Embedded Systems - Design Verification and Test

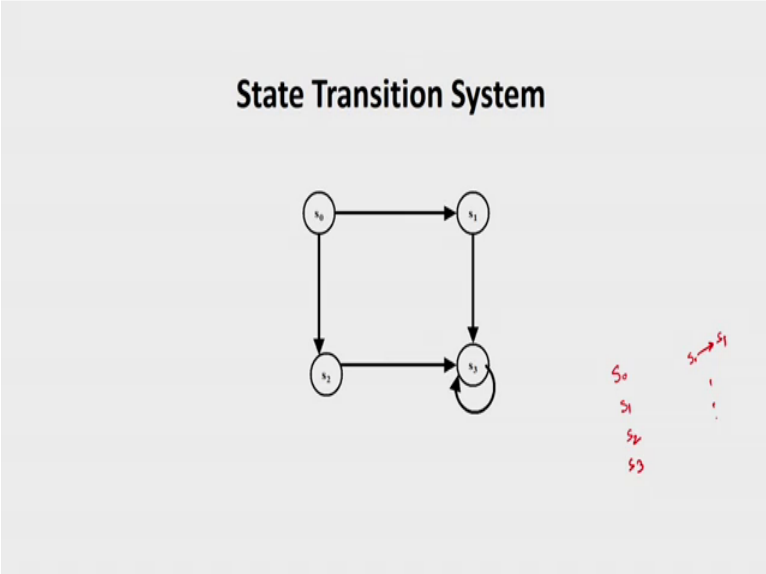
Dr. Santosh Biswas
Prof. Jatindra Kumar Deka
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture - 24
Use of OBDD's for State Transition System

Hello everybody, welcome back to the online course on Embedded System Design Verification and Test. So, in our last class we have introduced about BDD, Binary Decision Diagram and we have introduced what is Ordered Binary Decision Diagram: OBDD and ROBDD, Reduced Ordered Binary Decision Diagram. And, we have seen that, if you follow a particular variable ordering then ROBDD representation of a given Boolean expression is unique~~;~~. ~~that~~ That means, it is the canonical representation of the given Boolean expression, if we use ROBDD with a variable ordering. If we sense the variable ordering then, the structure of BDD best sense for ROBDD.

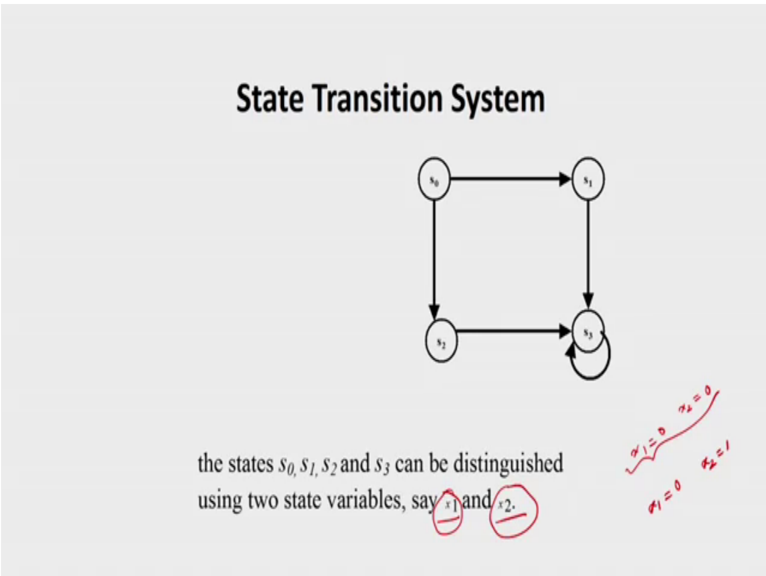
Now, today we are going to see what we can use or where we can use that BDD. So, we are saying that ROBDD can be used to representation of a state transition system and finally, that ROBDD representation of the state transition system will be used for model checking algorithm~~,~~. ~~in~~ In that case the model checking algorithm will be known as symbolic model checking. So, our basic objective is to find out how to represent a given transition system with the help of OBDD's

(Refer Slide Time: 02:15)



So, this is a state transition system so, we are having four states S0, S1, S2 and S3 and it is having some transition like say S 0 to S 1 like that, we having transition and this is a simple transition system that I am representing over here. And now, we are going to see how does information of the state transition system can be captured by BDD's or in particular OBDD's: Ordered Binary Decision Diagram.

(Refer Slide Time: 02:49)

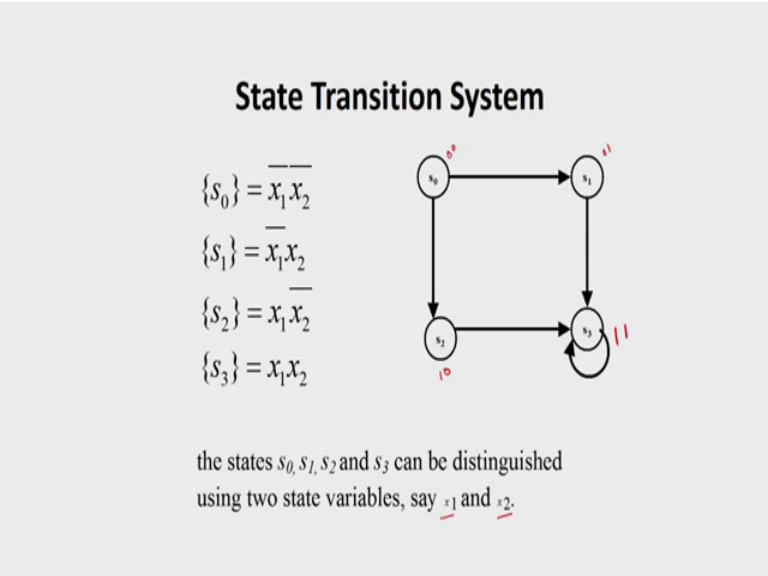


Now, to have this particular transition system and to represent the state, we need some encoding scheme and basically, we have going to use the state variables ok. So, since I am having four different states over here we need to state variable and we said these are x 1 and x 2. And when we correlate this transition system with Kripke structure, which is used for your model checking then, this x 1 and x 2 may be considered as an atomic proposition of the system.

And, these values may take or this variable may take the value either 0 or 1; that means, it may happen that x 1 equal to 0 and x 2 is equal to 0. So, this combination is going to represent one of this particular step and what is the meaning of these things; that means, you can say that now signal value of the variable x 1 is 0 and signal value of the variable x 2 is 0 or the atomic proposition values are 0.

Similarly, we may get another configuration x 1 is equal to 0 and x 2 equal to 1 in that particular case we are getting another state, where you can say that signal value of x 1 is 0 and signal value of x 2 is one or we can say that atomic proposition x 1 is false and atomic proposition x 2 is true. So, this is the way we are representing the states and if, you correlate with Kripke structure then we say that this is the labelling of the atomic proposition on those particular step ok.

(Refer Slide Time: 04:31)

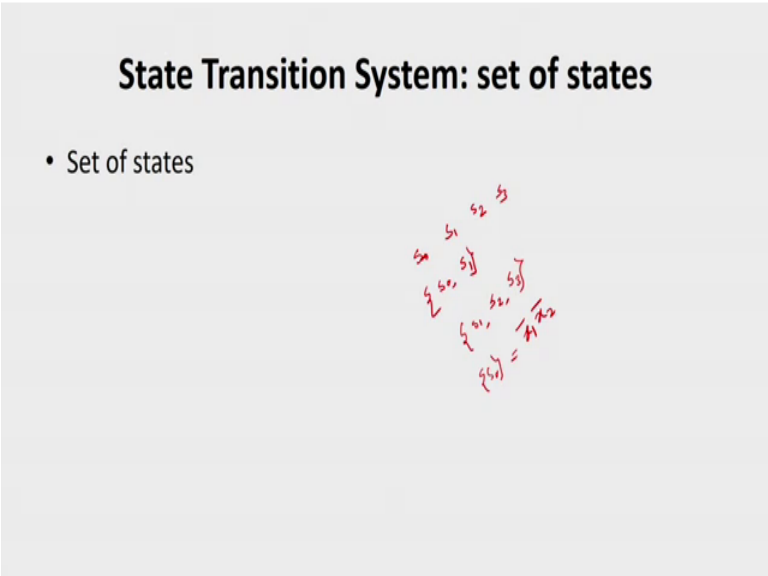


Now, for that we are using these 2 state variable x 1 and x 2. So, these 4 step can be represented like that S 0 the set S 0 is your x 1 bar x 2 bar; that means, the values of both the atomic proposition are the state variables are 0 0 over here; that means, we can say this is the state 0 0. S 1 is your x 1 bar x 2; that means, we can say S 1 is 0 1 S 2 is your x

1 x 2 bar, so S 2 is your 1 0 and S 3 is your x 1 x 2, so this is your 0 0, sorry this is your 1 1.

