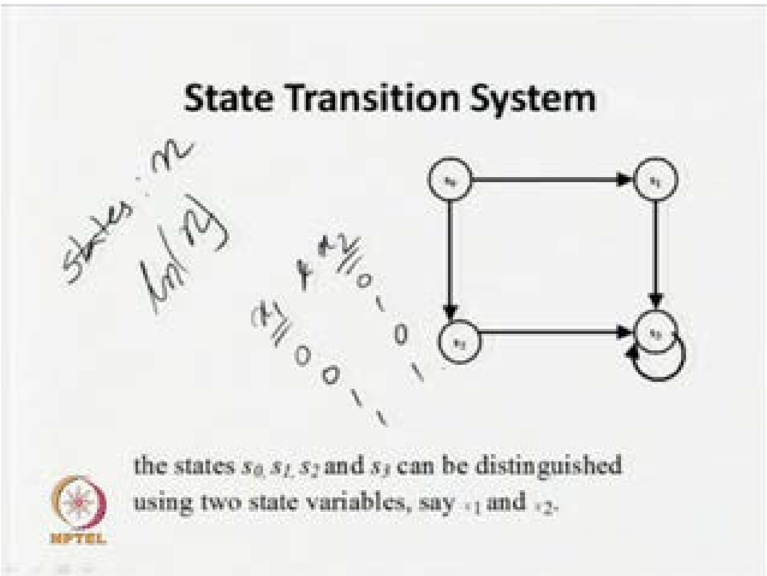
Design Verification and Test of Digital VLSI Designs
Prof. Dr. Santosh Biswas
Prof. Dr. Jatindra Kumar Deka
Indian institute of Technology, Guwahati

Module - 6
Binary Decision Diagram
Lecture - 4
Ordered Binary Decision Diagram for State Transition Systems

We are discussing a data structure call BDD or in particular ROBDD reduced are the binary decision diagram, by which we can represents our any Boolean function. Also we have seen some algorithm by which, we can manipulate the BDDs; like reduce, apply, restrict, exists like that. And on the other hand, we have seen that any digital circuit can be represented by Boolean expression, and that Boolean expression can be represent by ROBDD. And we have seen that with a particular variable ordering the BDD representation of any function is unique. Today we will see, how we are going to represent the sequential circuit; so how you are going to use ordered binary decision diagram for sequential circuit. When you look into the sequential circuit, then will find that or we can see that, it can be represented with the help of State Transition diagram.

(Refer Slide Time: 01:23)



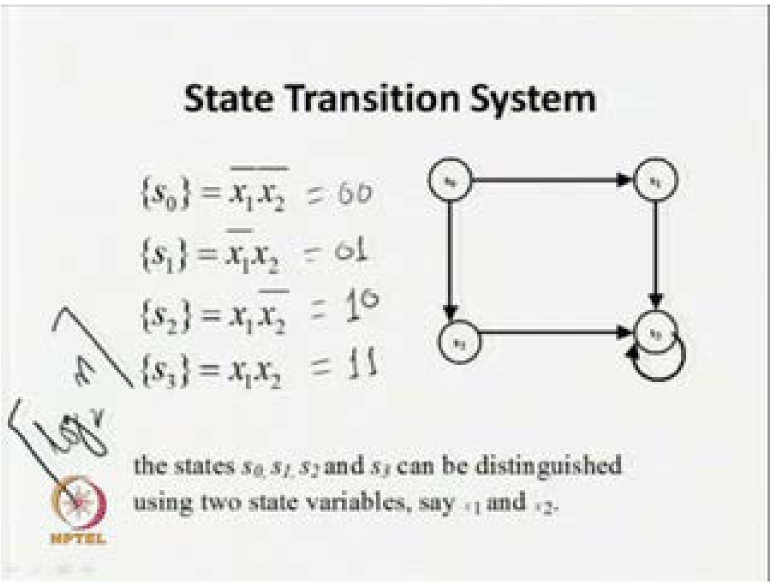
So basically, it says that my system is in a particular state, depending upon the present behavior; and the input sequence it will make transition to the next step. Now, we have to represent those particular transitions, when we are going to talk about the sequential circuit; and basically sequential circuit will be represented by the state transitions system or state transition diagram. Now just see this particular case, that I am having a state transition system. It is having a four state; S 0, S 1, S 2 and S 3. And we are having those particular transition, I am having a transition from S 0 to S 1; when we are in that transition S 1, then you can make transition to S 3. If I am in the S 3, then it will remain in S 3; on the other hand from S 0, we can go to S 2 and from S 2 I can go for S 3.

