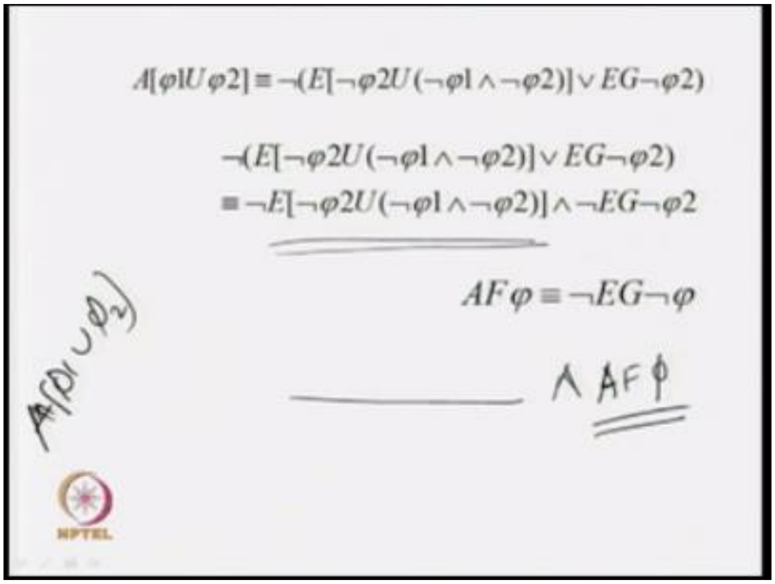
So we having one result this is says that if this is until operator A until this A until can be expressed with the help of your E until this is E until what it says that A φ1 until φ2 say if we are going to look for this particular formula so we are having an equivalent of this particular formula it says that negotiation this whole formula what is the negotiation of E we are exist the part not of φ2 until this conjoint not of φ1 and not of φ2 or E is the not of φ2, okay till now we have discussed some of the equivalent formulas and by looking into the construct of the equivalent formula it is AF that is they are in the equivalent and we can very well explain it also.

But if you look into this particular equivalent apparently it would not look at it is going to be equivalent because it is having some complex combination. But in their equivalent you are not going to have a formal probe to see whether their equivalent or not for that we need some other information also, but inerrably I am going to establish the result or with example I am going to show that these what equivalent, okay.

Now you just look into the right hand side that we are having this particular things over here so this right hand side this expression can be written in some other way so what we are doing basically it is nothing but we are simply using that Demorgans theorem to this particular expression that expression that I have be now right hand side. We know that Demorgans theorem say negotiation of p or u is equivalent to your not of p and not of q.

So this is a Demorgans rule so we are using this particular Demorgans rule through this particular expression so what I am getting I am just pushing this particular negation inside so I am getting ˥E and this particular portion and this R becomes and negation of this second part okay, so just using the demorgan's theorem and we aare getting conjoint of two particular formulas. Now what is this ˥EG ˥Ø2.

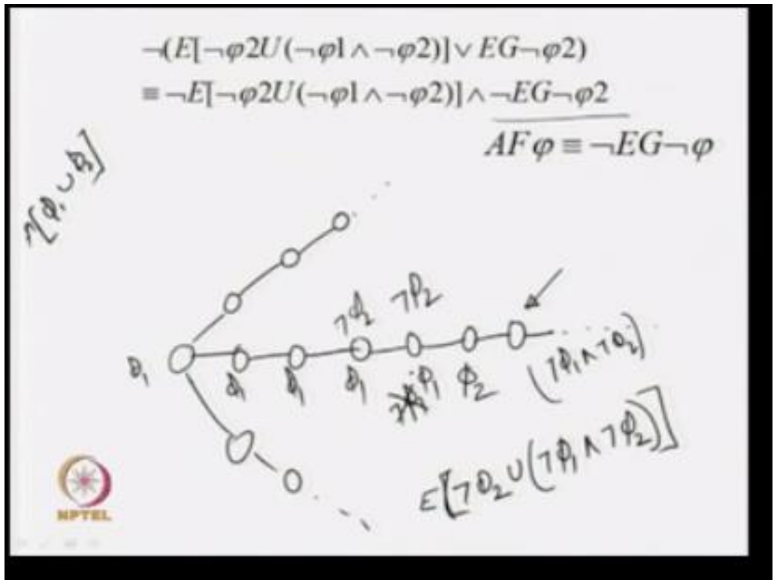
(Refer Slide Time: 27:29)



So already we have seen this are equivalence that AF Ø is equivalence to ⅂EG Ø2 that means we can say that we can replace this particular second part by AF Ø and first part is remain as it is so eventually we are getting this thing, now you just see that A Ø1 U Ø2 if I look into this particular things A Ø1 U Ø2 that means Ø1 should remains to until Ø2 becomes true in all part, or on the other what I can say that in all part in some future step Ø should must true Ø2 must hold and along with that it should satisfy that all other present steps Ø1 must true.

So that is the we must get some future step in all part where Ø2 is true this particular criteria are condition is kept sort by this particular part AF Ø2 okay and along with that now we are saying that in all preceding step Ø1 must be true so this particular other second condition is kept sort by this particular part, okay. Now we will see this things.

(Refer Slide Time: 28:49)



So basically if we are having this particular since this is the first part and this is the second part so F Ø so we can now look for all possible part so say that, I am having because this is A U so we may have many more excision part since A [Ø1 U Ø2] so in all part this must be true so we consider one particular part and same thing must be true in all other part so it says that in all part in future Ø must be hold to second part basically.