So, these are the 4 possible combination and we are representing those particular state with the help of these 2 state variables or you can say these are the atomic proposition and one end indicates that both the atomic provisions are true in this particular step. Again 0 1 indicate that atomic proposition x 1 is false and atomic proposition x 2 is true. So, this is special tangent system along with the labelling function.

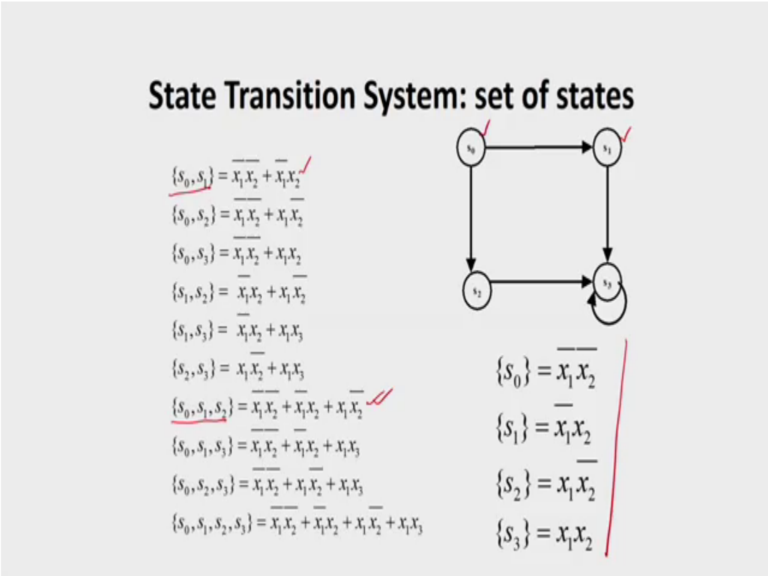
(Refer Slide Time: 05:40)



Now, we may consider the set of states. Now you just say that I am having four state S 0 S 1, S 2, S 3; maybe we are interested for a particular state of states say S 0 S 1 or we may be interested for another set of state S 1, S 2, S 3 ok.

Now, how we are going to represent those particular states. I think you can visualize that since I am representing state S 0 by atomic or say by their state encoding; that means, I can say that x 1 bar x 2 bar. Now, if we having more stat; that means, you have to collect all those particular state and finally, we are going to get a representation for those particular state of states.

(Refer Slide Time: 06:32)

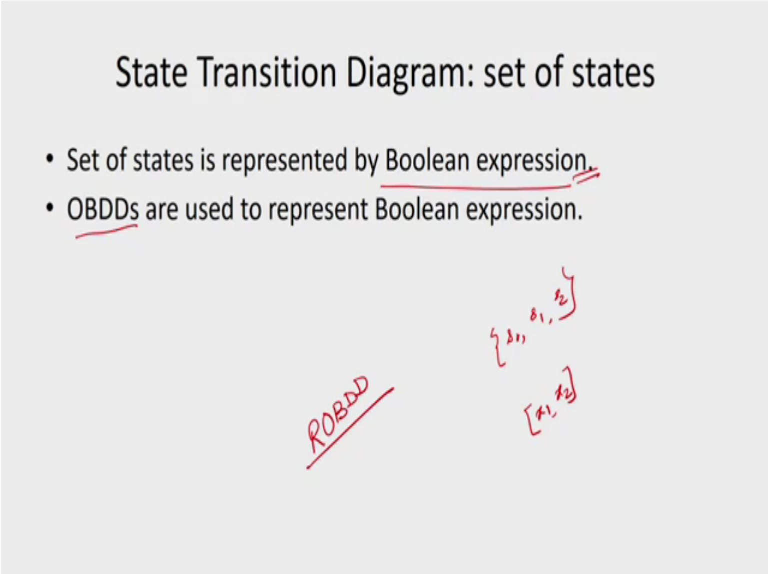


Now, for example, you just see that, we are having this particular same state transition system and a state encoding of these particular states are given is like that; S 0 is equal to x 1 bar x 2 bar, S 1 is equal to x 1 bar x 2, S 2 is equal to x 1 x 2 bar and S 3 is equal to x 1 x 2, you note these things.

Now, if I want to have this particular subset say S 0 S 1, that means I have to represent this particular state S 0 and we have to represent this particular state S 1. So, either of these 2 states or both will be there. So, to represent this particular 2 state I can use this particular Boolean expression, it will say that it is either this is true or this is true. That means, it is going to say that representing this particular state of state with these particular Boolean expression.

Similarly, if we are going to take another set state, S 0 S 1 and S 2 then, this can be expressed with the help of this particular Boolean expression ok. So now, what we have seen over here; that means, set of states can be represented with the help of Boolean algebra ok. That means, we are going to use the expression of the Boolean algebra to represent the set of states, if we are having a binary encoding of the states and we are using or we are taking help of the atomic proposition to have a binary encoding of those particular steps.

(Refer Slide Time: 08:03)

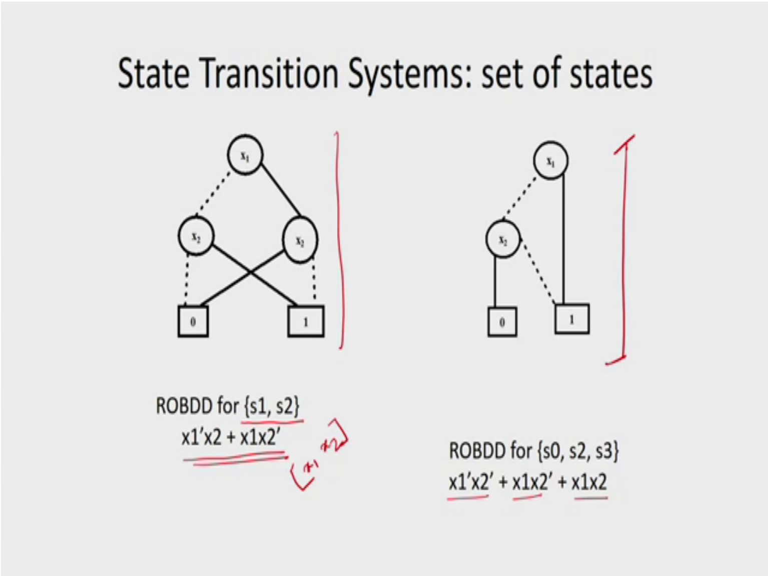


Now, what we have seen is that, first of all we have seen that set of state is represented by a Boolean expression; that means, we can use an Boolean expression to represent a state of state. And already we know that we can use OBDD's to represent Boolean expression. So, if I am having a set of states, consider any set of state say S 0, S 1, say S 2 this is a subset of the given state space. So, if I want to represent this particular set of state what I can do, I can use an Boolean expression to represent this particular set of states.

And we know that, there is a BDD representation of any Boolean expression; that means, that Boolean expression can be now represent that with the help of an BDD's or in particular may be OBDD's or the binary decision diagram. That means, we are going to follow a particular variable orderings of the variable maybe, we can consider a variable ordering like that x 1 x 2 to represent this particular BDD's

So, now what we have seen that, set of states or an individual state can be represented with the help of an Boolean expression and all the Boolean expression can be represented with the help of an OBDD's Order Binary Decision Diagram or maybe you can say that to be more specific, we can use ROBDD that means, Reduced Ordered Binary Decision Diagram ok. So, what we have seen that say set of states can be represented or can be constructed to represent those particular set of state a binary decision diagram or ROBDD.