So this is the state transition diagram and we know that. If we are having a state transition diagram, then we can synthesize the sequential circuit for that particular state transition diagram. So in this lecture we are going to see, how we are going to represent those particular state transition system with Boolean function, and eventually we are going to represent those particular Boolean function with the help of OBDDs, Order Binary Decision Diagram. To represent this particular State Transition System, it has having four different states. We represent this particular four states, we need at list two state variable; with the help of this particular state variable, we can represent this state of this particular state transition system.

So if in generally, if a particular state transition system is having some number of states is equal to n. Then the minimum number of state variable will required to represent this particular state transition diagram, is l n n; that means log to the best 2 n. so this is the minimum required number of variable, that we need to represent this particular system. But, we may take more state variable also, so depending on that the state including will be different.

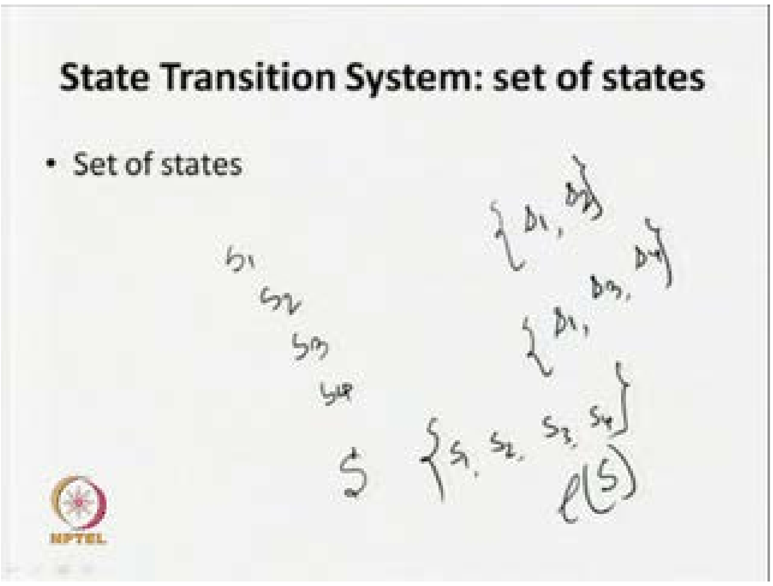
Now seen I am going to say that, we reach at list log two best, log n 2 best 2 number of variables to represent n states, since we are having four states over here. So at list we need two state variables, and we are representing this particular state variable as say x 1and x 2. With this particular state variable, we can represent this particular four state; because we are having a four different combination 0 0 0 1, 1 0 and 1 1. With this particular four combination, unique combination we can represent this particular of four states.

(Refer Slide Time: 04:18)



So this is basically, now we are going to say that, with the help of this particular state variable. We can say that the state S 0 will represented by x 1 bar, x 2 bar; that means you can say that, when both are 0, 0; we are going to represent the state S 0. So S 1 will be x 1 bar, x two; that means we can say this is your 0 1 state encoding. S 3 is x 1, x 2 bar; that means, state encoding is your 1 0. And S 3 is your x 1, x 2; that means, this is the state variables is 1 1. Then we can say that, these are the state for this particular state S 3. So we need at list two state variables to represent those particular four states.

(Refer Slide Time: 05:33)

