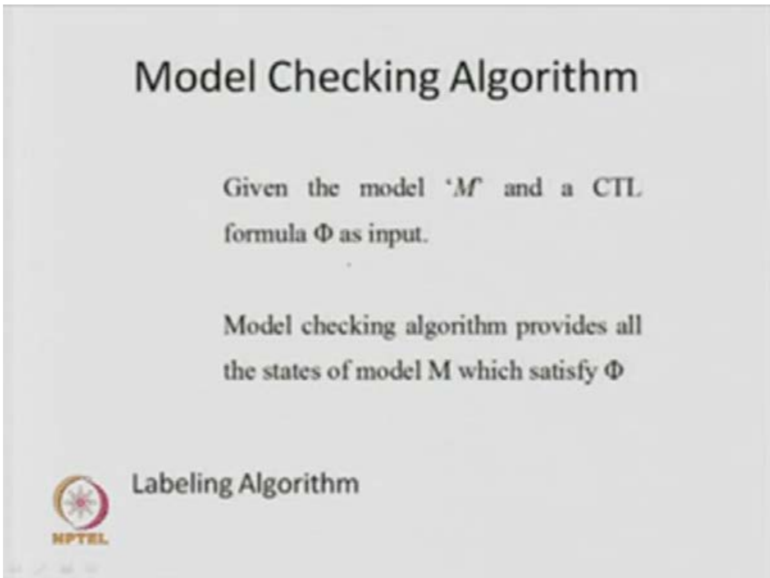
Design Verification and Test of Digital VLSI Designs
Prof. Dr. Santosh Biswas
Prof. Dr. Jatindra Kumar Deka
Indian Institute of Technology, Guwahati

Module - 5
Verification Techniques
Lecture - 3
Model Checking Algorithms

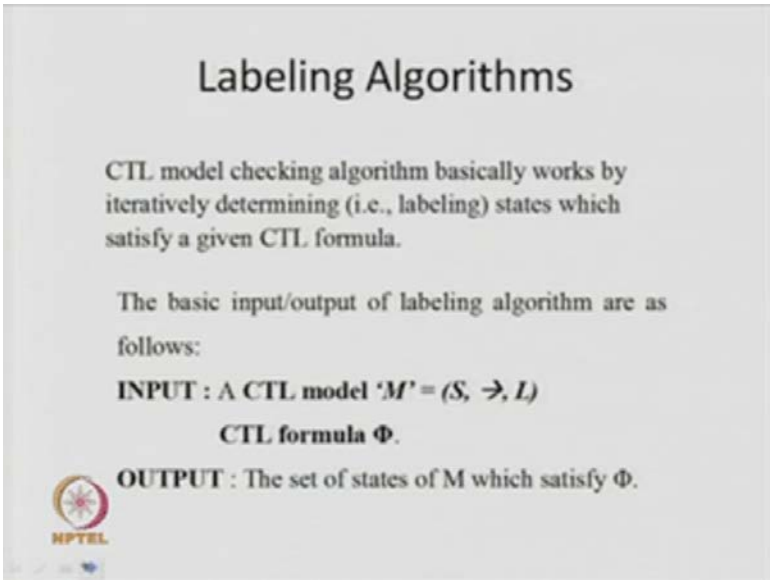
So, in last class, we have discussed about the model checking algorithm. And what is this model checking algorithm? Basically, we have seen its some sort of labeling algorithm, we are going to take the kripke structure, we will take the CTL formula. And after that we will run our labeling algorithm to level each and every step would the formula, which are true in this particular step. So, basically that is we are saying that.

(Refer Slide Time: 00:46)



We are going to give a model M and a CTL formula phi. So, model checking algorithm provides all the states of model M which satisfy phi. So, it is some sort of your labeling algorithm.

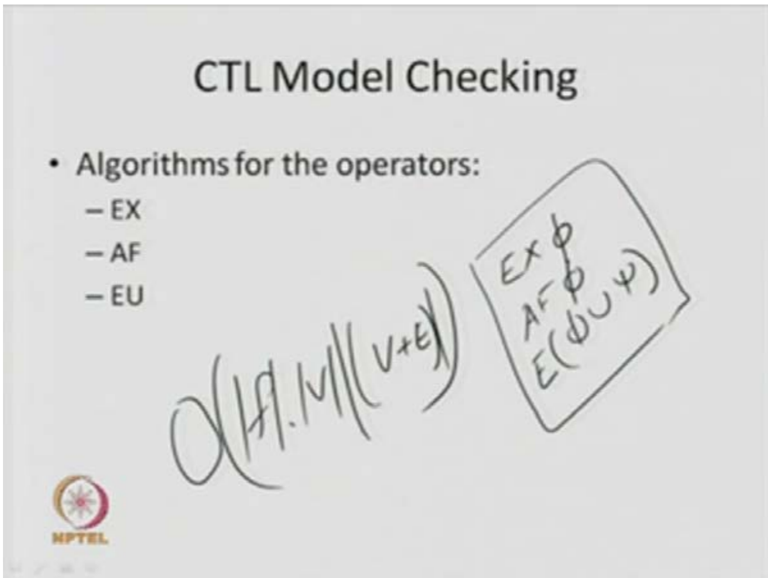
(Refer Slide Time: 00:57)



So, what basically we have, in this particular labeling algorithm, the input to this particular algorithm is your model M, which is having set of step as the transition relation and the labeling function. So, already we have discussed all those issues in our kripke structure. Along with that we are giving you formula phi then we will check whether this formula phi is to in this model, and basically it is going to give me the state of steps of M which satisfies phi. So, it is going to determine a state of step, where it is true. So, in last class we have seen 3 algorithms basically.

For EX, AF and EU, EX is basically EX phi; that means, where DR exist a part in next step phi holds AF phi in all phi in fuse of phi holes and E phi until psi here exist a part such that phi remains true until psi becomes true. And we have seen that these 3 are the adequate set of temper operators by which we can derive the other CTL operators, because we are having both temporal operators along with that 2. But quantify all together we are going to get at different combinations, but this is the adequate set of operators. So, if you know the procedure for these operators then we can look for the 2 type of any other CTL operators.

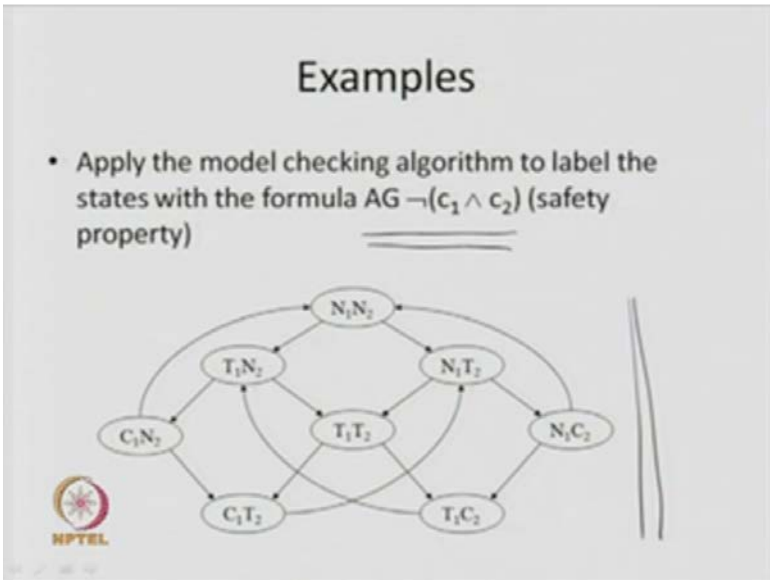
(Refer Slide Time: 01:37)



Now, in last class we have seen the algorithm and these are, these algorithm some sort of your graph traversal algorithm. So, we are going to take the kripke structure with your set of steps all the transition and along with the labeling function. Now, in all those procedure what happens we are going to traverse the entire steps phase and their graph and going to check whether the properties true in a particular state of model. So, in complexity issue we have seen that this depends on the length of the formula. So, what is the length of the formula then it is because in each and every state we have to look for it. This is your number of notes in this particular graph and along with that V plus E. So, this is basically the total state because we have traverse inverse situation its transition.

So, the complexity it is stand up to the O, order of your length of the formula and the number of notes and your total states steps. So, it is linear in the length of the formula and coordinated in the number of notes that we have in this process. And we have seen that we have got an well defined algorithm for this operator. So, that means, we can say that we are having a automated of CTL model checking and this is the beauty of CTL model checking. So, we are getting well define algorithm and it transform linear term. Now, we will see the huge of this algorithm with the help of example. So, here to in today's lecture, I am going to discuss some examples only and if time permits then we will see some also.

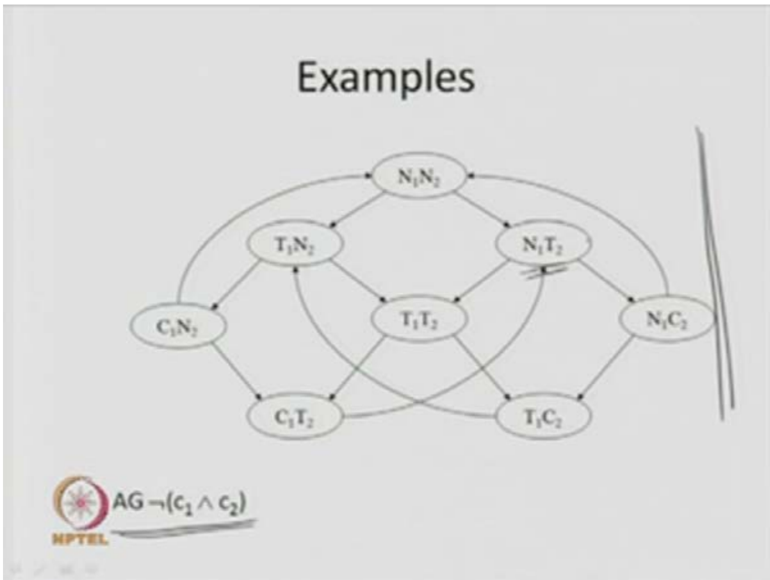
(Refer Slide Time: 04:07)



Now, those are in one of our class, we have discussed about the modeling of our mutual exclusion protocol. And going to check for the properties that need to be satisfied that particular model, we have considered model of 2 process p 1 and p 2. And we have seen the state steps that it is span up to be something like that that we have presented already we have discussed this particular example in details. So, these are the reachable steps that we have in our examples.

Now, we are going to check whether the safety properties satisfied or not in this particular model. So, what is the safety properties we are going to say that in all part globally not of C 1 and C 2; that means, at any point of prime more the process would not enter into the critical section. So, both of C 1 and C 2 need not be true simultaneously. So, this is C 1 is basically says that process p 1 is in your critical section. And C 2 says that process p 2 is in your critical section. Now, my properties safety property in CTL is AG not of C 1 and C 2.