(Refer Slide Time: 10:00)

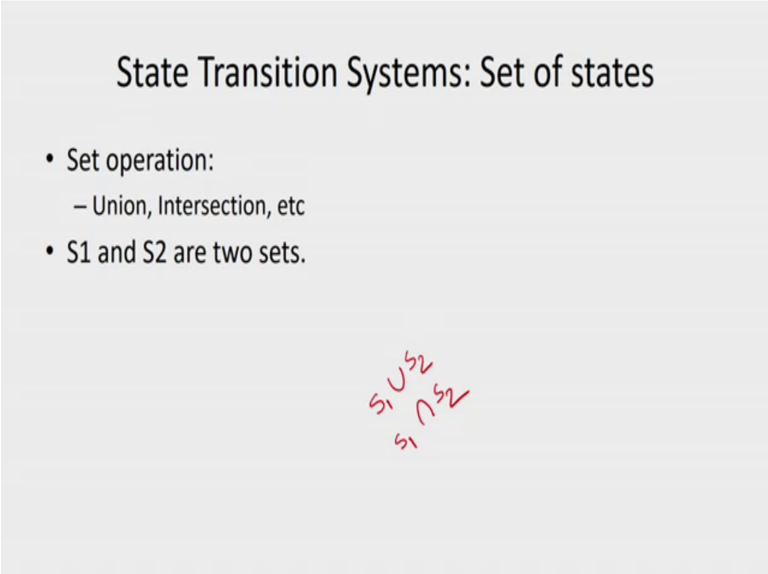


As for example, you just see that, I am considering this particular set of state s 1 and s 2. We know the Boolean expression this for this particular set of state, this is your x 1 bar x 2 and x 1 x 2 bar. Now for this particular Boolean expression we can construct an BDD and here we are considering the BDD's as your BDD and we are consisting that variable ordering as your x 1 x 2.

Now, we know how to constructed the BDD and we know how to reduced it and finally, we are going to get the reduce ROBDD for this particular Boolean expression. And these ROBDD's are going to represent the set of states called s 1 and s 2. This is another example for the set of states s 0 s 2 and s 3 and we know that s 0 is x 1 bar x 2 bar, s 2 is your x 1 x 2 bar and s 3 is x 1 x 3. So, this is the Boolean expression for this particular set of states. After constructing the BDD's with that particular variable ordering x 1 and x 2 and after doing all the reductions then finally, we will boil down to this particular ROBDD representation of this particular set of state.

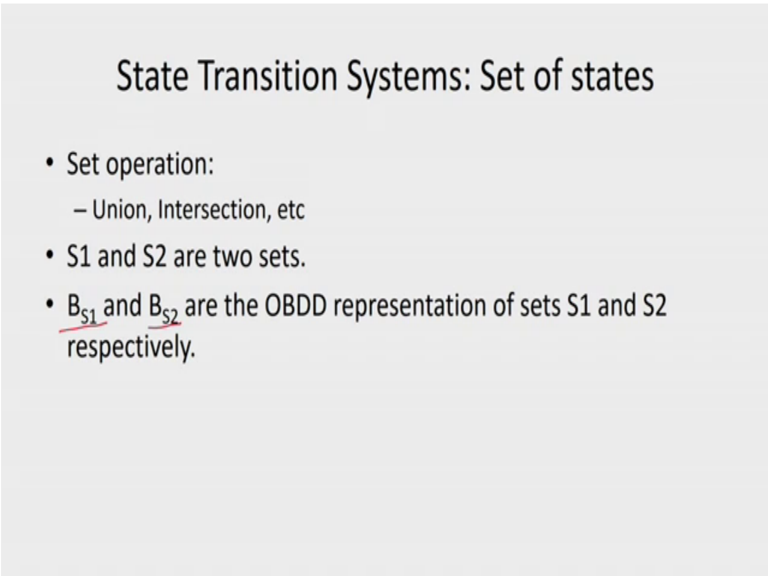
Now, we have seen that we can construct the ROBDD's for any given Boolean function or which basically, means that we can represent any set of states with the help of those particular ROBDD's.

(Refer Slide Time: 11:43)



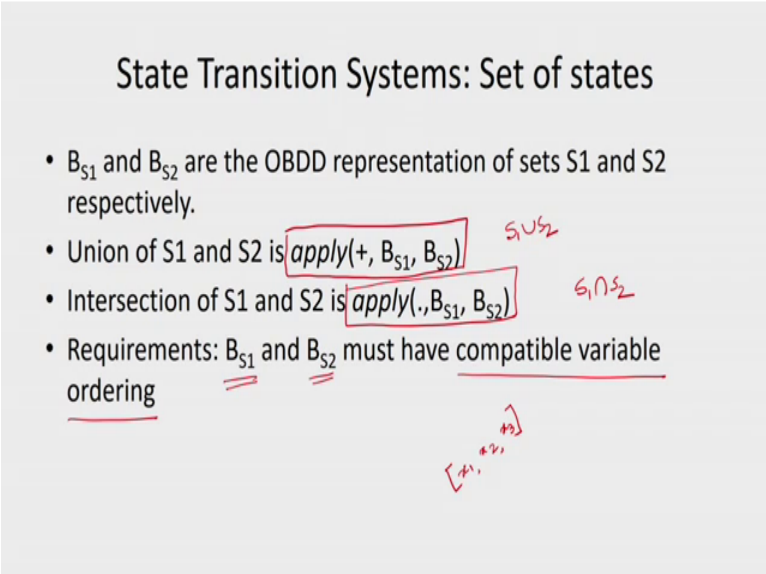
Now, generally if we are working with some set of states then we generally, perform some operation maybe union intersection set difference like that. So, if I am having S 1 and S 2 are 2 given state then, what will happen generally we can perform S 1 union S 2~~.~~ , ~~We~~ we can perform the operation S 1 intersection S 2 like that. So, whether such type of operation can be perform with the help of BDD's or not because, we are representing the set of states with the help of ROBDD's, now whether can we perform those particular set theory scale operation on BDD's or not.

(Refer Slide Time: 12:28)



It is possible, now how we are going to do? We are going to first construct the ROBDD's for this particular 2 states say B S1 is going to represent the set of states S 1 and B S 2 is going is the ROBDD representation of the set S 2 ok. Now, we are in last class we have discussed some algorithm, we have discussed algorithm apply also now, this apply algorithms is going to help us to find out the union or intersection of these 2 states.

(Refer Slide Time: 13:03)

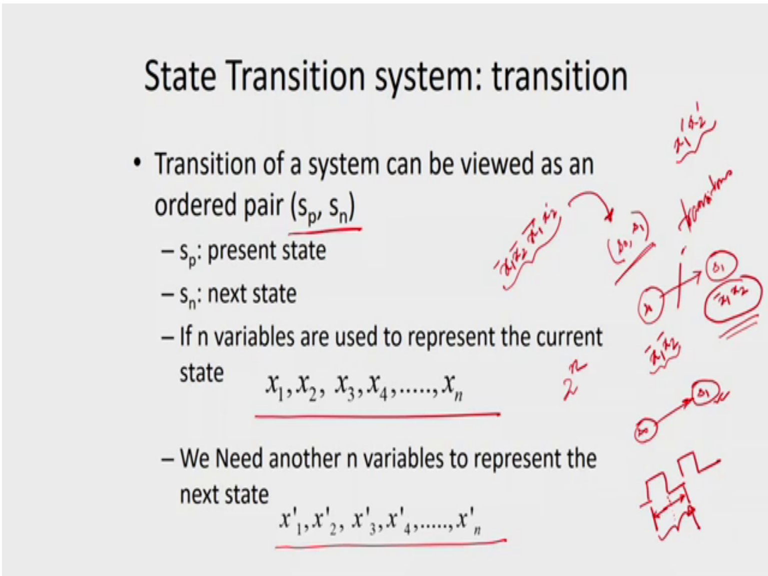


Now, what is the scenario we are going to do? So, for union we are going to use this particular method apply and this is the scenario apply plus B S 1 and B S 2; that means, BDD representation of the set S 1 and B S 2 is the BDD representation of set S 2. Similarly for intersection we are going to use this particular operation apply dot B S 1 and B S 2; that means, we are going to perform the dot operation between these 2 BDD's. And whatever resultant BDD we are going to get the resultant BDD is going to represent a BDD's for S 1 intersection S 2 and this will give the BDD representation of S 1 union S 2.

