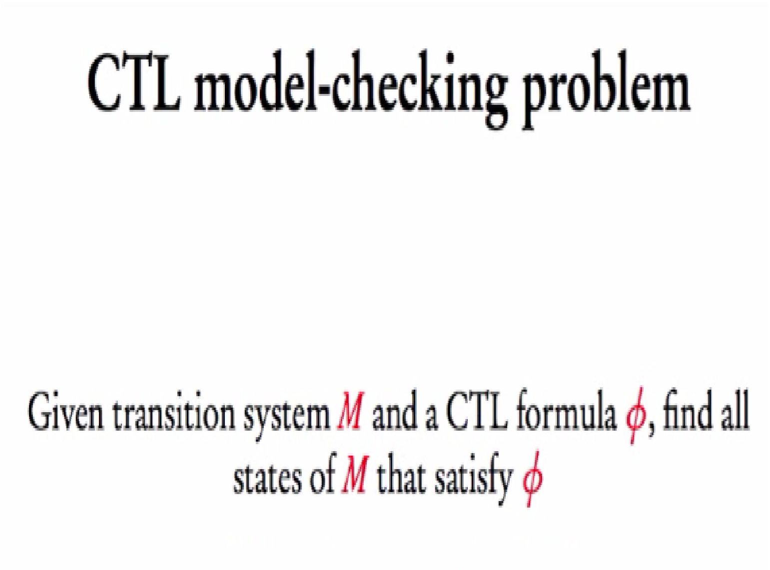
Model Checking
Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology – Madras

Lecture - 46
EX, EU, EG

Welcome to the second module of this unit. In the last module, we saw that a CTL formula can be rewritten in terms of EX, EU and EG. We called it the existential normal form. In this module, we will look at algorithms for CTL formulae of the form EX phi, E phi 1 until phi 2 and EG phi.

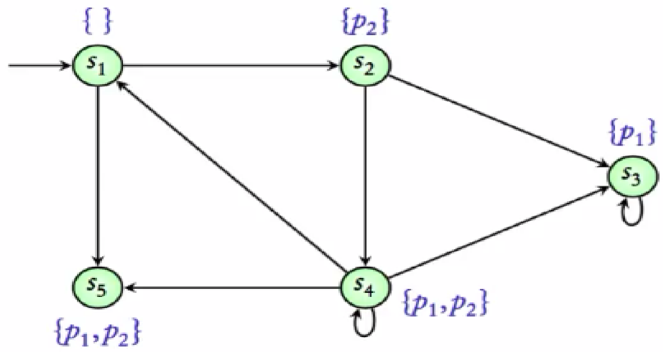
(Refer Slide Time: 00:41)



So here is the CTL model checking problem, given a transition system M and a CTL formula phi, find all states of M that satisfies the formula. This is the problem that we are interested in, in this unit. So in this module, we will look at the special case when phi is of the form EX phi 1, E of phi 1 until phi 2 or EG phi.

(Refer Slide Time: 01:23)

$$\mathbf{E}\,\mathbf{X}\,(p_1 \wedge p_2)$$



Let us start with algorithms for EX phi. This is the given transition system and these are the labelling of the states with atomic propositions. We want to check if the formula EX p1 and p2 is true in this transition system. How would we do it? As a first step, find the states where this sub-formula p1 and p2 is true. Here it is not true, here it is not true, here it is not true, here since the labelling contains both p1 and p2, the sub-formula p1 and p2 is true in s5, P1 and p2 is true in s4.

Once you do this to check the states when EX p1 and p2 is true, look at the states which have a transition to a state labelled with p1 and p2, okay. Look at s1, there exists an edge going to a state, which satisfies p1 and p2. Similarly, look at s2, there exists an edge which goes into a state that satisfies p1 and p2, so s2 satisfies this formula.