So I will say that in future I am going to have this particular Ø2 that means in this particular part In future I am going to the fact and along with that what it must say that it must true this particular portion ˥E ˥Ø2 until ˥Ø1 and ˥Ø2 so what basically it says that I am going to get the formula E˥Ø2 until ˥Ø1 and ˥Ø2, okay. So we are having this particular formula and it says that it must not be true in this particular part because I am having this particular negotiation, so we are going to see this particular program what it says that, Ø1 ˥Ø1 and ˥Ø2 to at some state and till that point you should have ˥Ø2, okay.

Now users at time getting this particular states say considering this particular state where ˥Ø1 and ˥Ø2 is true, so to become this particular formula 2 then what will happen on all those particular state ˥Ø2 must be true, okay. And if in all those particular state ˥Ø2 is true then we say that these particular part is true, okay and you should have the negation length so this is this should not be true over here.

That means it should not it should satisfy this particular formula over here okay, now since your ˥Ø2 is true over here so it is not true at that particular point so till this particular point ˥Ø2 we not remain true, so negation is true over here so but what will happen in this particular state, say if ˥Ø2 is true over here okay, so along with that say if ˥Ø1 is also true then what will happen ˥Ø1and ˥Ø2 will be true.

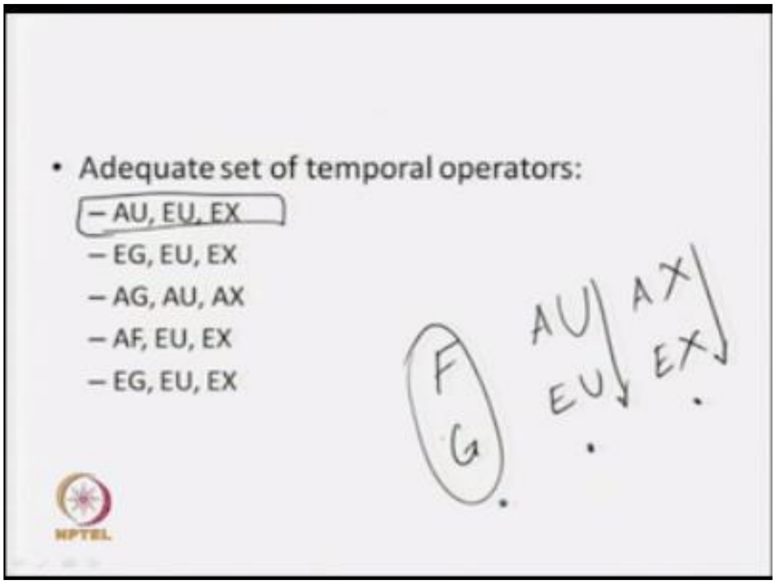
So in this particular case what I am going to tell so already I have said that in symmetric the fusion notion increase the presence also so since this is both of these are true that means eventually this formula will be true but our objective the negation must be true since negation must be true so if eventually what will happen that ˥Ø1 must not be true over here so Ø1 must be true at that point.

Because we need the negation of this particular formula must be true so with this same logic it will find that if ˥Ø2 is true over here then we will find that Ø1 is true over here also because we need this component must be maintained okay, so in this till this particular part so in this way we will find that in all case the Ø1 will be true so that means we are getting this particular part where Ø1 until Ø2 is true.

And with this similar logic you can extend it for all other part and we will find that on all other part Ø1 will remains to until Ø2 so what basically of we have seen that AØ1 until Ø2 can be expressed with the help of this particular formula, so which basically involve E and kilo quarter that means A until can be expressed with the help of v until or on the other hand you can say that E until can be expressed with the help of A until also that means we are having equivalence relation between A until and E until.

In all part anti operators and dual exist part until operator so now with this particular relations where you are having an equivalence between A until and E until so since we have this particular equivalence between these two operator then what will happen, now you can get some other educator operators.

(Refer Slide Time: 33:53)



Now what base thing so adequate set of operators temporal operators or in CTL operators we are going to get different sets so first one already we have discussed or already we have seen that AU, EU and EX okay now what did the other things use us today, if I am having until operator if I know that AU is there than it can be expressed as EU okay similarly if I am having the next state operator.

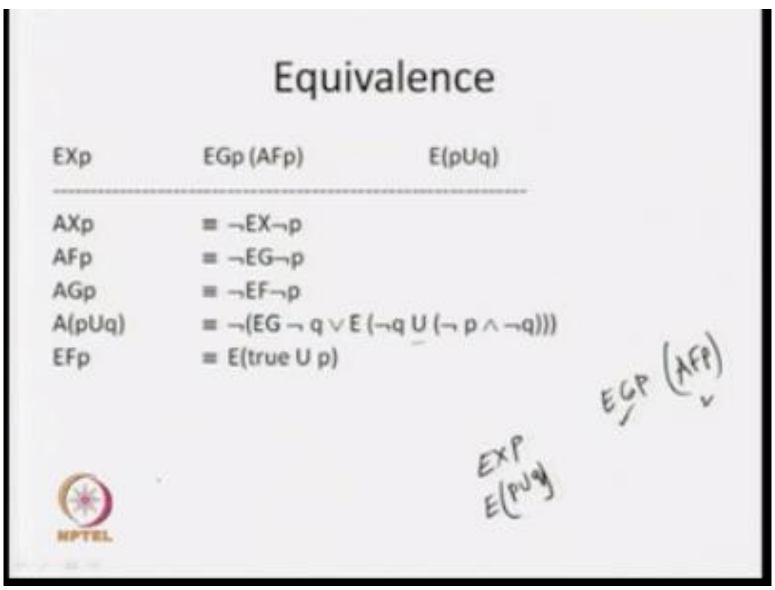
So this is you AX than EX can be expressed with the help of Ax, so we need anyone from AU or EU and we need anyone from EX and AX and other two so now out of that four temporal operator that we have discussed that until an x already it can be expressed either A or E now other two operators we are having F and G, in future and globally these are basically existent of quantifier of states along a path and USL quantifier of states along a path.