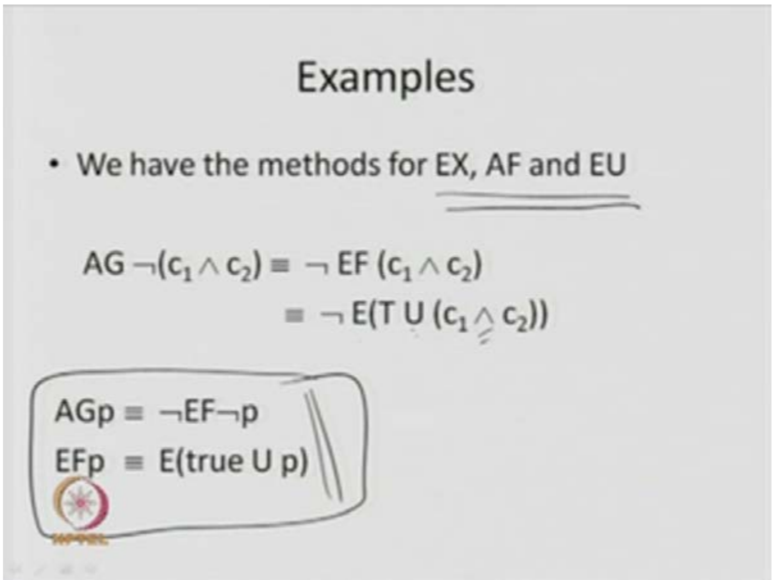
(Refer Slide Time: 05:18)



So, basically this is the thing that now, we having in our hand this is the model or kripke structure that we have. And this is the formula that we are going to check, whether what are in which are a step this formula is true. So, we have seen it is a kripke structure; we are having the labeling of those particular automatic proposition in each and every step. Like a simple example, I can say that N 1 T 1, it means that the process p 1 is in your non-critical section and process p 2 is trying into the trying to enter into the critical section.

So, that is why, I am saying that the automatic proposition N 1 and T 2 are 2 in this particular step and all other steps are these and all other proposition like, N 2 T 1 C 1 and C 2 are false in this particular step. So, if you say this is the very small system and by inspecting itself. We can say that in none of the step that not of C 1 and C 2 is true. But still, if we are going for a regard system by inspecting, we cannot do it we have to apply over and bottom. So, for the small example, we will apply over and bottom, now let see what will happen.

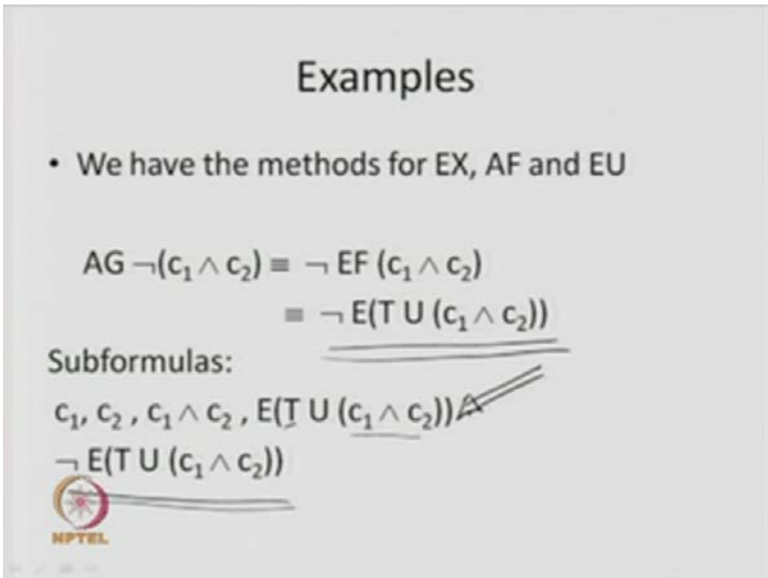
(Refer Slide Time: 06:25)



Now, we have seen that we are going to look into the adequate state of operator and we say that these are the 3 operators, that we will need EX AF and EU. And we have already discussed the algorithm for these 3 operators. So, whatever CTL formula we are giving somehow we have to convert into these 3 operators. So, already we have seen the equivalence about your CTL formulas and we know that AG can be express with the alpha BL.

So, AG not of C 1 and C 2 is equivalent to not of EF C 1 and C 2 and this can be again because we have EF, but we are having procedure called AF, but we are having procedure called EU. So, this EF can be written as E true until C 1 and C 2, because these are the 2 equivalence that we have in our hands. So, I have use these 2 equivalence AG p is equal to not of EF not of p and EF p is equal to E 2 until P. So, I have use these 2 equivalence and with the help of these equivalence, I have convert a this given CTL formula. And now user see that we are having this particular conjunct we have this E until and we have this equation. So, we have the procedure for this E until, so we are going to use this particular e until to check whether this property holds in some steps or not.

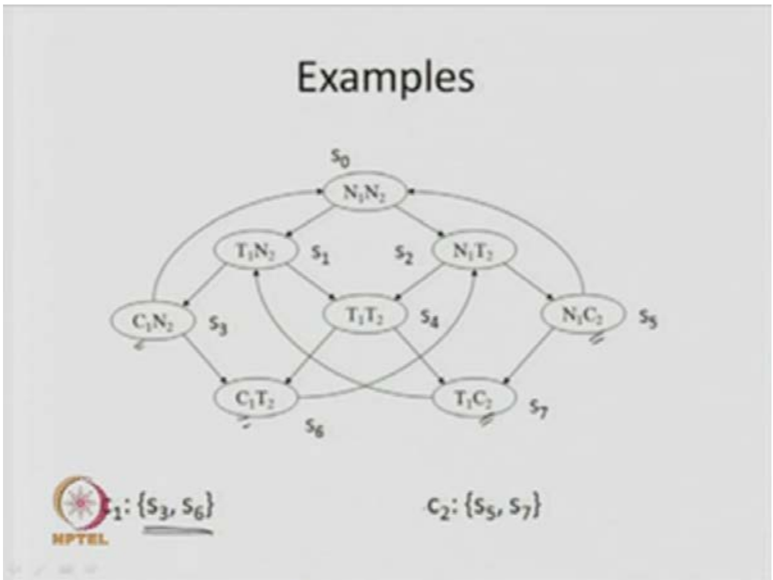
(Refer Slide Time: 07:52)



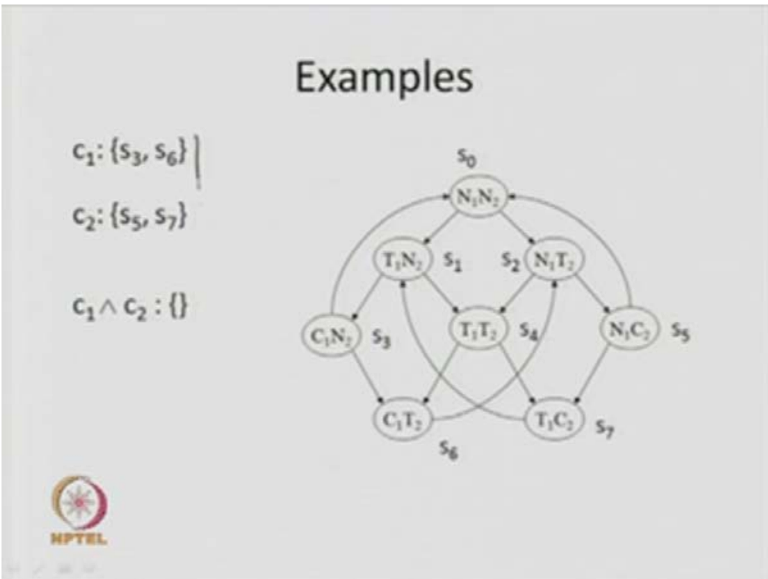
Now, by looking into this particular final formula, what are the sub formulas that we are having C 1 and C 2 are 2 automatic proposition. So, they are CTL formulas, since C 1 and C 2 are formulas C 1 and C 2 is also CTL formula. Now, E true until C 1 and C 2, since C 1 and C 2 is a CTL formula, T is also CTL formula. So, E true until C 1, C 2 is also CTL formula1, since E true until C 1, C 2 is a CTL formula. So, negation of this formula is also CTL formula. So, this is the formula that we are going to look into it, but to check for this particular formula, we have to look for this particular sub formula. Once we have the labeling of this particular, sub formulas than only we can talk about this particular given formula. So, we look for the automatic once C 1 and C 2 then C 1 and C 2 can this particular E true until C 1 and C 2.

So, now say we are having the labeling function we have label each and every step with the help of a automatic proposition. So, with the from this particular labeling function, we will get the, the C 1 that automatic proposition C 1 is true in your state S 3 and S 6. So, in S 3 C 1 is true and S 6 C 1 is true. Similarly for C 2, it is true in S 5 and S 7. So, this is the C 2, it is S 5 is label with C 2 and O S 7 is label with C 2. So, the state of steps where C 2 is true is in your S 5 and S 7. So, we know the labeling of these 2 sub formula C 1 and C 2, once we know the labeling of this C 1 and C 2, then we can look for the.

(Refer Slide Time: 08:49)

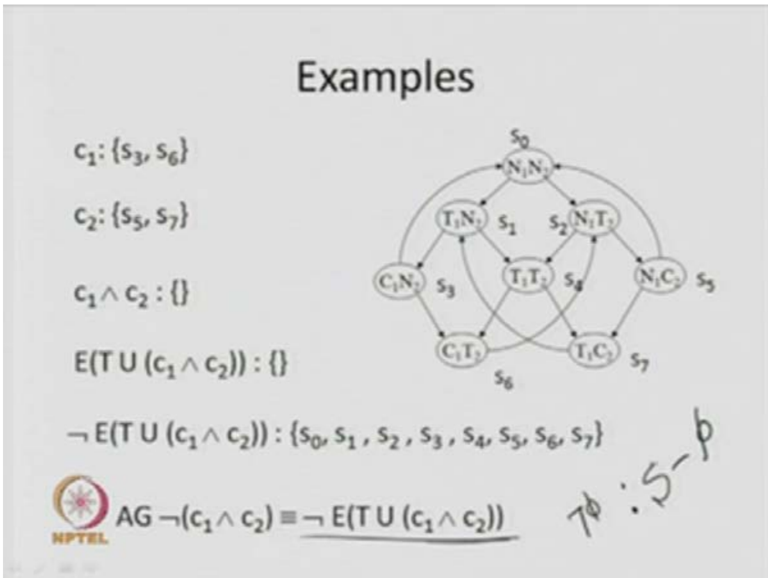


(Refer Slide Time: 09:35)



Labeling of C 1 and C 2 and what is the steps where C 1 and C 2 is true. This is nothing, but the intersection of these 2 particular steps. If you look into the intersection of these 2 particular step, then we are going to get a large. So, in none of the steps that C 1 and C 2 is true.