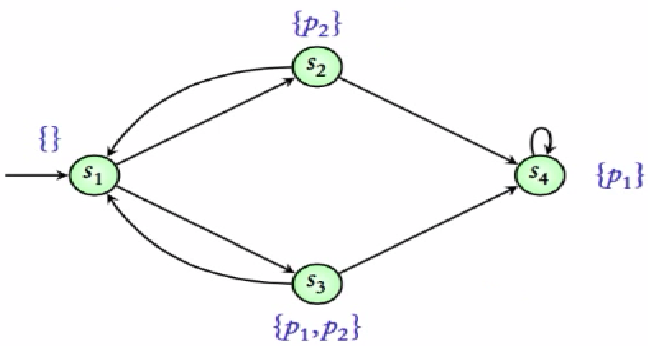
Similarly, look at s4 there exists an edge, so this is an edge right, there exists an edge back to s4 and s4 satisfies p1 and p2, hence s4 satisfies EX p1 and p2 as well, okay. If you look at the computation tree starting from s4, there will exists a path which satisfies X of p1 and p2 and finally this satisfies EX p1 and p2 as well, because of this transition.

Let me repeat this, the first step was to label the states where p1 and p2 is true. Then what you can do is consider each of these states, look at s4 first, look at all incoming transitions as in the transitions that come into s4, those states, so here there are two incoming transitions, one from s2 and one from s4 and the sources of these two incoming transitions will satisfy EX p1 and p2.

Because from these states there exists a transition to a state satisfying p1 and p2, which means from these states if you look at the computation tree, there exists a path satisfying X of p1 and p2. We have looked at s4, now look at s5. It is labelled by p1 and p2, look at all incoming transitions. There are two incoming transitions, s1 and s4, s4 is already labelled with EX p1 and p2, you need to label s1, okay.

(Refer Slide Time: 05:11)

$$\mathbf{E\,X}\,(p_1 \wedge \neg p_2)$$

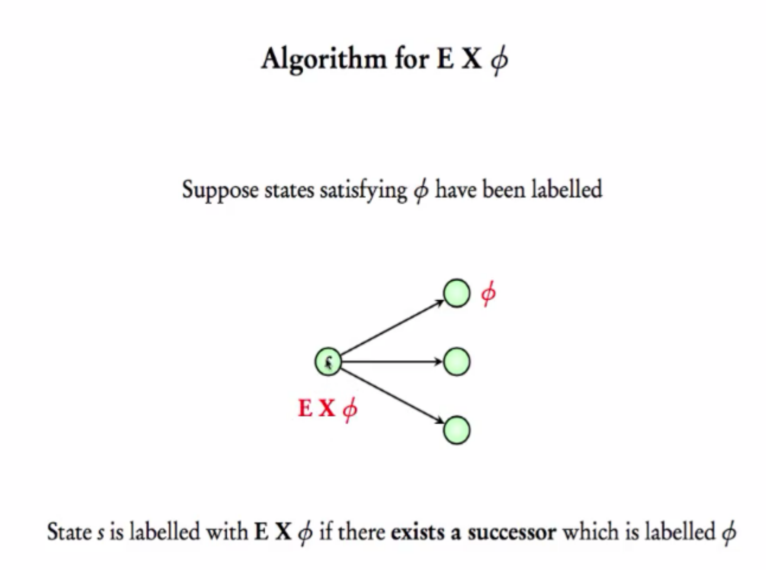


Let us see another example, this is the transition system that is given to us and we want to find the states of this transition system, which satisfy the formula EX p1 and not p2. The first step, look at this sub-formula, it is made of p1 and not p2. Let us first mark the states that satisfy this sub-formula p1 ant not p2.

s4 is one state, because it satisfies p1 and p2 is not there, so it satisfies p1 and not p2. In fact, s4 is the only state, because s3 satisfies p2 and s2 does not satisfy p1, s1 does not satisfy p1, okay. So s4 is the only state which will be labelled p1 and not p2. The algorithm for EX of p1 and not p2 what it will do? Once it has labelled this it will look at all incoming transitions to this state.

There are three incoming transitions, all these states would be labelled with EX p1 and not p2. So this state is labelled, this state is labelled, and this state is labelled. So this will be the set of states which satisfy EX p1 and not p2. The initial state does not satisfy EX p1 and not p2, hence the transition system as a whole does not satisfy EX p1 and not p2, but there are these states in the transition system, which satisfy EX p1 and not p2.