Now, whatever BDD we are getting we are getting an ordered BDD for these 2 operation resultant BDD's. It may not be a reduced one, already I have mentioned it. So, get the reduced OBDD what we have to do or what we can do, you will apply now reduce algorithm to get the ROBDD. Now, when we are going to perform this apply operation then, we know that both the BDD’s must have compatible variable ordering; that means, they must have the same variable ordering.

So, if we are using say 3 variables then, what we can say that if x 1 x 2 x 3 is a variable ordering or a for a BDD B S 1 then, we must have the same variable ordering for the BDD B S 2. Then only we can use this particular apply method~~. So~~so, this is a basic requirement that both the BDD’s must have compatible variable ordering.

(Refer Slide Time: 15:04)



Now, how we are going to represent the transition system or how we can viewed those particular transition system. So, in transition system what will happen, we are having a transition from one state to the other state. Now, first we have seen how to represent the states or if you are going to consider a set of state, we have seen how we are going to consider those particular set of states and how we are going to represent those particular set of state with the help of BDD’s.

Now, in transition system we are having those particular states say S 0, S 1 and we are having a transition from S 1 to S 2. Now, if I am going to represent the state transition system then, we must have some mechanism to represent those particular transitions ok. How we are going to represent this particular transition ok? So, for that you just see this is a state x 1 x 2 bar and say this is my state x 1 bar x 2 ok. So, in this particular case, it is this transition, say this is state S 0 and this is state S 1. Now, in this transition it involves that I am going from state S 0 to S 1 ok.

Now, in that particular case say we are in a transition, I am going from this particular state S 0 to S 1 ok. Now, in that particular case we know that to have this particular transition, we need some time and generally all the time shall be captured by a clock, that clock may be synchronous or asynchronous; whatever it may be I can say that this is my clock.

So, if my transition start at this particular appears then, we must get the affect over here or maybe during this particular entire clock period. So, during transition, now I am going from S 1 to S 2 and in next clock pulse what it is our scenario basically, my system in this particular state S 1~~;~~. ~~that~~ That means, my system will be represented by this particular S x 1 bar x 2 from this particular timing instant after completion of the current clock pulse.

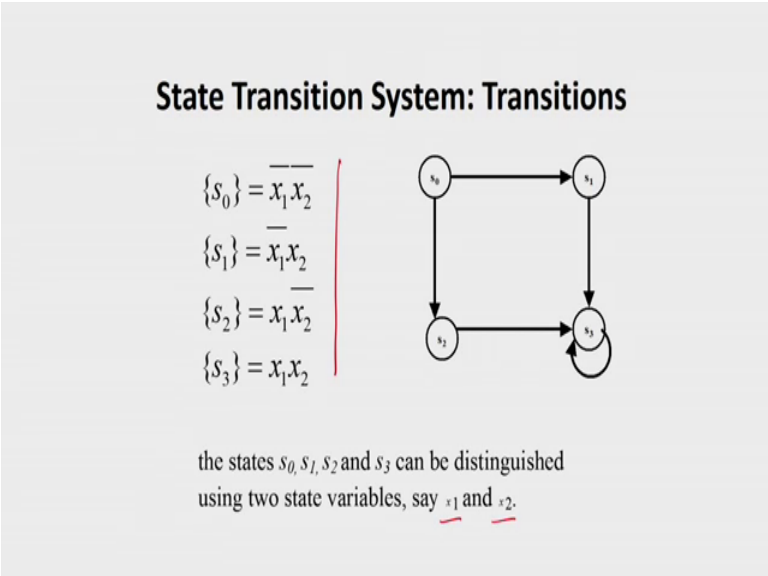
Now, during this scenario I am going to have this particular transition, now how we are going to capture this particular information. So, for that we are going to have an ordered pair called present state and next step combination. So, this present state and next combination is going to give me this particular transition; that means, this is your S 0 S 1, this ordered pair is going to indicate this particular transition system. And S p says that this is the present step and S n is said that this is the next step.

Now, I know that, my system is in this particular present state and I know the state configuration it is your x 1 bar, x 2 bar. When we are going to the next state to represent this particular order pair, we have to take help of another set of variables which are basically, known as your next state variable. So, the next state variable basically, we are going to indicate by the prime version say x 1 prime x 2 prime. So, to represent this particular transition we are going to use this particular next state variable and with the help of next state variable, we are going to indicate this particular transition. And after this particular clock pulse my state position will be your x 1 bar x 2 ok.

So, this transition now if I am going to take help of this particular state variable this transition may be having the representation is like that, my presence state is x 1 x 2 bar ok. And next state it is going to have your x 1 prime, x 2 prime and this is your x 1 prime bar because, this is the thing. So, after this particular 1 next clock pulse arrives; that means, my state variable here x 1 bar x 2.

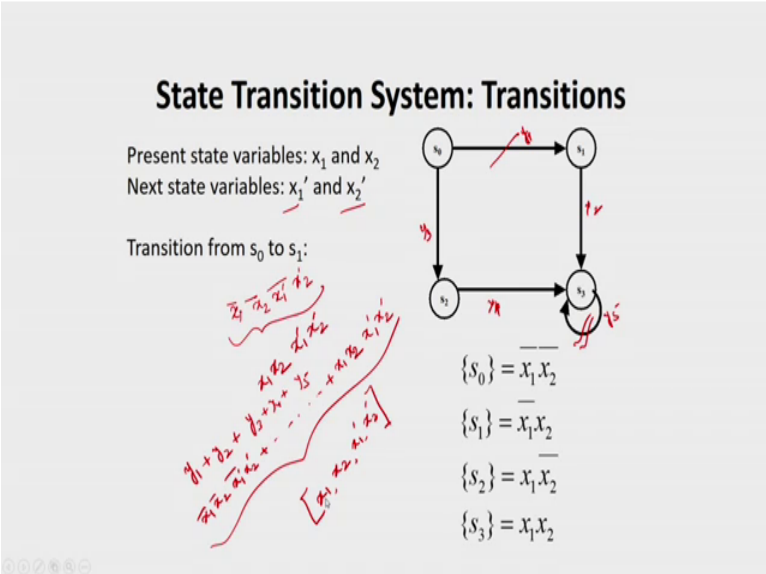
So, with the help of this particular expression, we are going to represent this particular transition from S 0 to S 1. So, that is why we are saying that if the state transition system is represented with the help of those particular variable x 1 to x n; that means, if I am using n variables, that means, total state space can have 2 to the power n state. And to represent those particular transitions, we are going to take help of another set of variables and we indicate with the help of their prime version. So, x 1 prime to x n prime is going have the next state variables ok. So, with this we are going to construct the or represent the each and every transition ok.

(Refer Slide Time: 20:32)



So, this is the state transition system we know this is the representation of 4 states and we are using these 2 state variable.

(Refer Slide Time: 20:42)



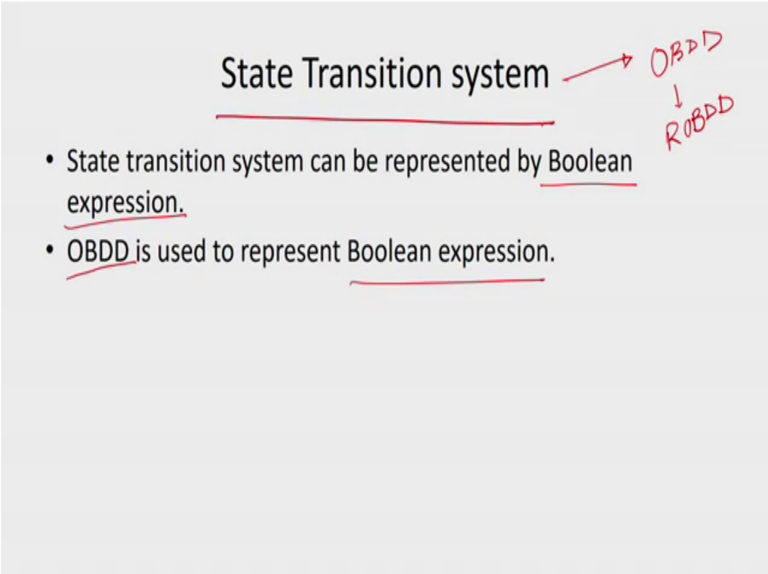
Now, how I am going to represent this particular transition systems. Already I have mentioned that we are going if we having the present state variable x 1 and x 2 then, we are going to take help of a next state variable x 1 prime and x 2 prime. To represent the transition from S 0 to S 1 and we know that S 1 is x 1 bar x 2 bar, so this transition is basically represented by x 1 bar x 2 bar. x 1 prime bar x 2 prime So, this is the transition from S 0 to S 1 and this transition will be represented by this particular Boolean expression.