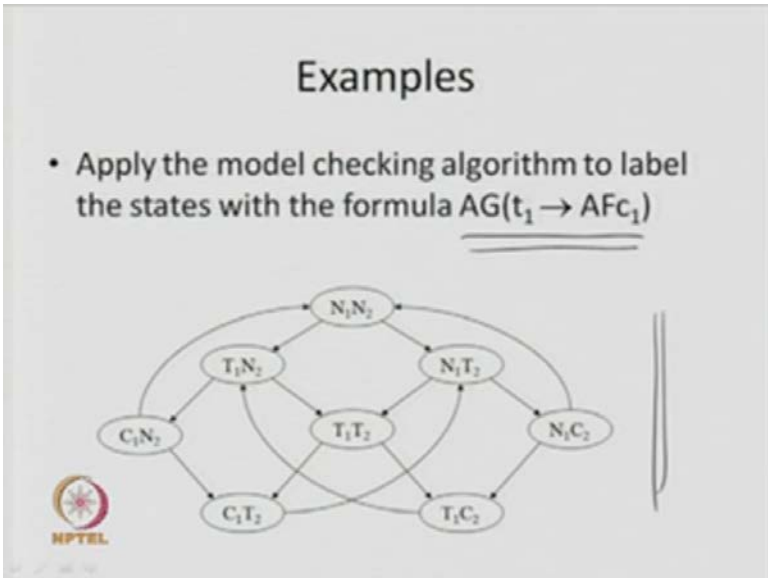
(Refer Slide Time: 09:55)



Now, what is our next formula this is your E true, until C 1 and C 2. In this case all, now it is very simple we are going to get null set because C 1 and C 2 is null. So, in the first case, we are not getting any state, so if we are getting some step then what will happen we are going to traverse the graph, since the initial state itself null. So, there is no point of traversing the graph.

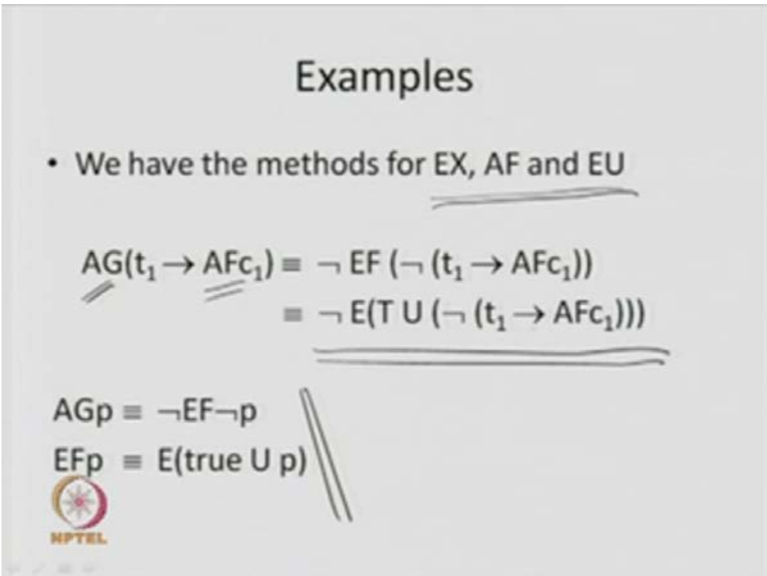
So, eventually we are going to get a ET true until C 1 and C 2 is also null set. Now, when ET 1, until T 2 is a null set then what is the negation of this things; negation of these things is nothing, but complete state phase minus where the formula phi is true. So, this is basically this sets give me the where the formula not of phi is true. So, my total state phase is given over here, so my this particular the formula where this particular the steps where this particular formula is true is given as your empty set or null set, so for negation of this formula will be true in all the states. So, that means what we are saying that this is the formula AG not of C 1 and C 2 which is equivalent to not of E true until C 1 and C 2. So, we have seen that in all the step this not of E until C 1 is true; that means, you can say that in all states all part globally not of C 1 and C 2 is true. So, from here we can compute the safety property has been satisfied by my model. So, this is the way that we can check this is a very simple example, now we will see one another example.

(Refer Slide Time: 11:38)



In second example, we are saying that again we are going to apply the model checking algorithm to label the state with the formula all part globally T 1 implies we are in all part in C 1. So, this is the property that we call as your liveness property. So, what we are saying that if any process wants to enter into the critical section that process eventually must get the chance to enter into the critical section this is the liveness property.

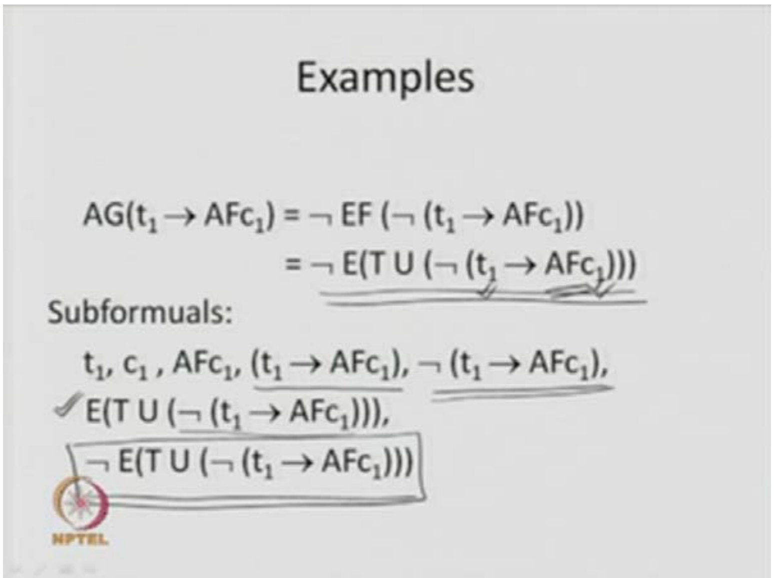
(Refer Slide Time: 12:16)



Now, we are going to check whether this particular liveness property is true in my model or not. Now, again you see that we are having method for EX AF and EU. So, somehow we have to express this particular formula in terms of these 3 form also. Now, AF we are having, so we have a process or procedure to check for this particular formula AF c 1, but we do not have method for particular AG. So, that is why we need to use some equivalence, so that we are going to have a method of process from this particular step. Now again similarly, say AG p is equivalent to not of EF not of p this is equivalent and the second equivalence is e f p is equal to e true until p.

So, we are again going to use these 2 equivalence, so eventually this given formula AG p 1 implies AF c 1 is stand up to be this particular formula not of t are exist true until not of t 1 implies AF c 1. So, this is the formula, now we have process for AF c 1 we have process for E until. So, these are the E until and this is the AF, so we can use our model checking algorithm to check this particular formula.

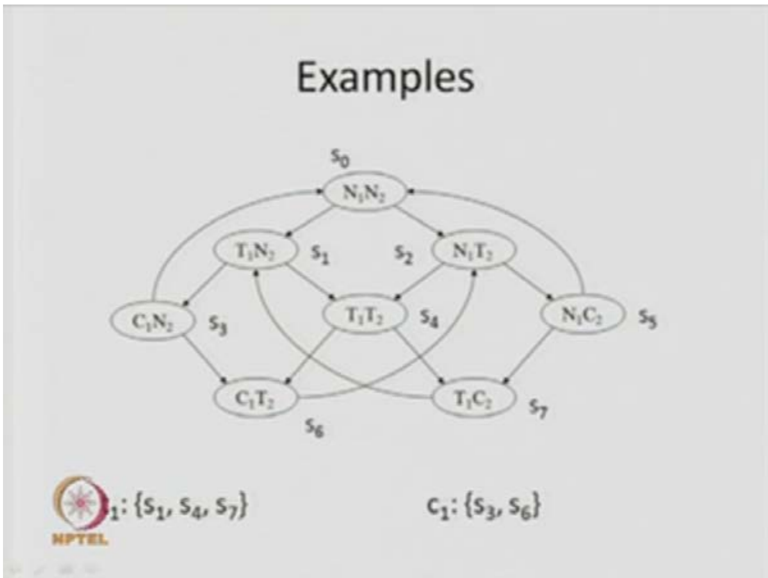
(Refer Slide Time: 13:34)



So, now when we are converting this particular given formula AG t 1 implies AF C 1 and this is the equivalent formula that we are getting. And now we are concerned about this particular formula. Now, what are the sub formulas? In this particular formula, now I can this thing, so t 1 is a sub formula, c 1 is a sub formula, since c 1 is a sub formula then AF c 1 is also sub formula, since t 1 and AF c 1 is also sub formula. So, t 1 implies AF c 1 will also be a sub formula or CTL sub formula. Since, this one is CTL sub formula negation of t is given CTL formula is again is CTL formula. So, we are going to get one sub formula, so from that now, since both the components true is always a CTL formula true until we are having this as a CTL formula.

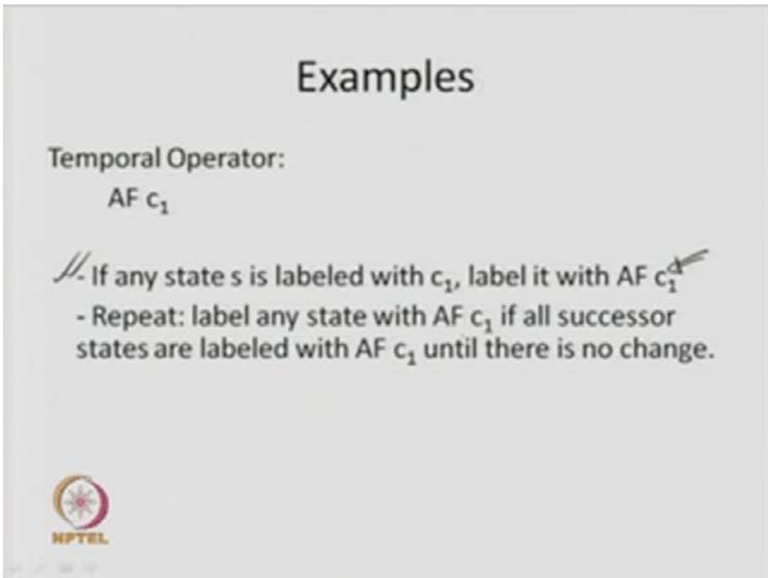
So, we are going to consider these as another sub formula and this is the final formula negation of this particular step. So, when we are going to look for a true values of these particular given formula, we must know the true values of all those particular sub formulas. We know the state of steps where the sub formulas are true or the states where these are satisfied. Now, we are going to see these things, now the sub formula we have t 1 and c 1 which are your basically automatic proposition.