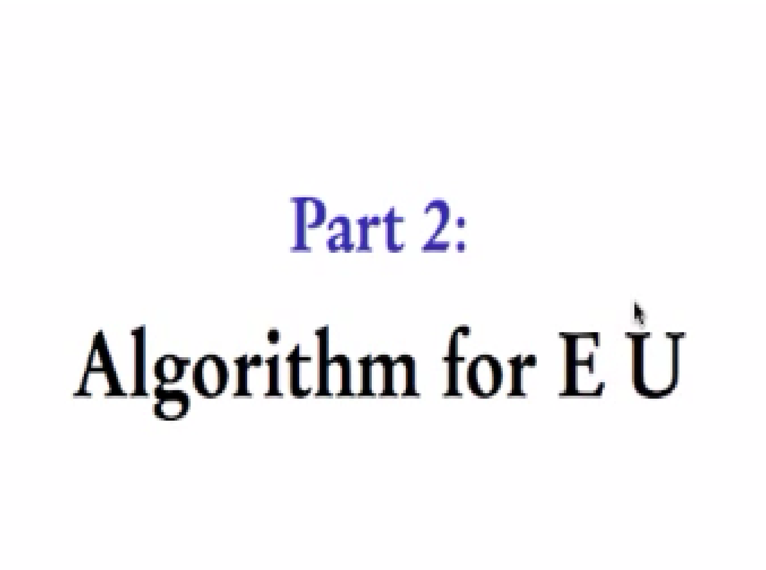
(Refer Slide Time: 07:10)



Let me summarize the algorithm for EX phi. First we need to label states satisfying phi, and then once you do that for every state satisfying phi, you look at its predecessor. In other words, a state s is labelled with EX phi if there exists a successor labelled phi. This is the algorithm, it looks at the every state of the transition system and marks phi first.

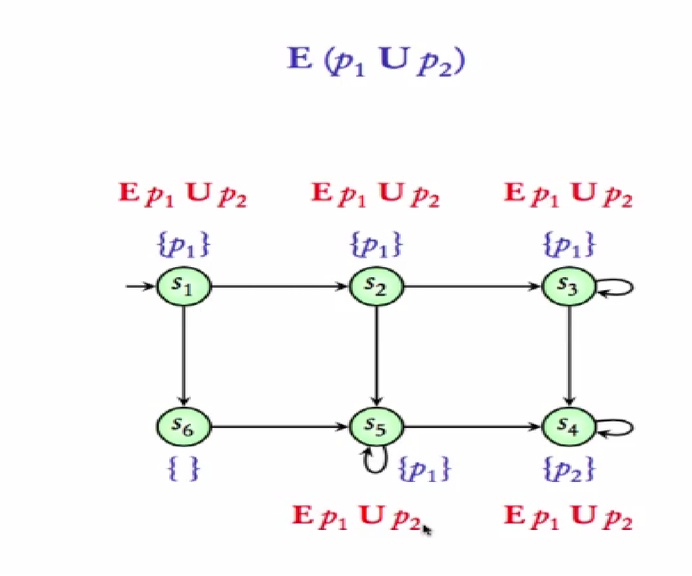
Once it is done, it will look at predecessors of the states that have been marked with phi, and those predecessor states will satisfy EX phi.

(Refer Slide Time: 08:04)



So we are done with algorithm for EX, now let us see algorithm for E of phi 1 until phi 2, which is shortly written as EU.

(Refer Slide Time: 08:18)



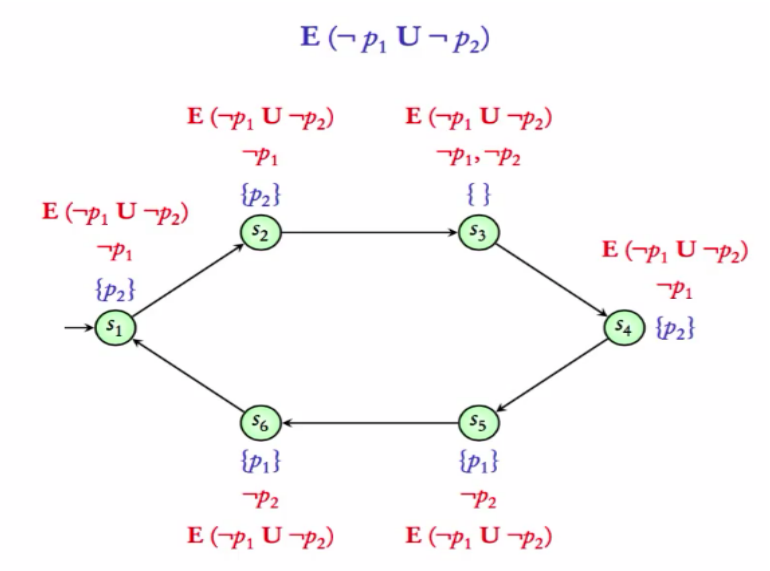
This is a transition system, we want to check, which state in this transition system satisfy E of p1 until p2. First step, look at all states here which satisfy p2, s4 satisfies p2. So clearly s4 will satisfy E of p1 until p2. We assume that there are no terminal states. So there will exist a path from s4 and in that path, the initial state itself satisfies p2, so s4 satisfies E of p1 until p2.