Now, if I want to represent this particular transition where, it is saying that there is a transition from state S 3 to S 3. So, S 3 is represented by x 1 x 2, now we are having a transition from S 3 to S 3 itself; that means, it is your x 1 prime, x 2 prime these are the next state variable ok. So, this is the way we are going to represent it is an every transition ok. And when I am going to represent the entire state transition system this is nothing, but the collection of those particular transition; that means, what I can say that, if this is my transition say y 1, y 2, y 3, y 4 and y 5 so, I am having total 5 transition. So, total transition system will be represented by your y 1 plus y 2 plus y 3 plus y 4 plus y 5. So, this is the complete transition system

Now, what is y 1, I am talking about this transition; that means, this is your x 1 bar x 2 bar x 1 prime bar x 2 prime plus y 1 y 2 y 3 and y 5, we have already said this is your x 1 x 2 x 1 prime x 2 prime So, again we are getting an Boolean expression for the state transition system. Now, I think you understand now, why we are discussing all those issues. Since, we are having a Boolean expression to represent our state tension system; that means, that Boolean expression can be converted to n or that BDD and we have to follow a particular variable ordering I can say that I may use the particular variable ordering x 1 x 2 x 1 prime x 2 prime, maybe, this will be my variable ordering.

So, with this particular variable ordering we can construct an OBDD and after that if required we will apply the reduce algorithm to get the ROBDD, reduced ordered binary decision diagram. That means, ROBDD can be used to represent our state transition system.

(Refer Slide Time: 24:08)

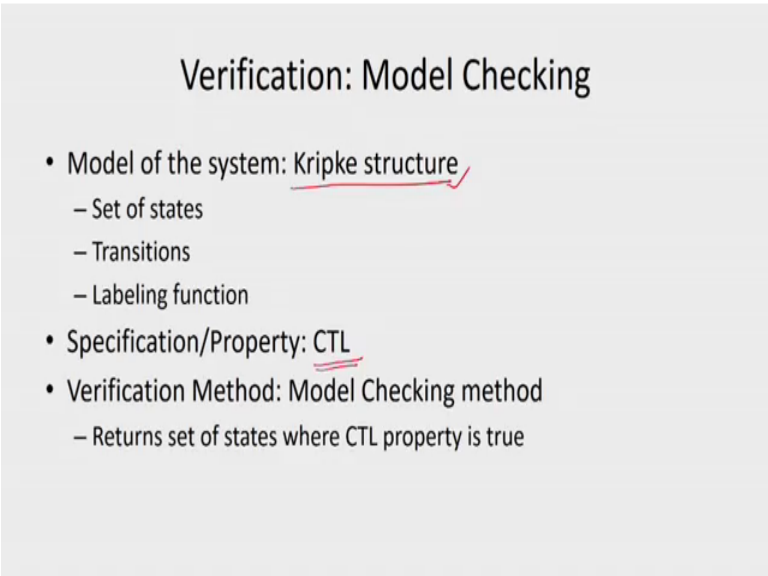


So, this is already I have said that, now if I am having a state transition system, that can be represented with the help of an Boolean expression. And we know that we can use ordered binary decision diagram to represent any Boolean expression. So, what does it means; that means, state transition system can be represented with the help of an OBDD. In general if we follow a particular variable ordering or in particular I can say that, we can have an ROBDD representation of any state transition system ok.

This is the way we can visualize it and now we have come to the conclusion that, any transition system can be represented with the help of an ROBDD and for model checking, we know that we use the term Kripke structure which is nothing. But, the state transition system with a labelling function and with the help of labelling function we label the atomic proposition, which are true in that particular state. And this is nothing, but you can say that that atomic proposition can be used as a state variables and it is going to give the state encoding.

So, what above already we have discussed about this method verification method called model checking, so what is the requirement of the model checking, we are having a model of the system and this model of the system means generally, called I was a Kripke structure. What is a Kripke structure? Just recap it we say that it is finite state machine, but with an additional property that states are labelled with the atomic proposition, which are nothing, but the you can say that these are state encoding variables.

(Refer Slide Time: 26:10)

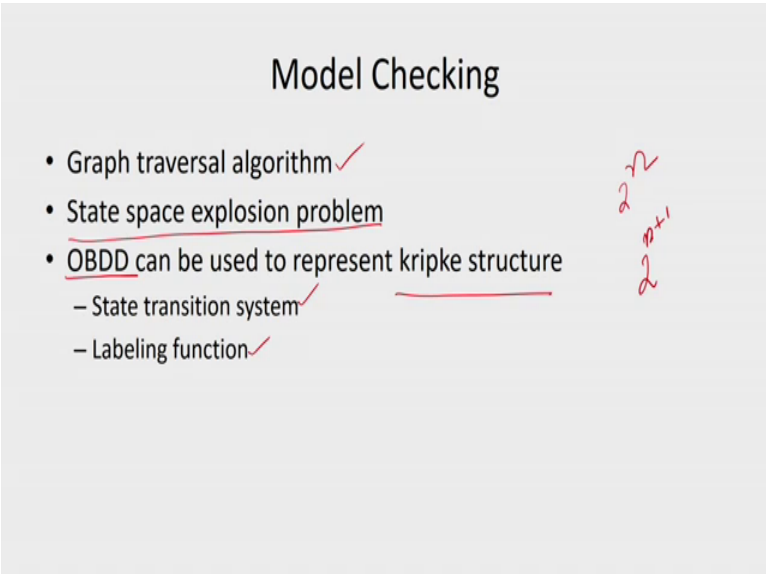


And one more requirement is there that, that transition function whatever transition function we are having that must be complete. That means, for every state we must have a transition to some other state, maybe to that itself also, otherwise we are not going to treat there is a Kripke structure.

Now, specification of the property, we are talking about the CTL model checking. So, we are using CTL, computational free logic to represent the property and specification of the system and we talked about the method, verification method, which is model checking method and what we have discussed about a model checking method, it returns the set of states where is given CTL property is true.

So, you just see we can represent this particular your Kripke structure with the help of an OBDD ok. We are giving a CTL formula, when we are going to verify it, it will return as a set of states where that particular formula is true. Now whatever a set of state we are getting, again we are that can be represented with the help of an ROBDD. So, my transition system or Kripke structure is represented with the help of an ROBDD, after model checking algorithm it will return me an ROBDD and what that ROBDD is going to represent; it is going to represent the set of states where the given CTL formula is true. So that means, we may have a method for model checking, where we are going to use BDD's ok.

(Refer Slide Time: 27:51)



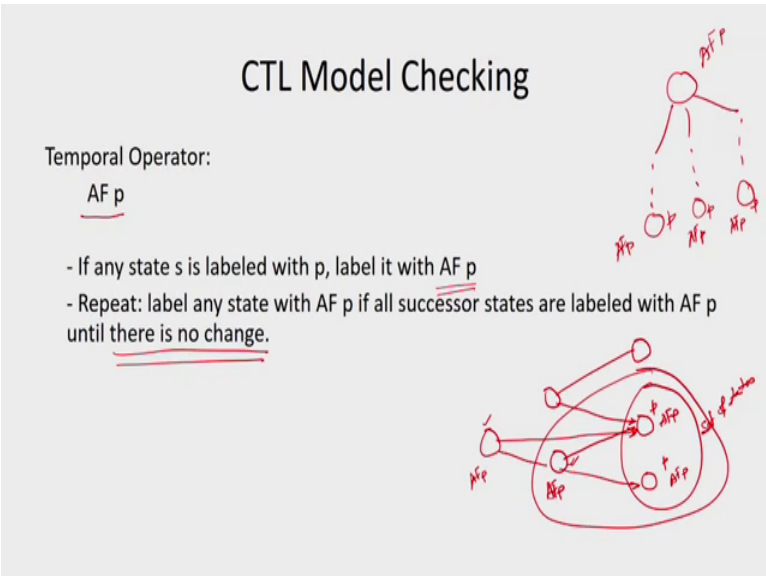
And if we use BDD's for model checking, that is basically known as our symbolic model checking ok, this time is given as your symbolic model checking. So, in general, when we talk about a model checking, what we are getting it is a graph traversal algorithm. Here representing a system with a finite state machine or finite state transition system and our algorithm what each and every state to check whether, a given CTL formula is true and not true or false.