(Refer Slide Time: 14:56)



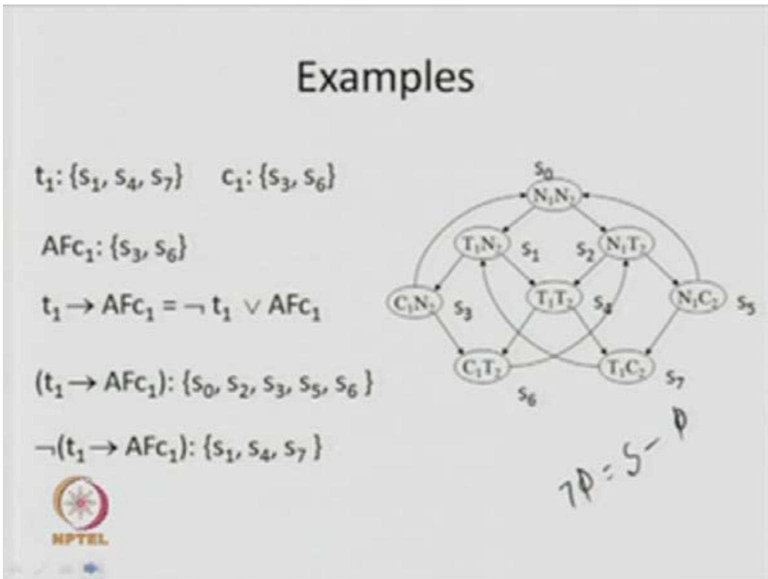
So, by looking into this particular state phase we will say that t 1 is true in the set S 1, S 4 and S 7. So, this is the step S 1, S 4 and S 7, in this 3 particular state the automatic proposition p 1 is true. Similarly, the state of step where C 2 is true this is your S 3 C 1 is true in S 3 and S 6. So, this is S 3 and S 6 in this 2 steps the automatic proposition C 1 is true and in another step C 1 and C 2 is true. So, we are getting these 2 steps where formula T 1 and C 1 is true. Now, next sub formula is your AF c 1, we know the state of steps where T 1 and C 1 is true, now we will look for AF c 1.

(Refer Slide Time: 15:44)



So, what is the procedure for AF c 1, so for any temporal operator for this procedure we having 2 step, first step is says that if any state S is labeled with C 1 and label it with AF c 1; that means, where the C 1 is true AF c 1 is also true. Because it is coming from our semantics above a fusers, because fuser inputs the present scenario after that we are going to repeat this process label any state with AF c 1, if all successor states are labeled with AF c 1 until there is no charge. So, we are going to repeat this case. Now, once wherever C 1 is true, we are going to set AF c 1 is true and will repeat. Now, since C 1 is true you know S 3 and S 6. So, AF c 1 is also true S 3 plus S 6.

(Refer Slide Time: 16:23)



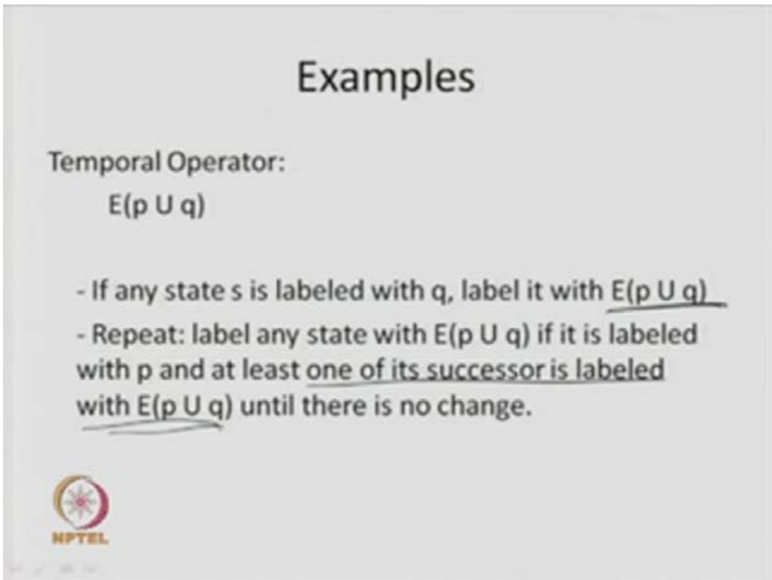
Now, this is the base case that we are having now, we will going to enter into the loop. Now, from this 2 steps we are going to traverse the graph in backward direction, we will see what is the predecessor of these particular steps. Now, since the predecessor is your S 1, now we will see where the it is having 2 next state S here and S 4. Now, S 3 is labeled with your AF c 1 in your base case, but you are S 4 is not labeled with C 1. So, S 1 both the successor are not labeled with your AF c 1. So, that one S 1 will not include in this particular states.

Since, S 1 is not include over there, so there is no chance of this particular say. So, we will terminate double; that means, AF c 1 is true in this 2 step S 3 and S 6. Now, the next formula we are getting t 1implies AF c 1. Now, when you are coming to this particular sub formula, we know the labeling of these 3 sub formula p 1 c 1 and AF c 1. Now, I can in this particular say that t 1 implies AF c 1 is equivalent to not of t 1 or AF c 1. So, this is simple your logical connective, so in abrasive that we can check it we need not to go for loop.

So, now we are going to check where t 1 is false, we know that t 1 is true in your S 1, S 4 and S 7. So, not of t 1 will we true in the remaining steps and C 1 is 3 in your S 3 and S 6. Since, it is true in S 3 and S 6, beside these 2 steps are going to come. So, t 1 employs AF c 1 will be true in your S 0, S 2, S 3, because S 0, S 2, S 3, S 0, S 2 and S 5, S 0, S 2 and S 5. In this particular 3 step not of t one is true not of t one is so due to this component these 3 steps are coming and AF c one is true in other 2 step your S 3 and S 6 so these are coming due to this zero on the other hand north of p one is also true over here so we are going to get this but, the remaining free steps S one S 4 and S 7 in this particular 3 step AF c one is not true and similarly, not of t one is also not true because t one is 2 in this particular steps so these 3 steps are not coming in to process.

So, t 1 implies AF c 1 is basically 2 in S 0, S 2, S 3, S 5 and S 6. So, what is not of this things already I have send that if phi is true in some step then not of phi will be true in the total state space minus the set of state, where phi is true. Since, this particular formula is true in S 0, S 2, S 3, S 5 and S 6, so not of t 1 implies AF c 1 in between S 1, S 4, S 7, these are the 3 steps S 1, S 4, S 7. Now, we have seen the labeling of this particular sub formula along with the negation of this particular sub formula. So, what is my next sub formula that E true until not of t 1 implies AF c 1. Now, we are going to check for this particular time.

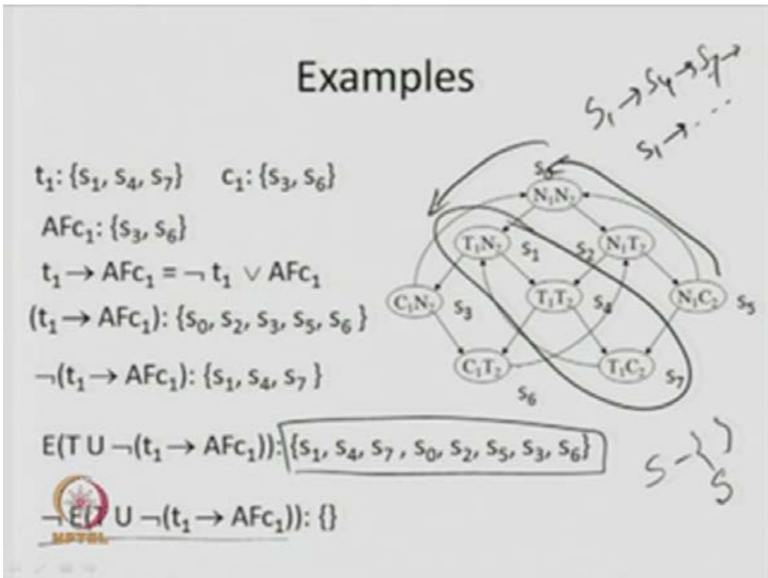
(Refer Slide Time: 20:12)



So, what is the procedure for E until operator? So, if any state S is labeled with q then labeled it with E p until q again, as for our semantic fuser, increase the present behavior. So, wherever q is true we are going to label it with E p until the time we are going to repeat this particular procedure until there is no change. What is the procedure label any state with E p until q if it is labeled with p and at least one of its successor is labeled with E p until though so.

First we are going to collect the step, where q is true will label this things with E p until q then we are going to traverse the entire state space. And we are going to check, if the p is true in the particular state and one of the successor is labeled with E p until q and until there is no change. That means we cannot collect any more steps, if we cannot collect any more steps then we are going to coming that our procedure over here there.

(Refer Slide Time: 21:16)



Now, in this particular case say we know not of t 1 implies AF c 1, we know these are the what are the states it is true in S 1, S 4 and S 7 for E true until not of t 1 implies AF c 1. This will be true in your S 1, S 4 and S 7, S 4 our for step in when is that S is labeled with q then labeled with it E p and until q. So, from this particular step what we are getting that this particular formula in between S 1, S 4 and S 7, because this not of t 1 implies AF c 1 is true in S 1, S 4 and S 7. So, these are the initial state we are collecting. Now, from this particular free space, we are going to travers the state space now next, then what will happen now? When you are going to look in to your S 1 then we will see one over which by this is our region S 0. Now, S 0 is labeled with true, now we have seen that it is label, because all the steps are label with true.

So, true until something, so if it is labeled with true and one of the successor is labeled with this particular your AE until your this things, because already we have got these are the 3 steps. Now, we are going to include this particular S 0 in this. So, it is not having any other predecessor then we go to S 4, from S 4, I can, we will see what are the predecessor, it is having one predecessor and this predecessor is also labeled with true. And now, since true is it is labeled with true and one of the successor is labeled with already O, these particular sub formula. So, S 2 will also come into these particulars similarly, and it is not having any other predecessor then we will go to S 7. Now, from S 7, we are going to see what are the predecessor it is having it is having.