Now what? Look at the incoming transition to s4, s5, s3 and s4. In these incoming transitions, if the state satisfies p1 then label those states with E of p1 until p2 as well. s5 satisfies p1 and there exists a path which satisfies E of p1 until p2. So s5 satisfies E of p1 until p2 as well. If you look at this path, s satisfies p1 and s4 satisfies p2, sorry s5 satisfies p1 and s4 satisfies p2.

Similarly, s3 satisfies p1 and s4 satisfies p2, so this path satisfies p1 until p2. So at s3 there exists a path satisfying p1 until p2, fine. Now, continue this process, look at the predecessor of these newly labelled states. Look at s5, look at the predecessor s2, s2 satisfies p1 and there exists a path to a state that satisfies E of p1 until p2, which means that this state will also satisfy E of p1 until p2, correct, because if there exists a path to a state which satisfies this, that means from this state there exists a path which satisfies p1 until p2.

So from here also there will be a path which will satisfy E of p1 until p2, okay. So this is a newly labelled state. Continue the process, look at the predecessor of this state. There is one predecessor and it satisfies p1. So mark it with E of p1 until p2. Note that, when we look at the predecessor of s5, we also had s6. But since s6 does not satisfy p1, we will not mark it with E of p1 until p2. So these are the states of the transition system, which satisfy E of p1 until p2.

(Refer Slide Time: 11:57)



Let us look at another example, we want to check the formula E of not p1 until not p2 on this transition system. Let us mark the states of this transition system that satisfy not p1, not p2. This satisfies not p1, this satisfies not p1, this satisfies both not p1 and not p2, this satisfies not p1, s5 satisfies not p2 and s6 satisfies not p2. Now what does the algorithm do, it first looks at all states that satisfy not p2.

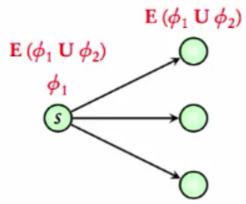
So this is the second formula here know, something until something. So it looks at all states that satisfy not p2 and labels them with E of not p1 until not p2, because if you look at this state there exists a path and already this state satisfies not p2, correct. So this state will automatically satisfy E of not p1 until not p2, same for s5, same for s3. Let us now look at the predecessors of these states.

Look at s5, the predecessor of s6 is already marked, so let us look at s5. The predecessor of s5 is s4 and it satisfies not p1, so it has to be marked with E of not p1 until not p2. Similarly, the predecessor of s3 also satisfies not p1, so it should also be marked with E of not p1 until not p2. Now look at the newly labelled states, this one and this one. The predecessor of this satisfies not p1, so this has to be marked with E of not p1 until not p2.

Now there are no more states left to be marked, so the algorithm will terminate.

(Refer Slide Time: 14:16)

Algorithm for E (ϕ_1 U ϕ_2)



- If any state is labelled with ϕ_2, label it with **E** (ϕ_1 **U** ϕ_2)
- ***Repeat***:
  Label any state with **E** (ϕ_1 **U** ϕ_2) if it is labelled with ϕ_1 and at least one successor is labelled with **E** (ϕ_1 **U** ϕ_2)
  until no change

Let us now summarise the algorithm for E of phi one until phi two. Assume that states with phi 1 and states with phi 2 are labelled. If any state is labelled with phi 2, then should label it with E of phi 1 until phi 2, that is the first step. Then we need to repeat this process. We need to label any state with E of phi 1 until phi 2, if it is labelled with phi 1 and at least one successor is labelled with E of phi 1 until phi 2.