And what is the problem in hand proper to a problem that, we are having with model checking, this is the state space expression problem, already we have talked about it. If we are having and state variable then, we are going to have 2 to the power and different state or may not be disable, but this is the total state space. If we increase the number of variables by 1 because, my design demands it to get the check behaviour now, instead of and we have to use and plus 1 variables then my state space will become 2 to the power n plus 1. So that means, the explosion is exponential in nature.

So, this is the main bottleneck of our model checking algorithm, when we apply the graph traversal algorithm though, we know that the complexity is polynomial time, it is on the number of states and the length of the formula ok. So, to contain this particular state space expression problem, then what we are doing, we are using OBDD's to represent this Kripke structure and accordingly, we will write our algorithm to check the given CTL properties ok.

So, in Kripke structure always said there is a state transition system and along with that we are having a labelling function. And this labelling function can be now treated as a state encoding of the states of the transition system So, this is already I have said symbolic model checking, it is similar to our model checking method, but to represent the state space, we are using BDD's and after that we will apply our algorithm to check a given property ok.

(Refer Slide Time: 30:11)



Now, just have a recap about that CTL model checking, we are going to consider that particular operator say AF p. We know that, what is AF p in all part in future p is true. So, if you are in a current state ok, wherever you go at least in all execution traces somewhere in future you must get a state where, the CTL formula p is true ok. So, in that particular case we said the CTL formula AF p is true over here. This is the meaning of this particular temporal operator CTL operators AF p.

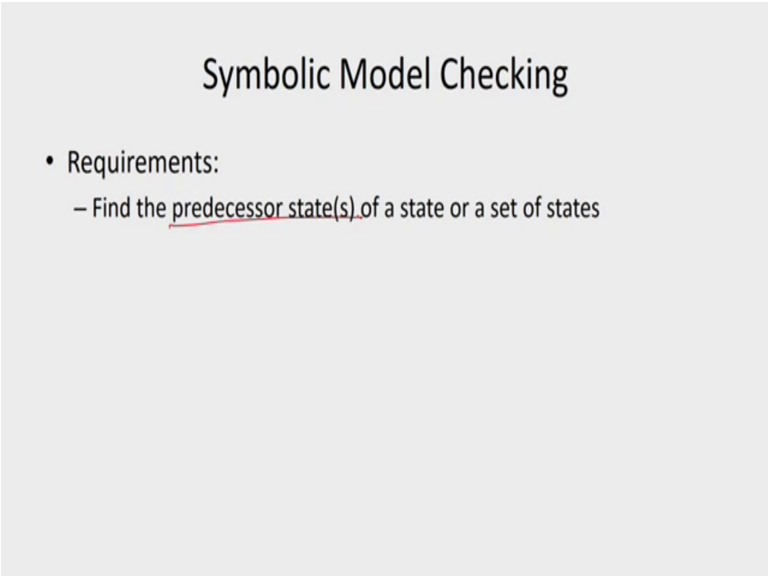
Now, we have already discussed about the algorithm or to check for this particular property. Now what are the step, it says that if any state s is labelled with p then label it with AF p ok. So, if any state if it is labeled with p then what will happen, labeled the state with AFD p because, this is according to our semantics that we have defined because, my semantics says that special includes are present also. Then after it will repeat this process label any state with AF p, if all successor states are labeled with AF p until there is no sense. That means, you just see what it is saying that if p is true over here then, I m going to say AF p's true over here. If p is true over here then I am going to say that AF p also true over here. Now if, I am in some other states and I am having such type of transition then, what we are going to say, we are going to repeat it label any state with AF p if all successor states are labeled with AF p.

Now, if you consider this particular state it is having 2 successor state they are labeled with AF p. So, we are going to label this one also AF p. But this cannot be labeled with AF p because, it is having another transition, another successor which does not labeled with AF p so, here cutting this thing. Now similarly, if I am having some more state than both the successor with labeled with AF p so, I am going to labeled it with AF p. So, this is the method that we have to check for this particular temporal operator CTL operator AF p.

Now, just see the nature what we are doing it, it is nothing, but somehow to find out the predecessor state of a set of states. So, initially it is marked with this particular set of state ok. Now, we are going to look for all the predecessor of the set of state and we are going to pick up those particular states which is satisfying this given property AF p; that means, all successors should have labeled with AF p, so that is why we are collecting it now we are getting this particular set of states ok.

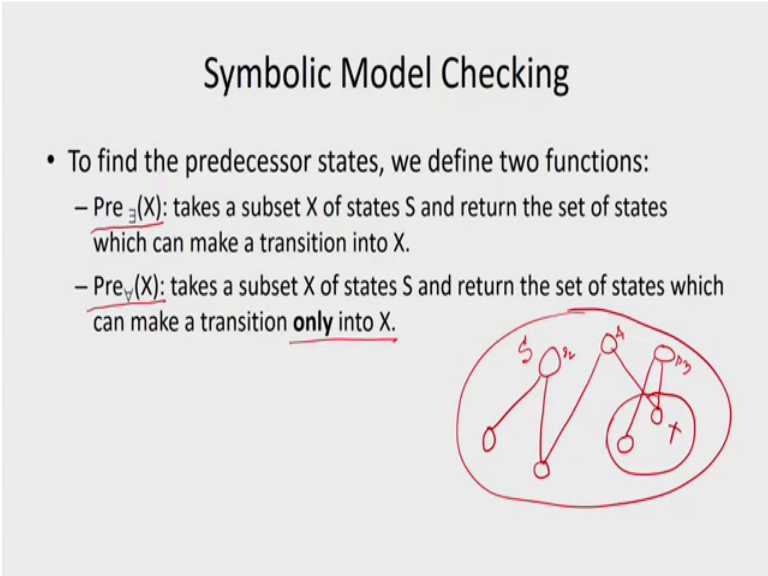
Now, after that again we are looking for the predecessor for those particular set of state and we are getting it. And if I cannot get any other predecessor said which is satisfying our requirement then, we will stop it because we said that until there is no sense, this there is no sense is talking about basically, this collect set of states where we are collecting new states, if we cannot include any more new state then we will time at all algorithm. So, this is the way we are checking AF p. Now, what is the clocks over here, what we have observed. We need some method to find out the predecessor states of a given set of states ok. Now we will see what can be done.

(Refer Slide Time: 34:21)



So, this is the requirement, we have to find a predecessor state or the set of states of a particular state or maybe a set of state. And we know that whether it is a set state or set of state, there can be represented with the help of an BDD's we can have a BDD representation of a set of state So, now so how to find out those particular predecessor step now we have to see this particular scenario.

(Refer Slide Time: 34:52)



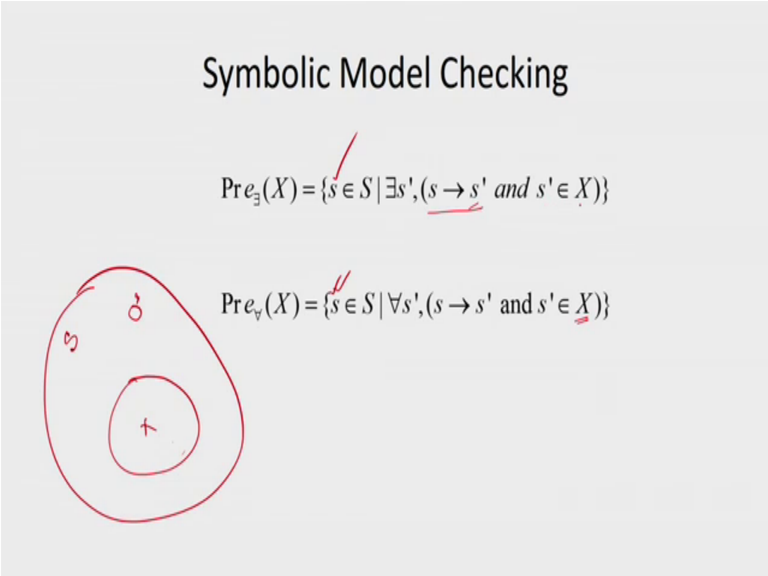
Now, to find a predecessor state here we are going to define two function ok; what are these two function one is known as your pre there exist X, what it says take a subset X of state S and return the set of states which can make a transition into x. That means, if this is my total state space S and I am considering a particular subset X over here. Now it says that take a subset X of state S and return the set of state which can make a transition into X; that means, if I am having a transition something like that, then what will happen in that particular case, this particular state will become a predecessor of this particular state X. But this S 2 will not become a predecessor of this particular state as because, none of the transition is going to this particular set of state X.