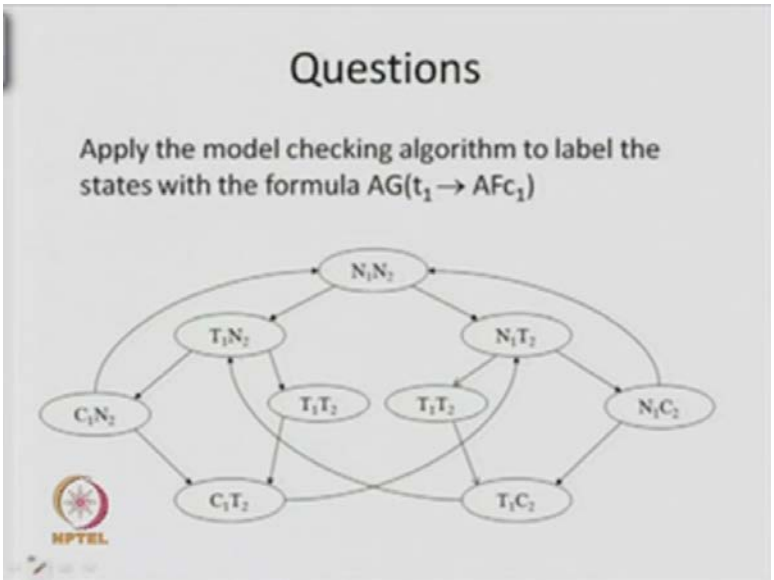
Of course, from S 4, we having S 1 also, but it is already labeled. So, from S 7, we having 2 predecessor S 5 and S 4, so S 4 is already labeled. Now, in S 5, we will see that it is already labeled with true and they are exist at least one successor which is labeled with your this particular given formula AF O, these things not of t 1 until EF c 1. That means, it is labeled with this particular given formula. So, S 5 also come into this particular step. So, in this situation, we have seen the predecessor of all these 3 steps and we have introduced. So, this time we are getting every guards. So, there is a change of my set of step. So, again we are going to repeat this particular loop. So, next time what will happen? Now, already we have seen these 3 things.

Now, we are going to see S 0, now when we are having S 0 then user see that what is the predecessor that we have. So, we have this particular predecessor, because we are having a transition form S 3 and we are having a predecessor form S 5. So, S 5 is already labeled, now we are going to see that S 3. Now, I can S 4 our definition which is labeled with 3 2 and it is having 2 successor. But one of the successor is labeled with particular formula, because already in the previous step we have labeled it. So, your S 0 will come in to this particular steps. So, S 0 is, oh sorry, S 0 is already included. So, S 3 will also will be included in this particular step. Now, already S 5 is there, so we, we already I have included it. Now, we have seen that again there is a change in this, particular steps. So, we will again enter into the loop, so and we enter into the next loop then what will happen. Already, we have checked those particular step, now only S 3 is remaining.

So, from S 3, we will see that what is this particular successor S 6 now we can in S 6 it is labeled with true. Now, one of the this things we are having these things what we call it is your S 3 is labeled with your, your the given formula E 2, until not of t 1 implies AF c 1. So, S 6 will be included in this particular step and the, so alpha this is the that we are having, so we have to include it. So, user see that since there is a change again we will go into it. Now, we will find that when we enter into the next step then all are included over a, because we have seen. So, all the steps has been included, so there will be no change. So, we are going to get in all those particular step; that means, entire step phase E true until not of t 1 implies AF c 1 is true.

Now, now my given formula is negation of these things. Since, so we are going get in all those particular step; that means, entire state space E true until not of t 1 implies AF c 1 is true. Now, now my given formula is negation of these things. Since, we are going to have S minus this particular step. So, since this is nothing, but equal to your states. So, we are going to get a null set; that means, now given formula is not true in non of this particular steps. So, this is the given we have loop for the true values of this particular formula and this is equivalent to my given formula. And we have founded none of the step this particular formula is true.

(Refer Slide Time: 27:02)



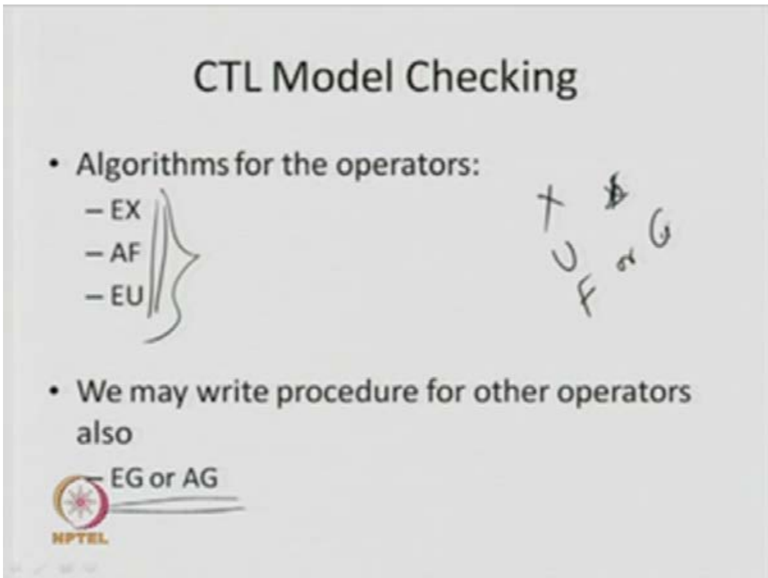
That means, our modeling of this things what will happen the model that we have no it is not satisfying all liveness property. So, if any process tries to enter into the critical section, eventually it may not get a chance to enter into the critical section, because in non of the step, this particular even formula is true. So, already we have explained that this is happening, because of this particular loop. So, we are coming from S 1 to S 4 from S 4 to S 7 and from S 7 to again we are going back to S 1 like that. So, we are going to get an infinite part in this particular.

So, from any step, if you look into other steps for every steps we are going to enter into this particular loop. So, if for example, say if I am in S 5 from S 5, I can come to S 0, from S 0, I can come to S 1. And after once, this into the S 1 and S 1, S 4, S 7, it will remain in this particular loop. So, due to this particular loop, this guidelines property is now true in our model. So, already we have explained these things, now we try to apply our model checking algorithm labeling, algorithm eventually I founded it is returning with the null step; that means, in none of the step the given formula is true.

Now, after that when I said we are coming up a model, we are trying to check for the property, whether my properties are true in this model or not. We have found that the liveness property is not true. So, in the particular case, what will happen designer is going to revisit the design he will analyze the design and he will try to modify the design in such way that the required property will be satisfied in the new model. So, already we have discussed this in previous lecture and, we have seen that we can come with a new model like that where the once the actually broken into 2 different steps over here. Now, I am giving a question that apply the model checking algorithm to label the states with the formula AG t 1 until AF c 1. So, this is the liveness property, already we have checked in my previous model.

Now, we apply the same procedure, the way I have explained it and see try to check whether this property is true in this particular model or not. So, that means, we have to find out the step of states why this particular formula is true. So, it is similar to my previous explanation on this, this particular model is different we had one simple states over here. Now, I have broken this particular state to 2 different state. Now, with this approach, now we try to look for try to get the equivalence and equivalence will come some sub formulas will come here also, because I am giving you the same formulas only this model is defined. Now, you apply the model checking algorithm to label the states with this particular formula, it is the very simple example simple problem. So, you can try it and thus say get a feeling how the models we are getting also over here. So, this is basically same formula, I have already explained that this given formula can be read up into this particular equivalent formula and these are the sub formulas that we are having. Now, we look for the state of steps where those particular sub formulas. So, it is similar to the example that I have explained, but here what will happen, I will slightly change the model. So, that this property will get satisfied you will take it as an exercise and do it.

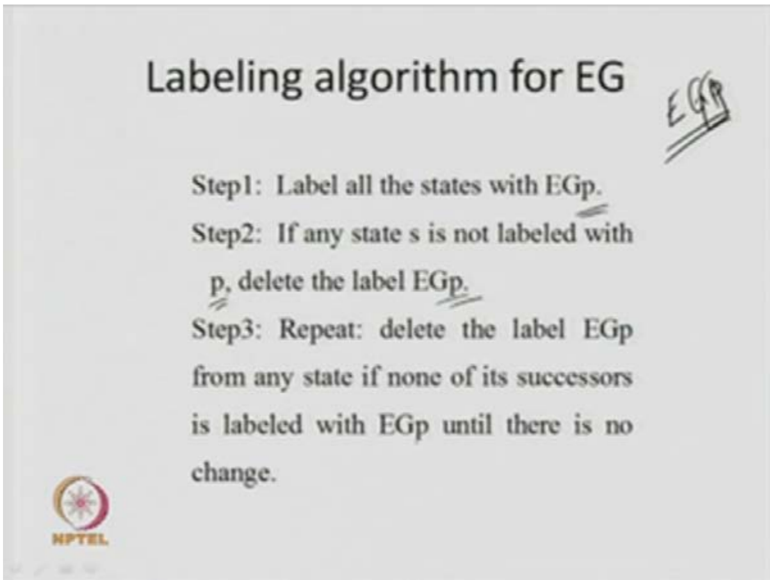
(Refer Slide Time: 30:39)



Now, what will happen, we have seen that minimum set of operators, we are going look for EX AF and AU and other operators can be explain with the help of these 3 operators. Now, we will see some, we can write some procedure for other operators called others operators also on that. So, for that I am going to take say some example of EG and AG, because if I having EG or AG, we have to write a equivalent formula which involve these 3 operator. But we will see, what you can write procedure for these 2 or not we can write a procedure for these 2 then what will happen? My adequate set of operator may be different, because already I have said, I need one next operator, we need one until operator and I need either F or G. So, for that I have looking into F. Now, for G all now we can say that these 3 operator, I can you see. So, I am going to see the procedure for EG and AG.

So, first we are going to look into EG, now what we are going to see that labeling algorithm for EG. So, it is slightly different, I am going to get it we will say the what is the first step. I am going to do label all the step with EG p, because I am going to look for the formula EG p. So, when I am coming to this particular formula, we know the labeling of this particular sub formula.