So one can be expressed in odder so we can take anyone of these two operators because Z can be expressed it as or F can be express it G, so we will take anyone of these so within one operators from these state picking in the operator for until and picking in the operator called as F and G we can now construct or you can get several educate state of operator, so that is why we are just listing some of them say these are treated as adequate set of operators.

So taking one X Ex taking one U EU and we are taking just one thing EG because if I am going to take EG then it will be expressed with your AF then we can express it with the help of your until or these things so basically these are the similarly here also EG the second I am taking AG or I am taking that in state A I am taking EU, EX and AF or EU, EX and EG like if you can construct if for an adequate state of CTL operators, okay. And we can see that we need atleast three operators.

With the help of three operators we can express the other five also, okay so we are going to get several educate set of CTL operators.

(Refer Slide Time: 36:12)



So out of this particular thing say I am going to consider say something and explain how we are going to see this, so either way I am going to take the educate set of operator say Ex –P I am taking at 1x I am going to take E(P U Q) and the third part either I am taking EGP okay there exist a part in future global that we holds or may be AFP because I know the E id dual of A, G is dual of P.

Either I can take this one or this, so in this particular case you just see that if I know the procedure how to evaluate Ex-P then I can very well evaluate A x –P by looking into this particular equivalent Ax P is equivalent to ˥E x ˥P we do not have any problem it is very simple now if AF P then what will happen, you just see that I am having say EGP then what I can say that AFP is nothing but ˥EG ˥P so if know the procedure for EGP then you can look for your distance AFP or on the other.

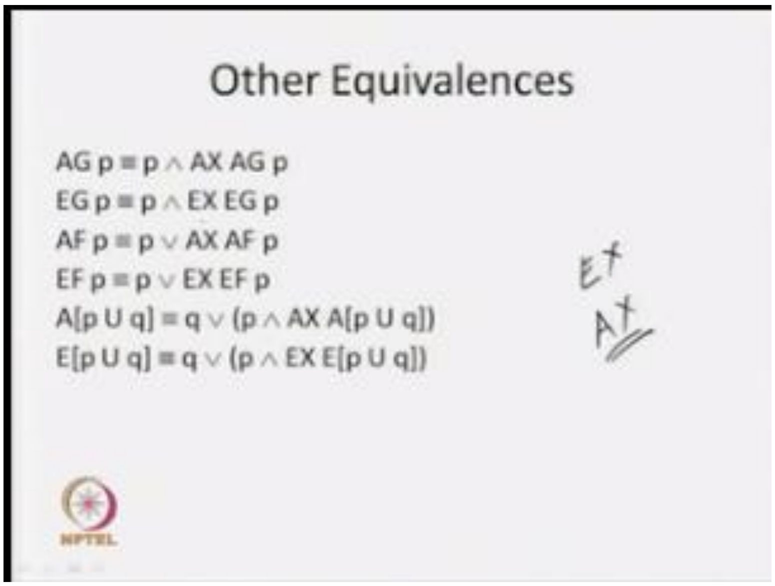
And if I am having this AFP then I can evaluate EGP okay so that is why I am saying that either of this I can take so I am having this equivalence AFP is equivalent to ˥EGP ˥P now what will happen in AGP, AGP is nothing but you can say that ˥EF ˥P that means you are having this particular future of pattern, so this is the future operator and this future operator can be expressed with the help of AUP, AP U Q is expressed with the help this particular equivalent that we have discussed in negation of this whole formula EG ˥q or E U okay and EF is, is at see that E true until P so we need this are EFP for AGP so we are going to say that E true until P, so if with the help of those particular equivalence we can say that if I am going to consider only these three operators then what will happen.

Others can be expressed and similarly if I take EX – P, EP U q and AFP can others three can be expressed with the help of this three operators, so we must know the procedure to evaluate at least three temporal CTL operators then other can be expressed with the help of this thing, so this is the power of equivalence formula and we have seen that now we need restricted said and we are going to walk with that restricted set.

So this is the notion of equivalent in CTL formula, so what is the notion it is slightly different from your probation calculus and predict calculus because here the meaning of CTL formula defines a one in model and that is why it said that two formulas will be equivalent provided in any step of any model if the truth values of one formula if true then the truth values of other formula must be true.

Then only we are going to say that these two formulas will be equivalent, okay. Now this is the notion equal like that we can find summary like will so but these are the basic equivalent formula.