Similarly, you having pre for all X, take a subset X, this is a subset X of a given state S and return the set of state which can make a transition only into X so, it is saying that which can have a transition only into X. Now if I consider these things say this is having 2 states, 2 transition and both the transitions are coming to a state which is a member of this particular subset X. Then this particular S 3 will be a member of this particular pre for all X function ok. So, we are talking that pre-function which basically gives me the predecessors of a set of states and there is 2 notion.

One is pre there exist X and pre for all X So, for there exist X at least I should have a transition to the set of state X, for all of X or for pre all and we are going to consider those particular predecessor whose all the transition are going into this particular subset

X ok. So, that means, we must have a method to find out those particular predecessors state there way we are defining it.

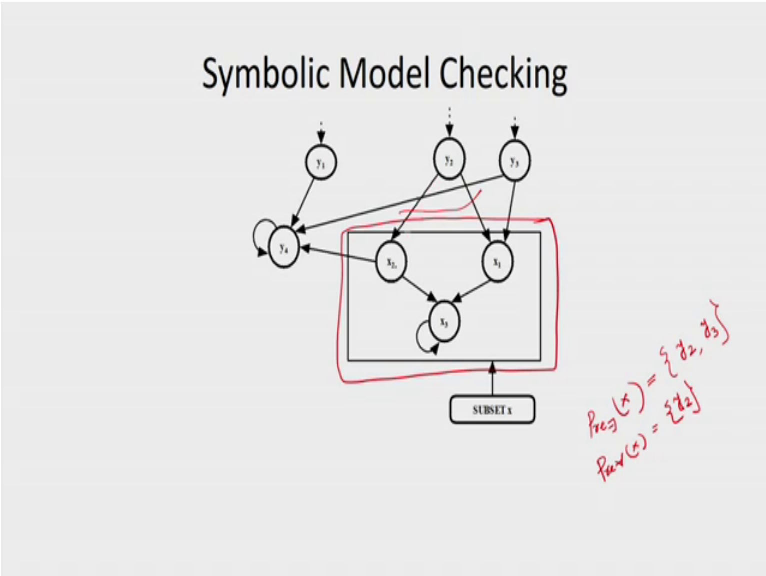
(Refer Slide Time: 37:24)



Now, this is the basic formal definition of pre there exist X already, we are just having the notion of pre there exist X it says that, if I am taking this particular subset X of a given set S then, I am going to collect such type of state s, which is a member of this X such that; it is going to satisfy this particular behaviour. We must have at least one s dash, such that we having a transition from s to s state and s dash belongs to this particular set X ok. So, such type of state we are going to collect and we said that this is the predecessor of those particular set of state X.

Similarly, for pre for all X then, again we are going to collect such type of state s and what are the properties of those particular state; it says that for all s dash we are having the transition from s to s dash and this s dash must be a member of this particular X. So, whatever successor we are having of this particular state s, all must be a member of this particular subset X ok. So, this is the formal way of defining pre there exist X and pre for all X ok. Now we have just defined it now we will see how we can identify those particular set of state.

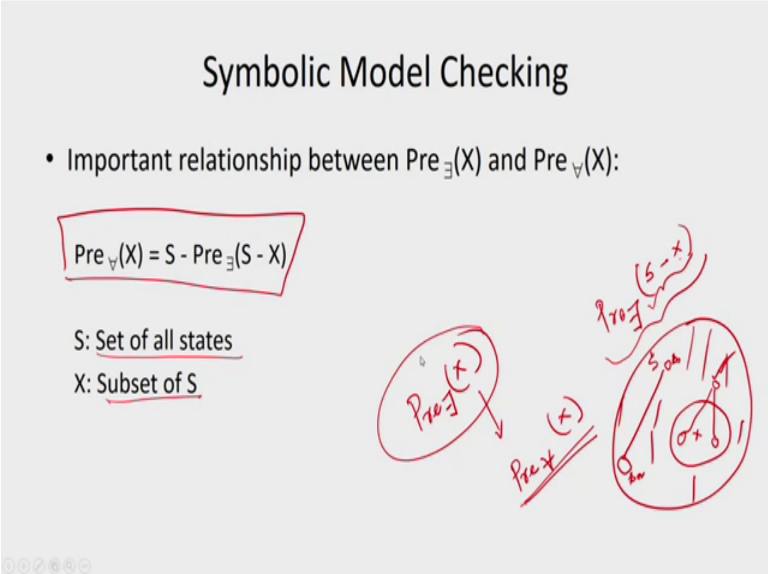
(Refer Slide Time: 38:58)



So, this is some example some figure I try to this thing, so this is a subset x ok, we are considering ok. Now, pre there exist x, then what will happen now, we are going to see this particular scenario and we will find that this whatever state y 1 y 2 y 3 and y 4 we having, which is outside of this particular subset x. Then pre there exist x will be your y 2 and y 3 because for these 2 state some of the transitions are coming over here because, here you are defining like that. We must get such type of transition from s to s dash and s dash must be member of this particular X.

So, in this particular case what we are getting that, pre there exist x will be your state y 1 and y 2. Now if I am going to talk about pre for all x then, what I am going to get, now I think you can visualize it the scenario. If I am having this scenario than it is going to give me state y 2 only because; all the transition from this y 2 is leadings to this particular subset x. But from y 1, one is coming to the subset x, but other one is outside of this particular s. So, pre for all x will going to give me y 2 because we are having such type of states where all transitions are coming into this particular subset S ok.

(Refer Slide Time: 40:48)



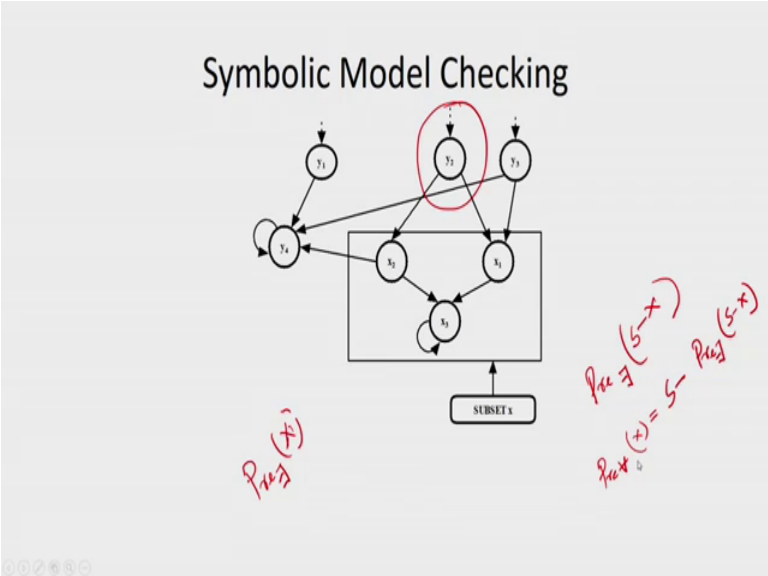
Now, we are having an important relationship between pre there exist X and pre for all X. So, pre for all X can be expressed with the help of this particular equation; that means, it is pre there exist S minus X can have this thing. So, S minus pre there exist S minus X, what is S, it is the set of all states and X is your subset of X ok.

Now, this is an important relationship, why we are saying. So that means, if we are having a method to find out pre there exist X then, we can use this particular method to evaluate pre for all X ok. So, this is the scenario that we are having now. So, this is a scenario now can you visualize it why this particular scenario is true ok.