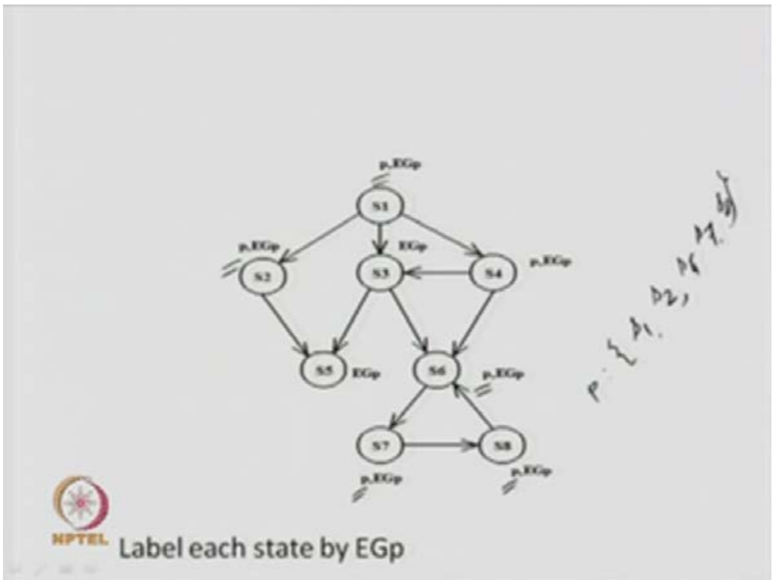
(Refer Slide Time: 31:48)



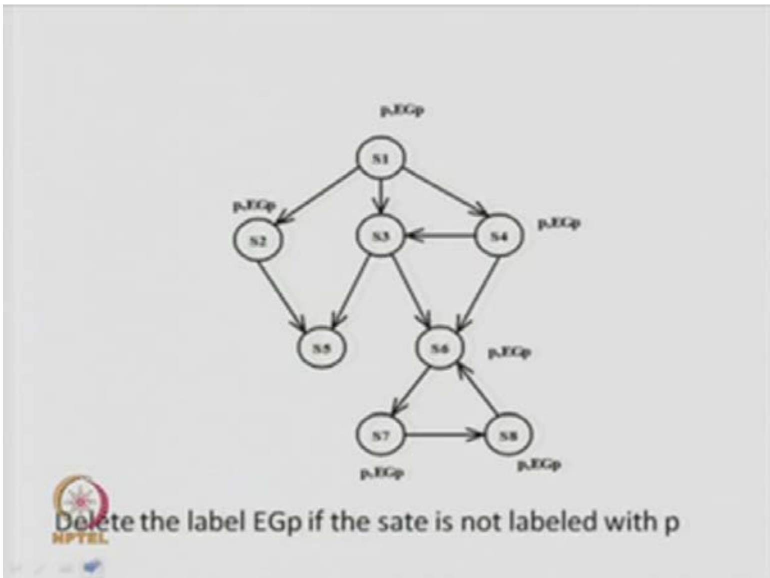
So, in this particular approach what we are going to do initially we are going to label all the step with you are the given formula EG p. So, if now step 2, what we are going to do if any state S is not labeled with p delete the label EG p. So, what we are starting, I am given a state space in all the step, I put the label EG p. Now, after I am going to check the steps where it is labeled not labeled with p, because we know the label of p know, when I am coming to EG p. So, I am going to remove the label of EG p from those particular step, where it is not labeled.

Now, in step 3, now what we are going to do delete the label EG p from any state if none of its successor is labeled with EG p. Now, in next state, what we are going to we are going to delete the label of EG p from ant state if none of its successor is labeled with EG p. So, and I am going to repeat this process until there is no step; that means, I am going to remove all the, the label. And after removing the label, what will happen, some more steps will come where the EG p is not true. And like that we will repeat this particular process, until there is no change. That means, we cannot remove any more labels. So, this is the procedure that we are going to follow say this is slightly different from a earlier approach. This is slightly different approach, but still it is going to work and I will see what is differences? Now, these the 3 step, we have to follow and let see whether it is going to work or not and it is simple example, I am going to explain.

(Refer Slide Time: 33:56)



(Refer Slide Time: 34:47)

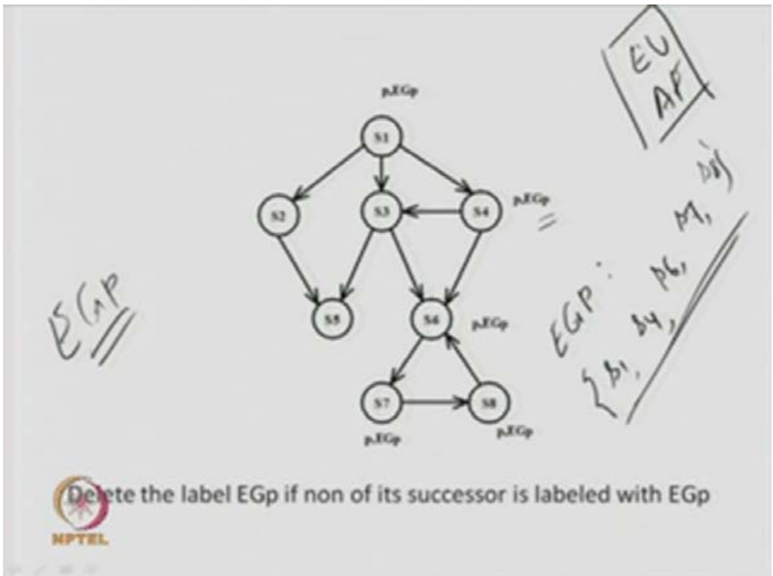


Now come up with this particular kripke structure or model, it is having the state space S 1, S 2, S 3, S 4, S 5, S 6, S 7, S 8 and initially I am having the label of p, when I am going for EG p, I must have the label of p. So, these are the step, where it is labeled with p, so basically I can say that p is true in the state S 1, S 2, S 6, S 7 and S 8. So, in the first step label each state by EG p, so I am labeling each state with the help of this particular EG p, now what is the next step.

Next step which says this remove the label level of EG p from the state where it is not labeled with EG p. So, I have seen that it is p is labeled in this particular state and it is not labeled in S 5 and S 6. So, S 5 and S 6 is not labeled with your p. So, I am removing the label EG p from this step S 3 and S 4. So, this is the step now what we are going to do next, I am going to do, going to remove the label from the state remove the label EG p from the state where none of the successor is labeled with EG p. So, in this particular case we find that already in the previous case, I have removed the label from S 3 and S 5 now if you look into this particular state space.

If I come for your say S 6 we will find that; that means, one successor is labeled with EG p when we come 2, I said we will find that one successor is labeled with EG p. When we come S 6 it is labeled one of the successor from, if 4 oh sorry, 6 is labeled S 1 also we are getting one is having 2 successor S 3 and S 4. So, one of the successor is label of EG p. But when we come to S 2 it is having on the one successor S 5 which is not labeled with EG p. So, that means now we are going to remove this EG p from S 2.

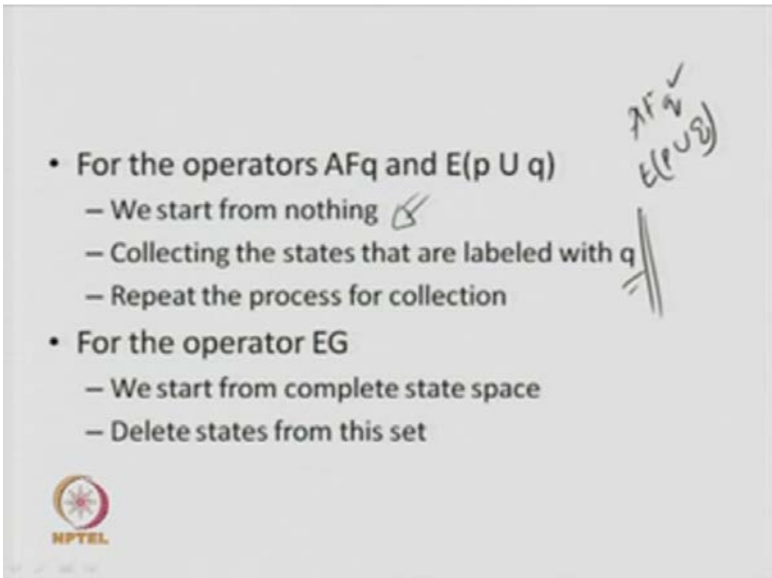
(Refer Slide Time: 36:15)



So, we are removing, now we need the label is we none of the successor is labeled with EG p. So, everyone then again, I am going to repeat this things and we will find that since we are removing it now this may be potential can deduct S 1. Now, in this particular S 1 what will happen you just see the from S 2 we have removed it. But it is having one successor S 4 where it is in labeled with EG p; that means, we cannot remove S 1. So similarly, we cannot remove the label of EG p from any of the steps.

So, eventually our algorithm stops over here. So, eventually what we are getting that is state of steps where EG p is true. So, we are getting the state like that S 1, S 4, S 6, S 7 and S 8 these are the steps because here from S 1 at least they are exist a part where. So similarly, for S 4 also because this is the sub part from this particular S 1. So, on here also we will see the which we are getting steps one part, where p is true in all the step. So, this is the state of steps that we are getting over here. Now, see we have seen the process for your EG p and the a process slightly different then the procedure that we have talked about EU and AF, we have discussed the algorithm for EU, AF. And now, we have seen the procedure for EG also the approach of EG is slightly different in the approach of EU and AF. Now what is this difference?

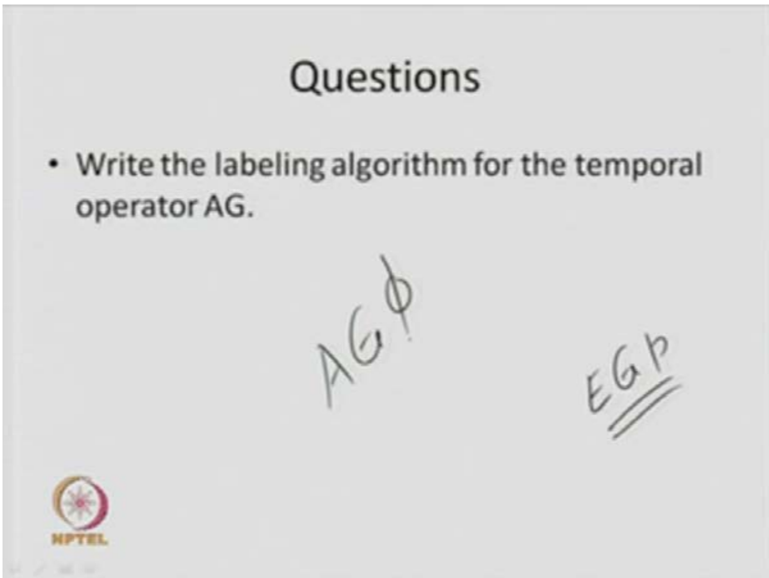
(Refer Slide Time: 38:14)



We consider operator AF and EU, basically we start from nothing. So, we are not considering any step or we consider initially your starting from a until step. Now, we are collecting the step that are labeled with q. So, in AF q and E p until q, so we know the, if a particular step q is true then AF q is true and in this particular step, if q is true then E p until q is true. So, we are collecting those particular step, where q is true; that means, we are starting with a minimal states this is the minimal state set of states where this particular formula AF U and EF U is true. And after that we are going to repeat our process; that means, while repeating the process we are traversing the entire state steps. And we are collecting more and more states, we are adding more and more state and until we are going to repeat this particular procedure till a point, where we cannot have any more steps. So, this is the procedure; that means, we are starting from a minimal state and we are trying to collect more and more state and eventually we will stop in the point where we can collect anymore state.