(Refer Slide Time: 39:22)

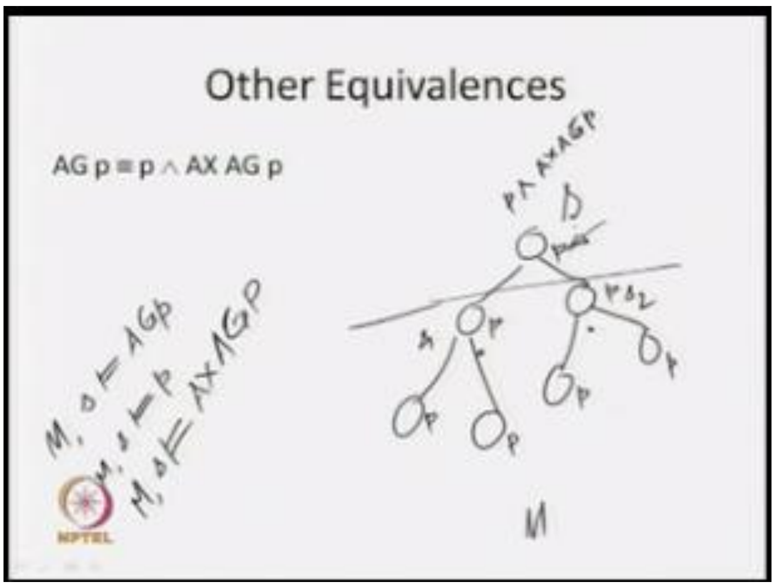


Now we are having some other equivalence also and we are saying that these are basically some sort of your recursive defiance and so what I am saying just see you first particular equivalence we are saying that AG p = p ^ AX AG p so in AG p = p ^ AX AG p similarly EG p = p ^ EX EG p, okay and AF P = P ∨ AX AF p, EP = p ∨ EX EF p and like until a p until p is equivalent the q or p n ax a until p and q ep until q is equivalent the q or p and e xep q.

And q see the next those particular equivalent actually this is some sort of you discursive definition of those particular temporal operator over here we having six temporal operator that means this six temporal operator is express by themselves along with the temporal operator ex and ax the next step operator.

Okay so the we note the meaning of ex and ax if it is having all part quantifier then this ax will come in to future if it is the quantifi4er is you existence quantifier either ex will come in this future so say acp I am having ax I am having ex so all this operators six operators are depend themselves so this is some sort of your discursive definition of those particular temporal operators and we are going to have those particular equivalent. So see those particular meaning how we are going to get it okay.

(Refer Slide Time: 41:17)



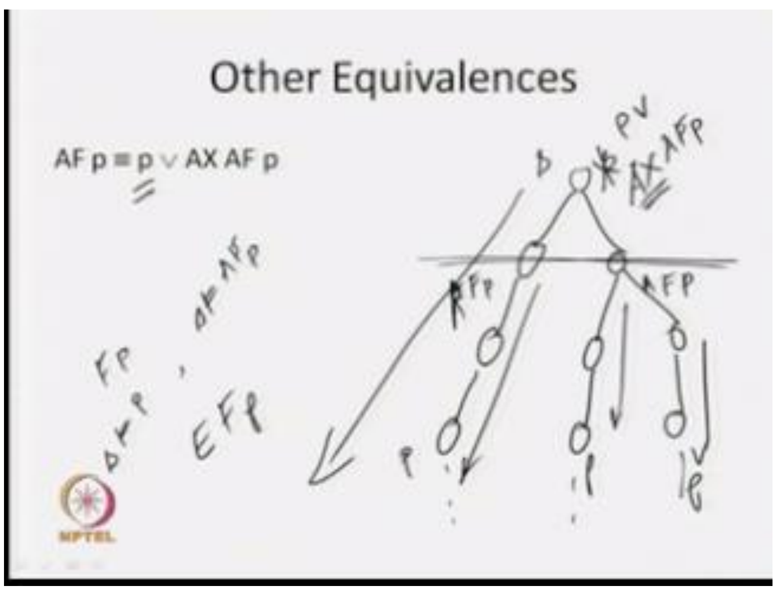
So first formula we have to saying that AGp now if I am going to have a model just explain with the help of model on the then it will be clear to you so what will hap[pen so this is your s I am going to say that whether in this particular model m in state s whether it happens AGP okay so that means in all part globally p must be true that means we can say that I am going to say that this AGp will true ion this particular step is p is true in all those step.

Now I can spread this AGp like that if I am going to look for this particular step that means in step m in step s of m it must model p if p is true is as then what will happen we are going to see all the next step over here, okay what are the two next step of that this is s1 and s2 in s1 and s2 now what it should happen again s1 and s2 AGp must be true so that why we are saying that in all what ion next step AGp must be true if AGp is true over here then o both the step that means her I am going to get that in all part in next step AGP is true.

Okay so that why we are saying that if p is true over here then I am going to look for those particular step whether AGp is true or not since I am going to look for all those step that means in this particular state I am going to say that in all part in next state AGp is true. So that means I am going to say that it will all over here if again state s of model f models AX AGp okay now you just see that if both these components are true over here p and ax Agp then what we can say that this formula AGp is true at that particular part.

So I think it is clear to you that we are getting this particular thing if both the components are true then I can say that agp is true over here.