We need to keep doing this until there is no change. Let us see, suppose this state is marked with E of phi 1 until phi 2. Look at state s, it is marked with phi 1, in that case this should also be marked with E of phi 1 until phi 2, okay. So this is the algorithm, you can check this with the previous examples. In the previous examples, we did E of not p1 until not p2. So we first marked the states with not p1 and not p2.

Next we looked at states where not p2 has been marked. And those states were labelled with E of not p1 until not p2. And then we did this step repeatedly until there was no change, you can have a look at it.

(Refer Slide Time: 15:50)

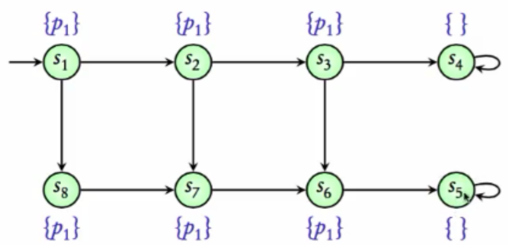
Part 3:

Algorithm for E G

We have seen algorithms for EX and EU, let us now look at an algorithm for EG

(Refer Slide Time: 16:00)

E G p_1



No state of the above transition system satisfies **E G** p_1

So here is a transition system, we want to check the formula EG p1. Here the algorithm would be bit different from the last time. In the first step, label all the states with the formula EG p1. Now you look at states which do not satisfy p, s4, s5 do not satisfy p1. So first remove the labels from them. Now look at the predecessor of s5 which is s6. Look at the paths out of s6, or rather the transitions out of s6.

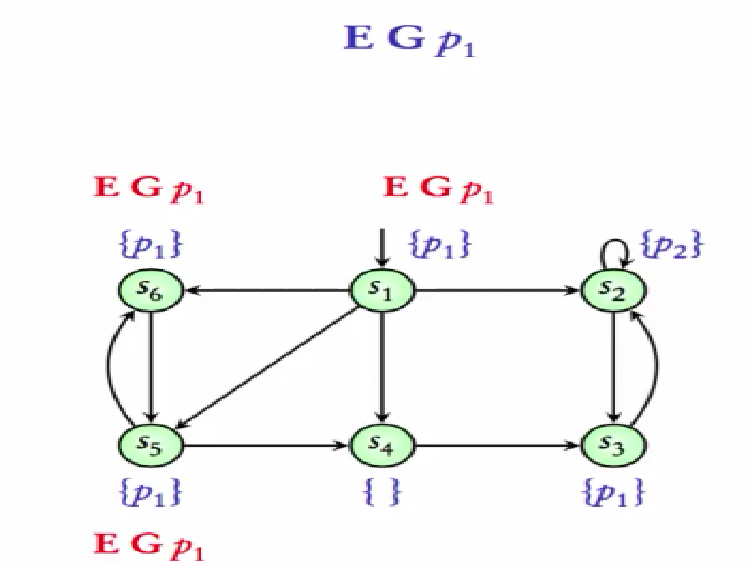
There is only one transition out of s6, and this transition leads to a state which does not satisfy EG p1. So we need to remove EG p1 from this state as well, because the only path out of this state does not satisfy G p1. So remove it from s6. Now look at the predecessors of s6, s3 and s7. Look at s3, look at all transitions out of s3. There are two of them, one going to s4, and one going to s6. Both of them do not satisfy EG p1.

As in we have removed the label from both of them. In that case, we need to remove it from here as well, because there does not exist a path satisfying G p1 from this state. Similarly look at s7, the only transition out of it leads to a state, which does not satisfy EG p1. So remove the label from s7 and s3 as well. Look at the predecessor of s7 now, s2 and s8. Look at s2, all transitions out of s2 go to a state where EG p1 is not present.

So you should remove it from this state. Similarly, for s8, the only transition out of it, so all transitions out of it lead to a state where EG p1 is not labelled, so we removed it from here. Finally look at this state, all transitions out of it lead to a state where EG p1 is not marked. So EG p1 has to be removed, here as well. In fact, no state in this transition system satisfies EG p1, you look at any state.