So, this is the approach that we have use in case of AF and EG. But in case of AF and E q EU we are exist a part until over here. But for the operator EG; that means, we are exist a part globally smoothing is true one, one. So, say globally p is true or not in this particular case our approach is here starting from the entire state steps. We are initially thinking the in entire state steps this particular formula is true.

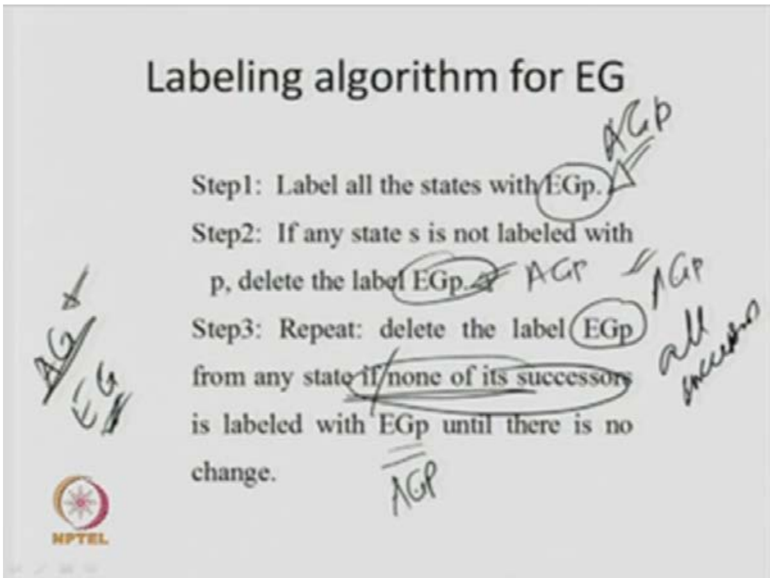
(Refer Slide Time: 40:40)



After that what will happen? We are trying to remove states from this particular state space by looking in to the criteria whether EG p is true. Over here also, this is you just see that approaches slightly different we are trying starting from the entire state space. And now eliminating states where EG p is not true. So, slightly approach is different one is we are trying starting for the minimal step. And in the second approach in EG p we are starting from the entire state space and trying to minimize it, I am going to get the state of steps where EG p is true.

Now, after discussing this particular EG p we are exist a part globally p holds or not. Now, I am giving a question to you, because already we have seen this particular algorithm write the labeling algorithm for the temporal operator AG; that means, in all part globally some phi holds or not. I am saying that try the labeling algorithm for the temporal operator AG phi. Now, already we have discussed about EG p now, I am saying that you write a algorithm for this I think this is the simple one. If you look into it what is the necessary of EG p and AG phi straight away, I think you can write the procedure for AG phi it is similar to EG what we did in EG.

(Refer Slide Time: 41:31)

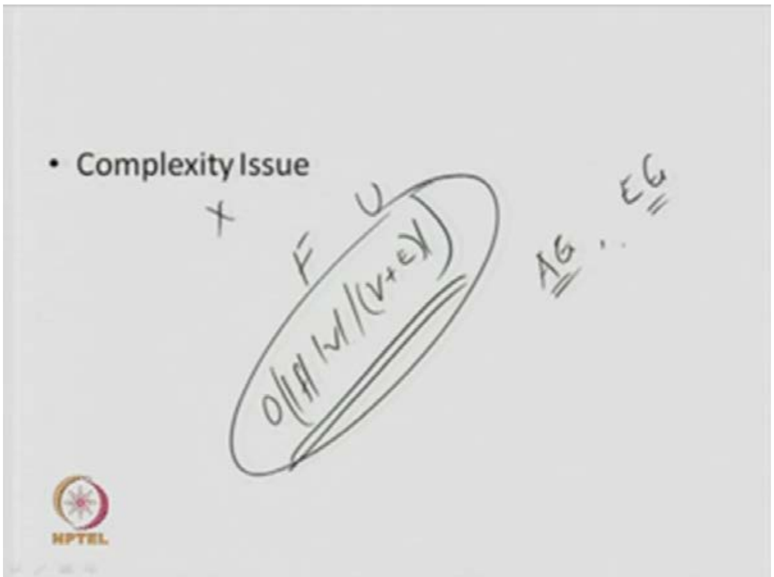


You just see that we are starting with the entire step where label all the states with your EG p. Now, if any state S is not labeled with p then delete EG p fine, because user AG and EG, AG is the stronger condition than EG, EG may be true in some step. But AG may be false, but if AG is true in some steps then EG must be true over here, because we are talking about all part and EG we are talking about some part on any part. So, if AG is true in a particular model then EG must be true, but if EG is true, AG may not be true. So, in this particular case, we can say that if p is not labeled then we are deleting EG p. Since, EG p is not true; that means, AG if there is no question of having AG p true over here. So, straightway we can use this.

Now, what we are repeating over here, delete the level EG p from any state if none of its successor is labeled with EG p. What we are saying then if none of its successor is labeled with this. So, we are considering it is in case of your EG p if none of its successor is labeled with EG p. Now, for AG I am already telling with the stronger conditions. So, instead of none of its successor, what we have to say for any state if none of its successor is labeled with EG p. We have to say all successor basically, we have to look for all successor, because it is your AG in all part globally and with the stronger condition when in this.

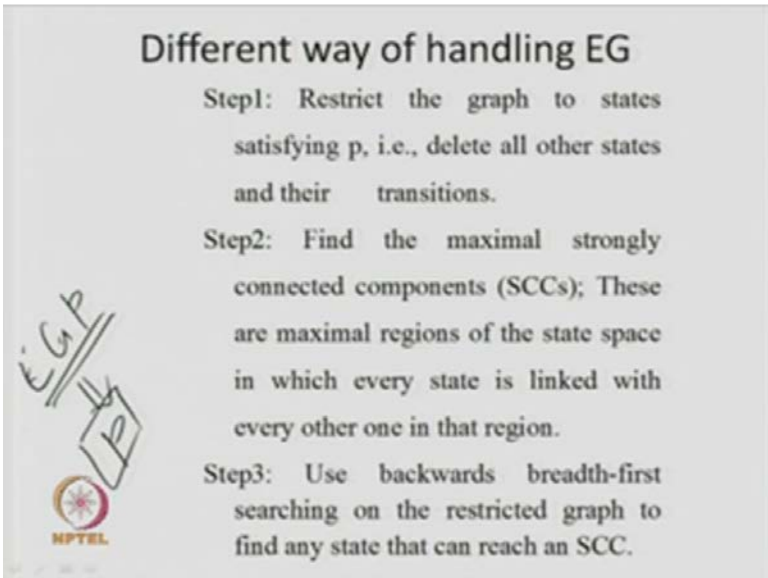
So, you just see the straightway you can say come for your EG instead of your none of the successor you can say that all successor are not labeled with you are AG p can remove then remove this label until this. So, label is state with your AG p if anyhow S is not labeled with p delete EG p from this particular state and delete the label AG p from any step if all successors are not labeled with AG p. Until there is no change you just see the since AG is the stronger condition, so from EG we can.

(Refer Slide Time: 44:01)



Now, what is the complexity issues of this particular and that AG or EG. Already I have seen that we have discussed about you in fuser and until, and of course, in both the cases we need this particular next. So, this is basically complexity, we have talked about that it is depends on the length of formula number of steps we are having and the total state space. Now, we have seen this is in case of F and U, now if you look into AG and EG though we are starting from the total state space. But again we have to traverse the entire state space we have to look for each and every step whether anyone of this successor all successor are all labeled with EG p or not. So, complexity will remain same or AG and EG also, but whether can we input this particular complexity. So, can we do in some other way. So, if you look for AG and EG we may have some other way to look for these 2 operator such we will see one this operators now.

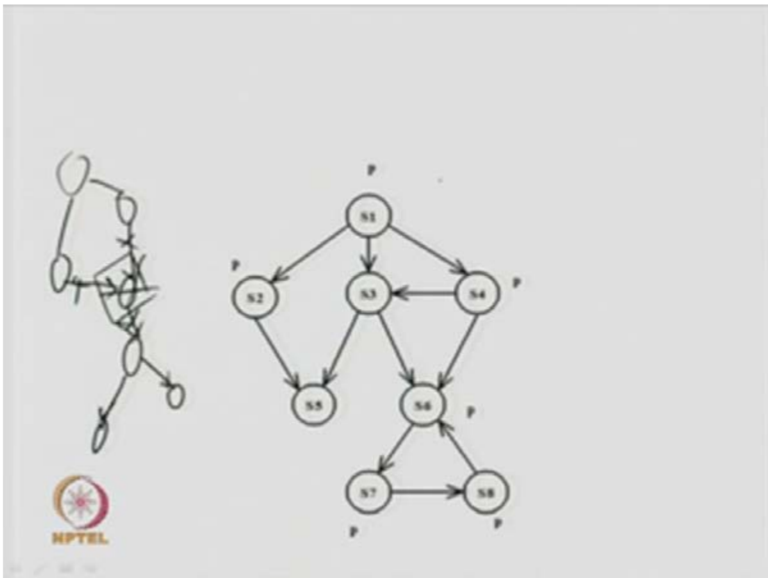
(Refer Slide Time: 45:17)



What is the way that we are going to delete, now this is slightly different approach. Now, see the algorithm in step 1, we are saying that restrict the graph to states satisfying p that is delete all other states and their transitions. Now, what you are saying you are giving me the entire state space when we are going to look for EG p; that means, what will happen when I come to EG p; that means, I know that is state space is already labeled by these particular formula p or. Now, we will take the entire state space and I am going to work with a restricted graph what we are going to say that delete the steps where p is not true and along with its transition. So, we are restricting it basically you see the going.

To look it which say if I am having a state space like that just explained here. Now, say these particular state is not have a label of p; that means, we are going to remove this particular states. And when I am removing this particular states then what will happen this transition will remain diagonal. So, we have to remove this particular transition also. So, initially we are starting with this things, so we are going to restrict our graph buy deleting the steps where p is not true an along with your transition. So, we are going to restrict the graph with the steps where p is true.