(Refer Slide Time: 43:38)



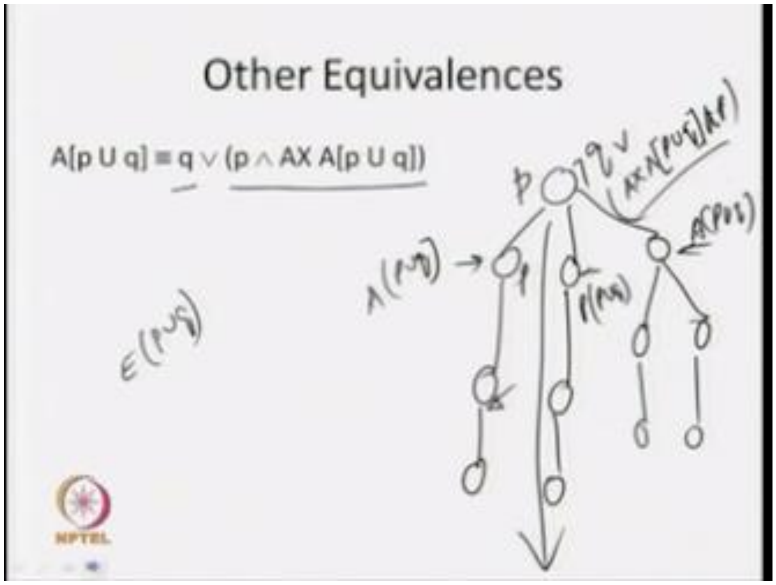
So with the similar notes on we can go for Agp is equivalent to p and ex egp so this is similar to your desort but instead of looking for all part it is your EGp so we are concern about a one particular part so what we are going to say p must be true over here and they are exist the part in next state AGp must be true so AGP must be true so this is the things that call EGp p and EX EGp okay.

Now similarly we are having AFP is written as like a p or ex AFp now again just see that similar things that we are having okay now what will hap[pen ion this particular case I am say that if p is true over here then I am getting this particular part that a p over here but already we have seen the symmetric that in this particular notion of time it says that future includes the present behavior also that means if p is true her itself say I am concern about this particular step if p is true in state that means if s model is p then we can say that s model f p also okay.

So if p models here I am going to said that fp model said f that means that particular design this is the design of two formula if p is true over here and I am going to say that afp is true here itself but if say p is not true over here then what will happen then I am going to say that in all part in next step whether fp holds or not so that why I am saying that it is p or this things if p is holds there itself then I will sy that p is true over here so if p is not true then I am going to look for this p[reticular component in all part in next step whether fp holds or not.

Whether in next step ap holds or not when ap will involve I am going to get some fuses that where p is true so this is the way that we are going to say that a is express with the help of f1but along with that particular ax of part of it so similar notes on will go for fp also in thin particular EFP what will happen instead of a we are going to have e so instead of looking for all parts we are going to look for one particular baton okay. So this is way that we are expressing so this is another equivalent with respect to your future operator.

(Refer Slide Time: 46:24)



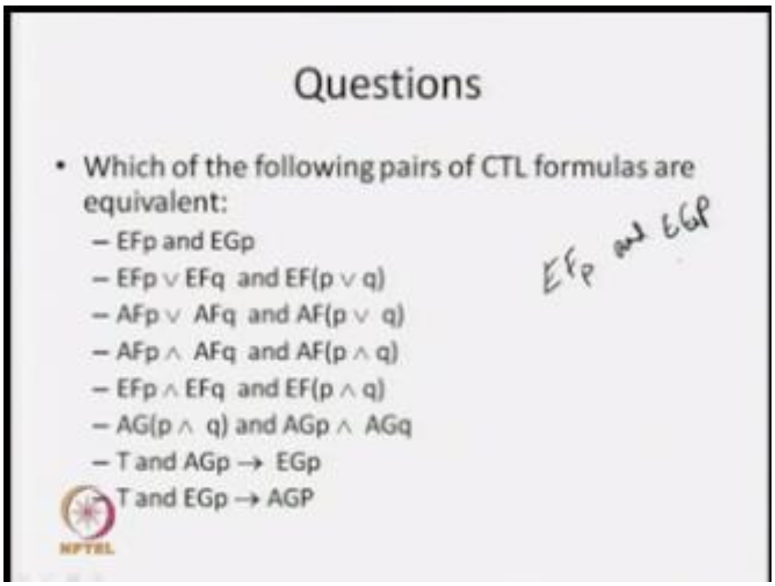
Now look for the other one that antelo part a p and p and q again we are saying that a p and p ᴗ is = q or pn x p until pq even you just see that particular so either what happen the notion of time it to look in to it we have seen that when we are looking to a future behavior we say that present scenario is also included in the future behavior so future includes the present so that is why we are going to look for ap and till give p remains to until q becomes true.

So this is having again 2 decedent okay either first part or second part okay so if first part I true we will say that ap until w is true that means if q is true here itself then will say that ap until q is true then we do not look for it. So if q is not true at this particular point so if not of q when we are going to loom for the second component ax a p Λ uq and along that n p so what does it means that means if it is q is to dear itself and say that ap and until q is true but if it is not true then we have look for a second part and what it must do that p must be true over here p and this is the p and ax all next step what it must hold it must hold again ap until q must be true over here.

Okay so this is the way that we are going to look in it now when I am going to look for the whether ap and until q is in this particular step again I look for this two condition either first I gust two dear itself or not if q is not true then I will see whether p is true over here if p is true then I am going to see whether in next step ap until q is true it. So these are another equivalent that ap and until q is express with the help of ap until q but along with your ax over.

So similarly you are going to get ep until q in this particular case instead of looking all possible parts will be concern about any part if this things happens in one particular part will say that ep until q is true in this particular step.