Here we are talking about pre there exist, pre there exist S sorry, first we are evaluating this things; that means, if this is the set of state S and this is X. Now what is the set we are considering, S minus X ok. That means, if I am going to have a transition state say S 0 and if it is having a transition to say S n that means, S 0 will be a member of this particular pre there exist S minus X because, S minus X is going to give me this region except this particular X ok.

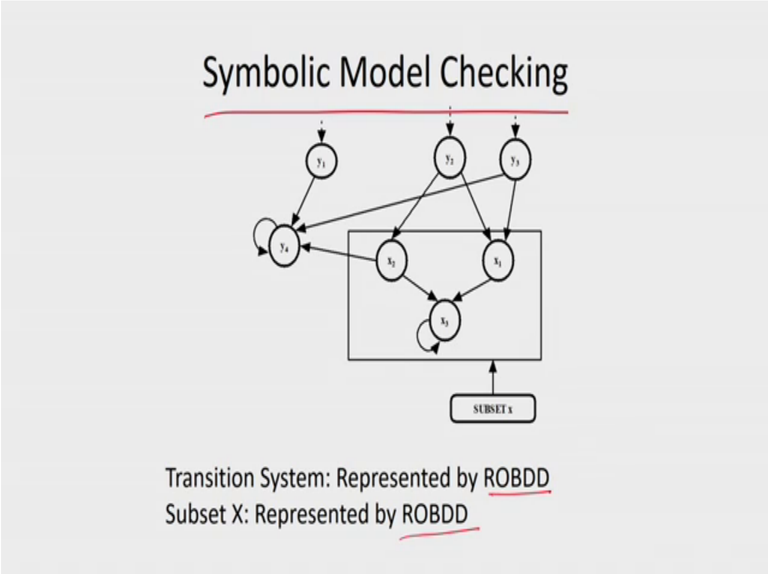
So, what it says that, this is the state which is not having any transition to X ok. So, if we are not getting such type of scenario consider this particular state where I am having 2 transition both are coming to this particular state, then it will not be a member of this particular pre there exist S minus X. So, this is going to have the property that all the transition is going into the subset X. So, that is why S minus pre there exist S, S minus X is going to give me pre for all X ok. So, this relation helps us to find out pre for all of for all X provided we are having a method to find out pre there exist X ok.

(Refer Slide Time: 43:29)



So, this is same model. So, if I am going to talk about say pre there exist S minus X and X is having this subset. By looking into this scenario what we are going to get pre there exist at X minus S, you will find that we are having transition which are going outside of this particular X, all those states will be captured, but this state will not be captured because, all are going inside this particular X. So, pre for all X if I am going to have then, what I am going to get S minus pre there exist S minus X ok. So, basically these state will common topics. So that means, we need a method to evaluate pre there exist X pre there exist X. So, if we are having a method for this one, then we can have a method for other one also.

(Refer Slide Time: 44:39)



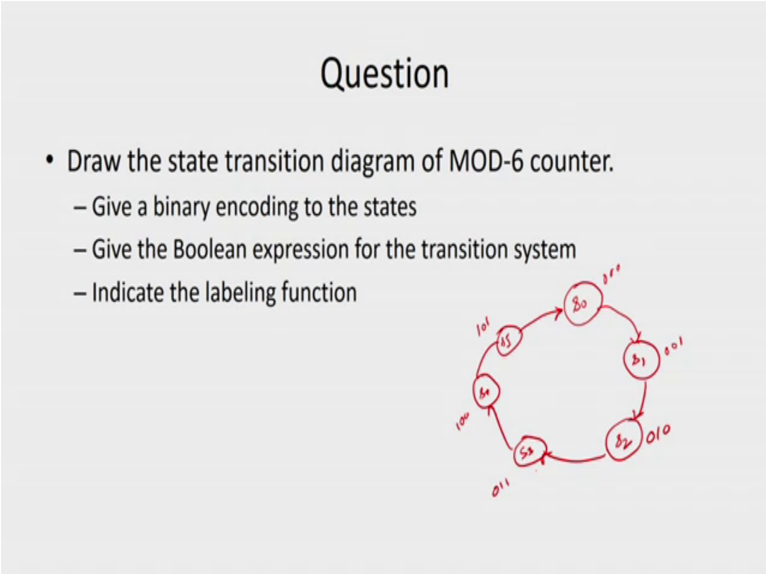
And we have seen that for our model checking algorithm our basic requirement is to find out the predecessor state or by given state of states ok. Now we must have a method for this thing. So, if we are having a method to find out the predecessor states of a given set of state then, what will happen we can implement our model checking algorithm. And when we implement a model checking algorithm with the help of our BDD's then we say this is my symbolic model checking algorithm ok.

So, what will happen now that, transition system can be represented by your ROBDD again subset X can be represented by ROBDD. Now, we should have mechanism or we should have method to find out that particular subset, which had a predecessor of a given set of state. Again if I am having a given set of state, that set of state can be represented with the help of an ROBDD. So, these are the resource, transition system, I am having an ROBDD, given set of state, we are having an ROBDD ok.

So, we need a method which is going to give me the predecessor states of that given set of state; that means, again it will return me a subset, which will be a which have some states and that states can be represented by again an OBDD. So, basically it is going to return me the OBDD's which is going to represent those particular set of states which has despise those particular predecessor property.

Now, we have to see how we are going to implement it. May be today I am going to wind up my lecture over here. In next class we are going to talk about the implementation of this predecessor, function predecessor for there exist X and we will use that particular function to have an symbolic model checking algorithm ok.

(Refer Slide Time: 46:43)



So, consider one scenario, it says that giving a very simple question. Now, we have you know all of you know about the state transition system, now we are saying that BDD's can be used to represent this particular state transition system. Now, how we are going to do it? We are having an state encoding of all the states and state encoding can be done with the help of your state variables and state variables can be treated as an atomic proposition of my system. So, just I am saying that now draw the state transition diagram of MOD-6 counter. So, all of you know about counters, you have designed several counters maybe it is your synchronous counter, asynchronous counter, maybe up counter, down counter, so you know all those things.

Now, I am saying that draw the state transition diagram of a MOD-6 counter, we know what is MOD-6 counter, it counts from 0 to 5 and all of you know about your that state transition. So, if the system is having in say S 0 which is going to count 0 then it will going to state S 1, which will give me the count value 1. So, if it is my count value 0 0 0 then this is your 0 0 1 then, it will go to S 2 count value is 0 1 0 then it will go to S 3 count value is your 0 1 1. Then, my next transition will be your S 4 which is your 1 0 0. Then next state is your S 5, the count value is your 1 0 1 then again it will go back to 0. So, this is a MOD-6 counter which count from 0 to 5

Now, all of you know these things and you have when you are going to design a MOD-6 counter generally all of you start from this particular state transition diagram. Then you are having the state table representation then, you are having the excitation table. Looking the excitation table you find out the inputs for the flip flops and finally, you come up with the design.

Now, since we are talking about a BDD's now what I am saying that, consider a binary encoding of the states because, we need 3 variables already I have put the encoding. Now give the Boolean expression for this particular transition system. Now, you just try to find out the Boolean expression how to represent this particular transition system. Already we have discussed about it, we are having a state variable to represent those particular transition. We need an another set of variable, which is tutorage your next state variable then, we can construct a Boolean expression to represent the entire state space ok. So, this is the scenario we know.

Now, once you get the Boolean expression for this particular state transition system of MOD-6 counter, now you try to construct the BDD's ok, just try it let us see how what is the BDD that you are going to get. Now, just here I am talking about a indicate the labeling function because, labeling function is going to give me or can be simply find out from the state encoding that we are using to represent difference state

So, you just go through it, maybe this is a very simple example I am giving. You may try for some of the transition system also where, you can capture the behaviour of the transition with the help of Boolean expression ok. You just attempt try to do 2-3 more examples so, that you can visualize it is really with Boolean expression we can represent the state transition system ok.

With this I will wind up today, in next class already I have mentioned that we are going to now discuss about how to find out the predecessor state or how to implement of the pre there exist X, where X is a given subset and we will use this particular method to implement the model checking algorithm. If the inputs will be used as my, your BDD; that means, transition system is represented that with the help of BDD's then we say this is my symbolic model signal. So, that is all.

Thank you all, good bye