Finally, it has to end up either an s5 or an s4, any path from that state will have to end up in s4 or s5, and both s4 and s5 do not satisfy p1. No state in this transition system satisfies EG p1.

(Refer Slide Time: 19:25)



Let us look at another example, this is a transition system and we want to check EG p1 on this transition system. First step, mark every state with EG p1. Second step, remove this label from states which do not satisfies p1, s4, s2. You remove the label from s4 and s2, because they do not satisfy p1. Now look at the predecessors of these states. Let us start with s2, its predecessors are s3 and s2 itself.

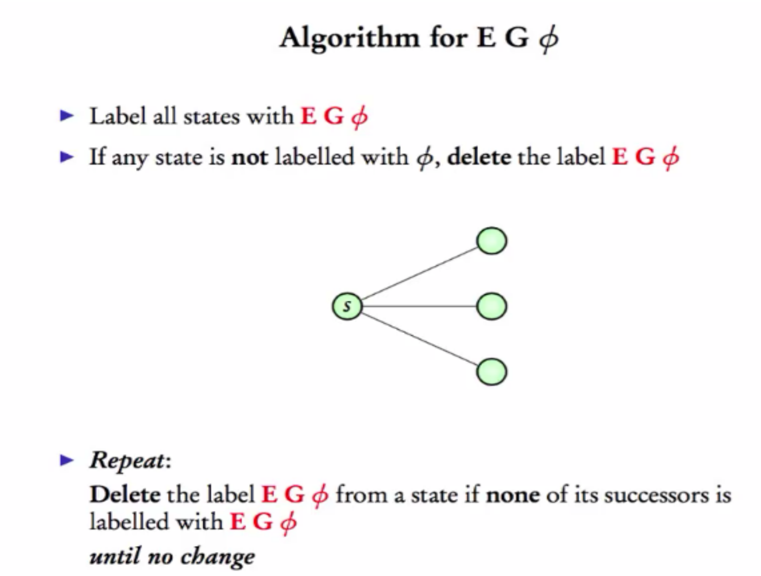
We have already removed the label from s2, now let us look at s3. Look at all transitions out of s3. There is only one transition and it goes to a state which is not labelled with EG p1 so the label at s3 should be removed. Now look at the predecessor of s4, say s5. s5, s6 and s1 are still marked with EG p1, let us see what happens. Look at s5, there exists an edge which leads to a state satisfying EG p1. So we do not remove the label as of now.

Look at s1, there exists a transition that goes to a state satisfying EG p1. So we do not remove the label from this state as of now. What about s6, there exists a transition leading to a state which satisfies EG p1. So we do not remove the label from this as well. In fact, these three states indeed satisfy EG p1. Since there is no change in the labels the algorithm will start. The algorithm starts like this; all states are marked with EG p1.

Then states where p1 is not present, you remove EG p1. Then look at a state, if all edges out of it lead to a state where EG p1 is not marked, then remove the label. Now in the rest of the states check this property, from every state is there an edge which leads to a state marked with EG p1. In that case do not modify this label, with this property these three states the labels will not be removed, and the algorithm will stop.

So if you look at this state, there exists a path that goes to s6, (s5, s6), (s5, s6) which satisfies G p1. Similarly, from s6 you can keep doing this again and again, and G p1 will be satisfied, same for s5, do (s6, s5), (s6, s5), and so on and that path will satisfy G p.

(Refer Slide Time: 22:50)



Let us now summarise the algorithm for EG phi. Step one label all states in the transition system with EG phi. Second step, if any state is not labelled with phi, then delete the label EG phi. Then, there is going to be a repeated step, delete the label EG phi from a state if none of its successors is labelled with EG phi. So let me illustrate it. First all states are marked with EG phi, then by doing this process if you end up with this picture.

That is the exists a state from which none of its successors is labelled with EG phi, then we need to remove the label from the state as well. We need to keep doing this process till it saturates in the sense that, we need to keep doing this till there is no change in the labelling of the transition system. Once you do this, the states which are labelled with EG phi are the ones from where there exists a path satisfying G phi.

So this is the algorithm for checking EG phi on a transition system. So in this module, we have seen three algorithms, given a formula of this kind, we have seen how the algorithm labels the states that satisfy the corresponding formula. In the next module, we will look at a generic algorithm for any CTL formula.