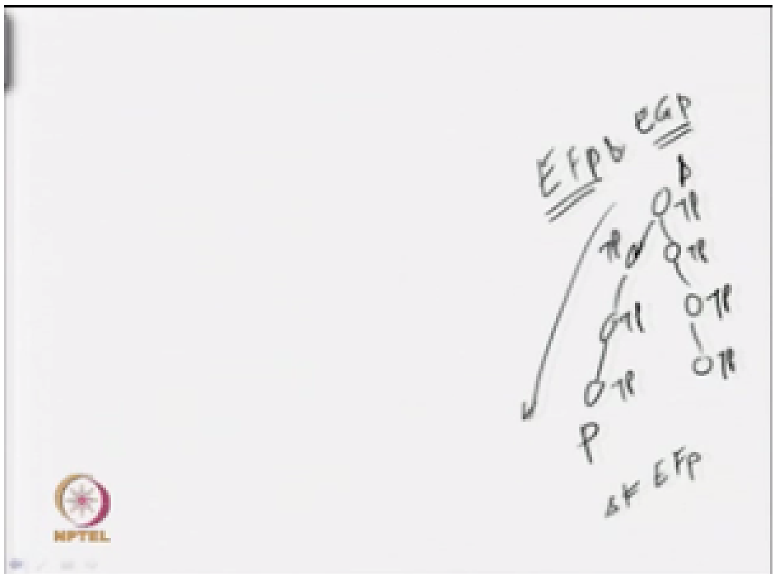
(Refer Slide Time: 48:53)



Okay now after going true this particular equivalents now just look for some questions basically we are going to check whether some of the some pairs of your CTL formula as are equivalents or not so the question I have given over like that which of the following pairs of CTL formulas are equivalent okay, here I am giving some pairs of pas I am saying that EFp and EGp second one I am saying that Efp or EFq and Ef and P or q third one is your EFp or EFu and Ef nd P or q so like that I am going to check whether those particular pairs are equivalent or not.

So first one you just see that I am saying that Efp and EGp so while I am going to check for equivalent if they are going to equivalent in this then what will say that it will try to establish that logical okay but if they are not equivalent in that particular case what will happen will try to give a counter model will give an counter example whether I will say that in on particular step one is true but other one is false and in that particular case we will say that this two are not equivalent. Because I am saying that what is the notion of equivalent in any step in any model if one is true then other must be true now is it I am having this particular formula Efp and EGp okay.

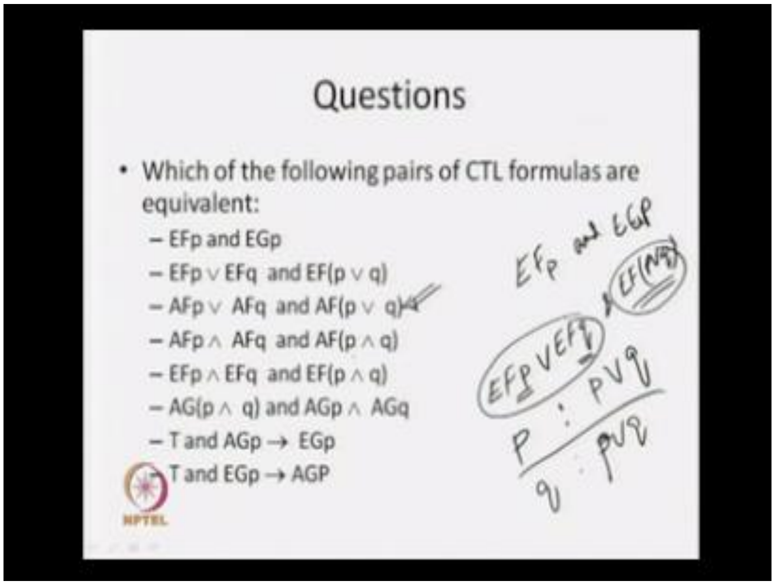
(Refer Slide Time: 50:27)



So EFp and EGp so in that particular case I am going to consider one model I will say that p is true over here so in this particular case in this particular state s I will get that s models EFp and say all are that say not of p is true so in this particular model in s I am getting Ef it exist the part in future p holds okay but EGp I am not getting any part where in all step not of p will be true so at least I am getting coming up with one example where in this particular step one formula is true but the other one is false.

So in this case I am not going to say that this two are equivalent so this two formulas are not equivalent.

(Refer Slide Time: 51:33)



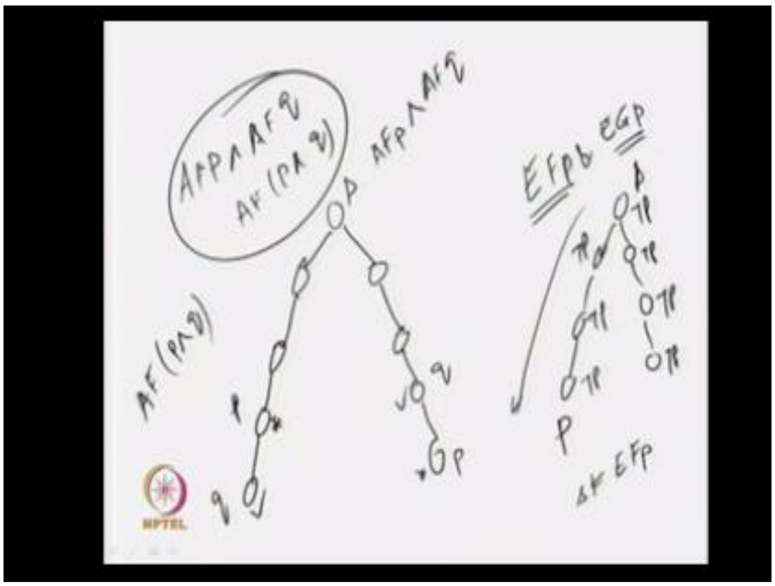
Now look for the second formula we are talking about EFp or EFq and EFp or q now in this particular case what I am saying that we are exist the part in future p holds r we are exist the part in future q holds okay and what are this equivalent they exist the part in future p or q holds or not so we know that if p is true then what will happen always p or q is true because either order p is true then p or q is always true if p is false then it depends on q okay.