(Refer Slide Time: 46:14)

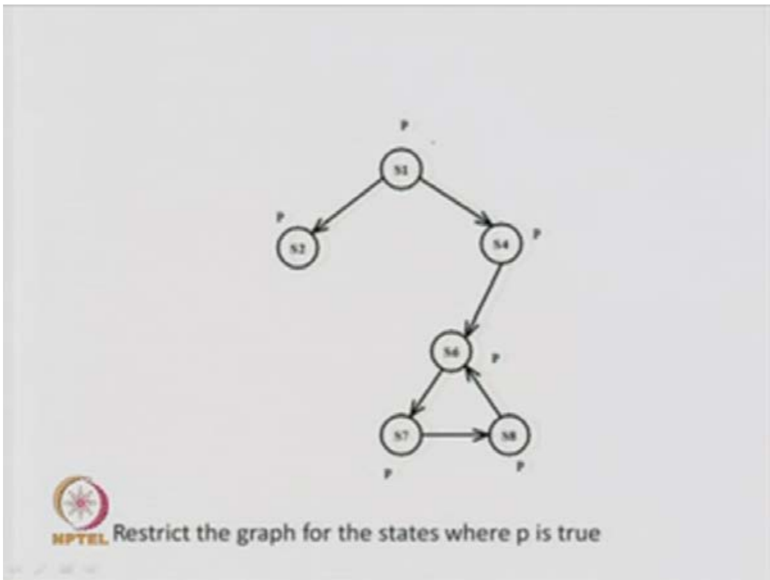


Now, next step is slightly what you are going to do find the maximal strongly connected component of these restricted graphs. So, this is maximal strongly connected component that we are going to perform, we are having a procedure to do this things. So, what is the maximal connected component these are maximal regions of the state space in which every state is linked with every other state. Basically, maximal strongly connected component, basically considered these are the complete cap, complete sub cap on my restricted graph.

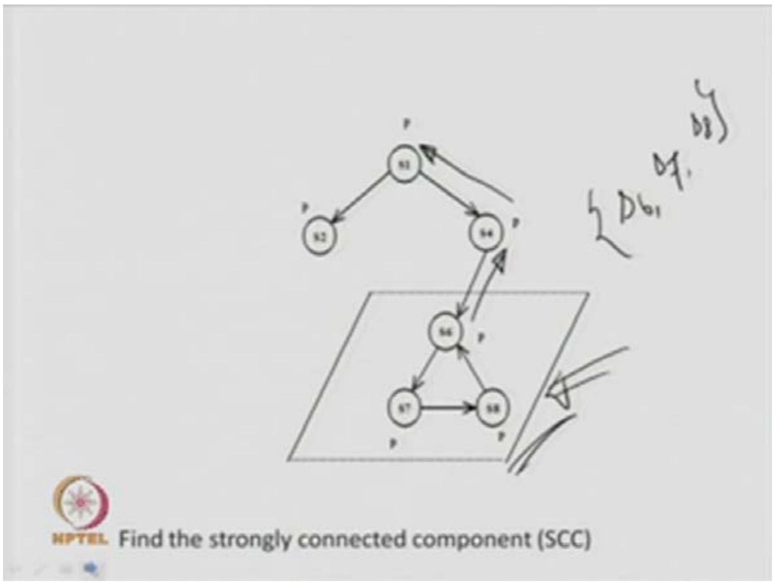
So, that means from any state, we can go to any other state. Of course, it is need not be component, but we are having a part from any states to any other states. So, this is maximal strongly connected component it need not be component. Now, in step 2 what we are going to use backward breadth first searching on the restricted graph to find any any state that can reach an SCC. Now, we are going to follow a backward breadth first search on this particular restrict graph to find any step that can reach an SCC. So, we are identifying the SCC s strongly connected component. Now, from this particular strongly connected component we use backward breadth first search. I am going to see whether form any state we can reach this particular state.

So, if we can this is the; that means, from the particular step also it will be true. Now, you see come to this particular some example that this is the state space that we are having; we are having state from S 1 to S 8 and we are going for EG p. So, in that particular case, what will happen with this? Already the state space is, is labeled with this space. So, these are the space where these true.

(Refer Slide Time: 48:55)



(Refer Slide Time: 49:51)



Now, what happens restrict graph for the states where p is true. So, in this particular case if you see out of those step S 5 and S 3 are not labeled with p. So, we are going to restrict the graph where p is true; that means, we are going to delete these 2 particular steps. When we are going to delete these 2 steps then along with that we have to delete all those particular transition. And we are coming to those particular steps and which are going on from these particular steps. So, for that we are going to get this particular restricted graph, now see here in all the states are labeled it here, because there will be, these are the steps will be candidate for EG p. They are exist a part globally p is true if your if any step p is not true then there is no question of having EG p true at that particular steps. So, we are restricting the graph where all p is true.

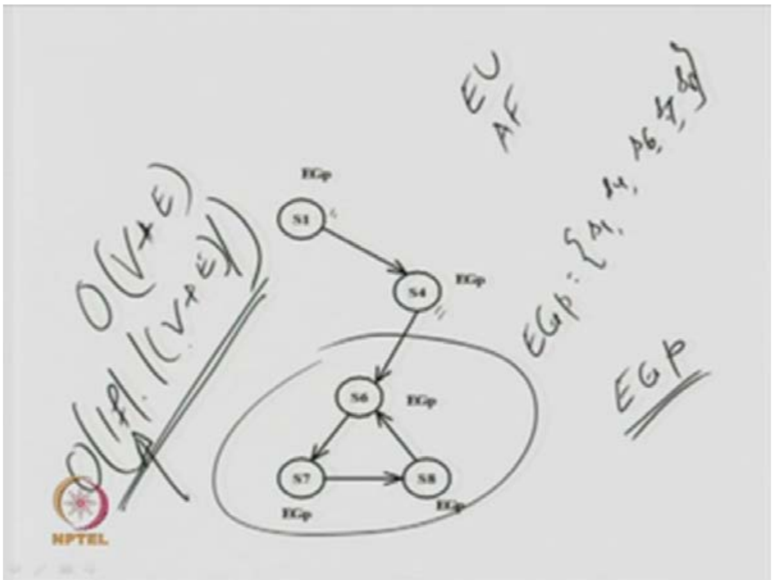
Now, next time what we are going to say that find the strongly connected component what is the strongly connected component. It is the maximum, maximal region where we can reach each state from any other state. So, if you look into this particular state space you will going to get this particular portion as a maximal strongly connected component. Now, which is having state S 6, S 7 and S 8, so see that if I am in S 6 then I can go to S 7, I can go to S 6. If I am in S 7 then I can go to S 8 and I can even go to S 6 also.

This particular part and if I am in your S 8 then I can go to S 6 and S 7. So, it is also whether if I am in any state, I can go to any other state, but now this things this is the, if you are in S 4 then you cannot go to S 2. So, that is why, X 4 is not come into this secondly, if it is consider S 4 from any one of this step, I cannot go to S 4. So, it will not come into the maximal strongly connected component. So, I am getting these as a maximal strongly connected component.

Now, what we are going to do, now we are going to have a backward breadth first search from this particular strongly connected component. And if any states can be reach by this particular backward breadth first search then we can say that that will be now formula EG p will be true. Now, you just seismic that when we are starting from this things backward breadth first search. So, we can reach your p and if you go for required then I can reach S 1. But from here, I cannot go for backward breadth first search and I cannot reach this particular S 2, because this is the directive. So, in this way these 2 steps will come into picture that means.

You just see that from your this maximal strongly connected component. We can use backward breadth first search to reach S 4 and this S 1 and, and no more sketch will come in to picture. So, after that what about we can reach from this particular maximal strongly connected component those sketch will be having easy picture, because you just see that.

(Refer Slide Time: 51:47)

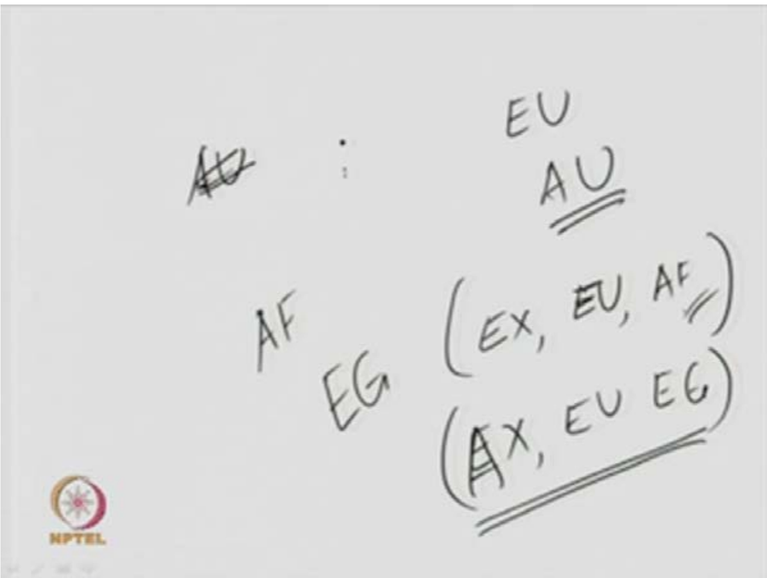


Now, wherever you go you find that p is true, because this is eventually I coming into this particular strongly connected component and wherever you go you will find that p is true. So, these are the state space where EG p is true. Similarly, we are getting that EG p will be true S 1, S 2 p, S 4, S 6, S 7 and S 8. Now, in this particular case you just see that what the advantage we are getting over here, whether any advantage we are getting you just see that what will happen. Since, we are getting this particular strongly connected component.

And along with that the backward breadth first search we are having algorithms which can be done in linear which is proportional to the, my state space number of vertices and number of edges. So, we are having algorithm for your backward breadth first search and finding strongly connected component which works in the order of V plus E. So, that is why we are having algorithm which works in order V plus E. So, that means, in this particular case the model checking algorithm that we are getting down time complicity will be it is depends on the length of the formula. And U size of U state space or size of an graph this is some sort of your algorithm. So, eventually the complexity will turn up to be like that.

So, for EG p now what complexity we are getting it is linear on the length of the formula and this linear on the size of my graph. But in case of your EU and you are AF the complexity is basically linear in the length of the formula. But it is quadratic in the size of my state space, it is quadratic in the number of nodes that we have in my state space. So that means, if you say that instead of AF, because EU I cannot avoid EU, I cannot avoid because A.

(Refer Slide Time: 54:36)



AU, I cannot avoid sorry, I can say that EU, I cannot avoid, because I am having the operator AU. AU have to be express with the help of EU. Now, I have this AF, but we are having equivalence. So, instead of AF either I can go for EG, because we know that we are having equivalence. So, we can look for one complete set of operator EX, EU and AF. This is one set another operator, I can say EX, EU and say EG, one can look for Ax also it is similar.

So, if we are going to have these things or I can say that operator, I can say that it could be AX also. If i am going to look for these 3 operator and what will happen that complexity time complexity will be slightly less. And if we are going to look for these 3 sets and your time complexity will be slightly more, because it is quadratic in the size of my, but here we are getting linear operators this way. Either we can go for this particular procedure, I can go for this particular set of states also.

So, basically we are going to look for 3 set of operators, 3 operators and others will be express with the help of these 3 operators. So, this is the another approach, we have seen and we have seen that the time complexity may be slightly reduce. With this I will stop here today. In next class, we may look for some examples where we are going to have a going to make model and going to see what are the property, it needs to be satisfied and apply the model checking algorithm to do it.

Bye, bye.