So if both or false it is going to give me false if anyone is true then it is going to give me two fellow so if p is true I am going to say that p or q is true that means if they are exist the part in future if p is true necessarily I am going to get the they are exist the part in future p or q is true similarly if they are exist part in future q is true then I am going to get a part where p or q is true because if q is true then p or q also true but if they does not exist any part when p is false and q is false then p or q also false you just see that if this particular part is true in any step then this part is also true if this is false the this one is also false.

Sop in that particular case what I can say that these two formulas are equivalent now look into the third part. It is saying that AFp or AFq and AFp or q. So this also if you look into this thing it is similar to the second formula instead of A I am E. So this is the same properties holds for this particular OR operator so these two formulas are also equivalent you can check it okay.

You can check it I am saying that this third formula is also equivalent because this is the neutral particular of OR if P is true if P or Q is true then Q or P is also true. Now look for this particular forth operator forth equation AFp and AFq and AFp and q.

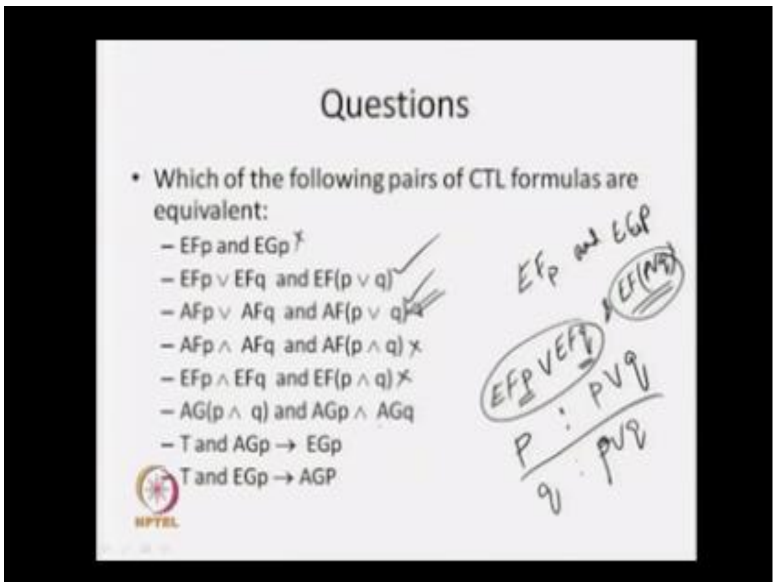
(Refer Slide Time: 53:59)



So we are going to set up FP and FQ now AFp and this is the formula I am looking to it AFp and AFq. AFq and the second formula I am doing that AFp and Q now you just see that loop for any model. But at this the two things are true or not I am going to check the set up P is true over here Q is true over here. So P is true over here Q is true over here and what will happen in all part if I am looking for this particular things I am going to set up in all part in feature P is true. If I go along this particular path getting this particular step is true if I go along this particular path I am getting this particular thing.

And AFq if I go along this particular path this Q is over here and if I go to other part then Q is also along all path AFq is true. But now if you look into this particular formula AFp and q. then along this two path I am not getting anything step were P and Q is true because in this particular step either P is true or Q is true. So if P and Q is not true since AFp and q are not true in any of the future step so AFp and Q are not true. That means these two are given over here are not equivalent okay. You understand these things so you have to see because the equivalence is in any step in any model.

(Refer Slide Time: 55:59)

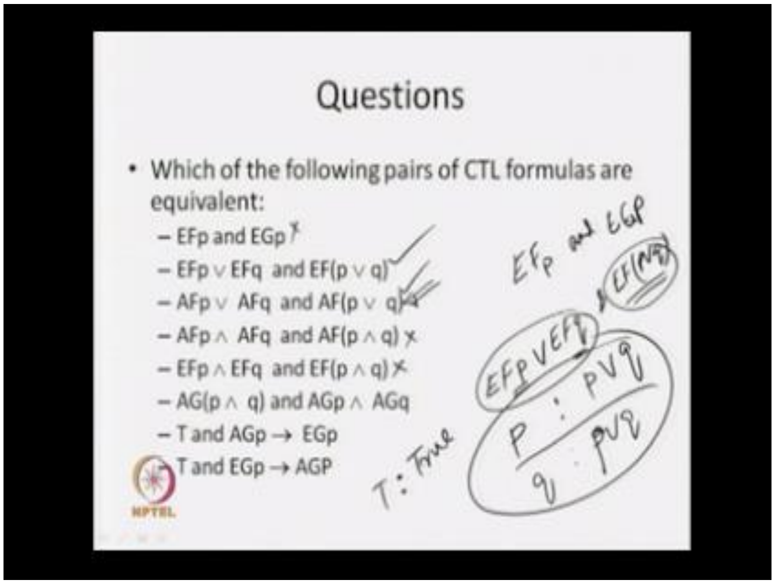


If the two formulas one is true then other must be true this is the non solving equivalent okay. Now so we are saying that this is not equivalent this is not equivalent first one, second one, and third one is equivalent. Now similarly I can EFp and EFq and Efp and n so this is similar to the previous formula instead of your all part now I am having the existing part now in previous part previous example I have shown two different parts now you consider anyone of this particular part.

You will find that Efp and EFq may be true in a step but Efp and Q will not be true. So this formula is also not equivalent okay. Now the next formula you see that AGp and q and AGp and AGq. Now we try it yourself and see whether these two are equivalent or not this is similar way if you are going to set this equivalent then logically you have to establish it you can use any properties now we have used a property over here and then P is true or Q must be true. Q is true then P or Q must be true like that you look for some of that particular part and try to establish if they are equivalent.

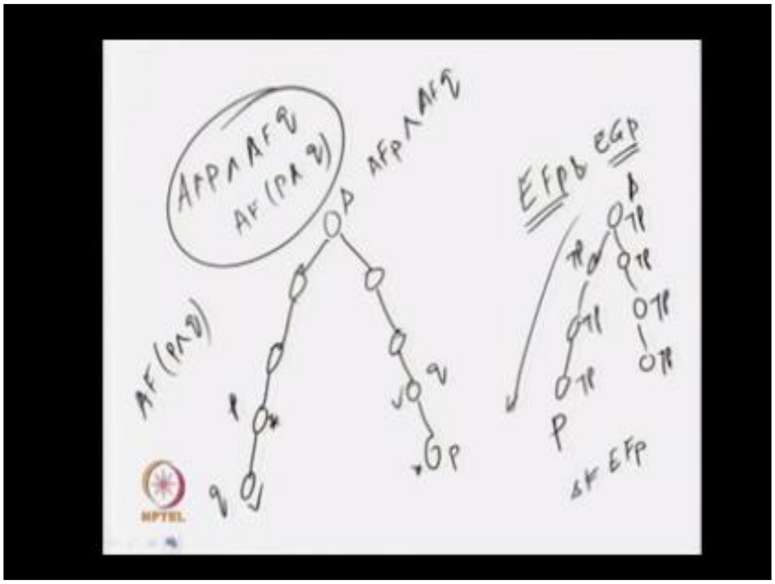
And if you feel that they are not equivalent then try to give a contra example try to come with a model were you show that one formula is true in a particular formula but the other formula is not true. So if you are going to establish that or going to show that these to formulas are not equivalent then you should give me a contra example Or if they are equivalent then what you have to do you should logically establish it then next formula I am saying that

(Refer Slide Time: 57:40)



EN AGp implies AGp and pn AGp implies what is this, this is nothing but the top I have assumed that this is the true value true okay.

(Refer Slide Time: 57:53)

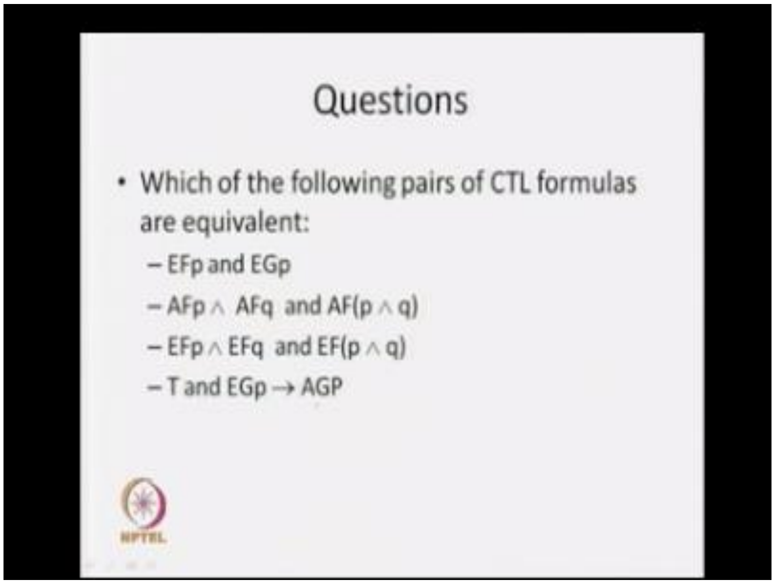


So I will show one thing T and the second formula I am using that AGp implies EGp that is true is always true. I have to check whether the this is a proplogy or not okay. now if this is your propology then the true values will always be always true and in this particular case all these are equivalent now what I will in reline this is the propology because what I am saying that if AGp so if I am going to say that, that means P is true over here in all step then I can say in this particular step.

AGp is true now what is particular application P implies Q if P is true, then Q is true that we can say all part no P holds. In step S in all global a P holds so instead I can say that models or all part globally P holds because I am just going to forsake of completion I am going to give a self prove over here and you see that all step is true so I can say that instead AGp is true. So now what the second formula say there exist part no where P holds. If P proves then second part second program must be true.

Here I am doing several parts now you can think of one particular part may be this particular part because we know that all part no where P is true the exits part were P is no were true okay. We are going to any part where it is true so if this part is true then the second part is also true. So this is going to give me the will prove so that is why I am going to9 say that P true and this particular formula because this is some sort of propology we get because if this is true this will always be true and true and true is going to give you always true okay.

(Refer Slide Time: 1:00:17)



So that is why I am saying that the formula and this particular true or equivalent so similarly you can look for the other one also which I am going to talk about P and EGp implies AGp okay we see that they are equivalent or not if they are not equivalent then you can give a contra example if they are equivalent logically that be stop there okay. We this I will stop here today again we will be meeting in our next class thank you.

Centre for Educational Technology
IIT Guwahati
Production

Head CET
Prof. Sunil Khijwania
CET Production Team

Bikash Jyoti Nath

CS Bhaskar Bora

Dibyajyothi Lahkar

Kallal Barua

Kaushik Kr.Sarma

Queen Barman

Rekha Hazarika

CET Administrative Team

Susanta Sarma

Swapan Debnath