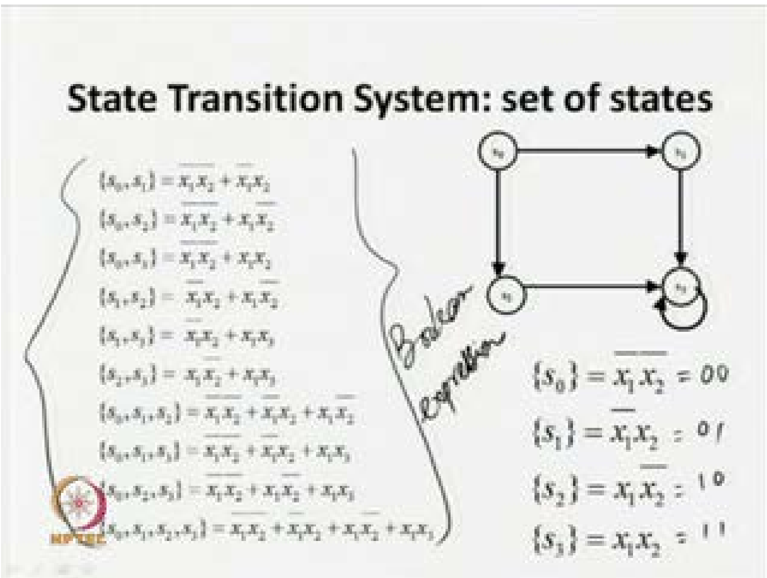


And already I have mentioned that in general, if I am having n different states then we need log n to the best 2 number of variables. Sealing of that particular thing that means, log to the best 2 n. And I am going to take the sealing of that particular values and this is the number of variable required, minimum number of variable that we need to represent the state n, different state unique.

So for this particular state transition diagram, I am having a four different states and this are the state at coding of this particular four states. Now when we are going to talk about your state transition system, we are having four different states say: S 1, S 2, S 3, S 4. Sometimes we are interested for the set of states, I am going say that the set of state is your say S 1, S 2.

Sometime I mean, interested to look for the set of state S 1, S 3, S 4; that means, we can see that these are the power set of this particular set of states. I am having a set of state as your S 1, S 2, S 3, S 4; this is the set of states, I can say that this is your S. Then what are the set of states? What are the different sub set we are going to get, this is nothing but a power set of this particular S; that means, we are going to get 16 different combinations. Now, how to represent those particular set of state? We are going to see this particular scenario.

(Refer Slide Time: 06:33)



Now come back to this particular previous state transition diagram, I am having state as S 0, S 1, S 2, S 3. And these are the binary state encoding for this particular four states S

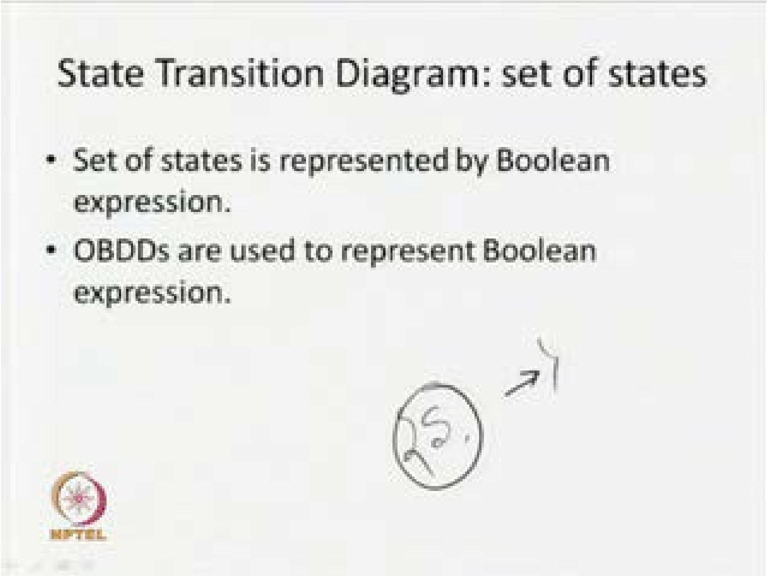
0, S 1, S 2, S 3; that means this is a x 1 bar, x two bar already I have mention this is: 0 0, 0 1, 1 0 and 1 1. Now when we are going to look for the set of states I can talk about at S 0, S 1 is set of state; S 0, S 2 is a set of states; like that S 0, S 2, S 3 is a set of states.

Now I can represent all those particular set of states with the help of Boolean expression. Because S 0 is represented by x 1 bar, x two bar; and S 1 is represented by x 1 bar, x two. So this is x 1 bar, x 2 bar plus x 1 bar, x two; similarly, S 0 S 2 is your say x 1 bar, x 2 bar plus x 1, x two bar. Now if you see this things, is these are the representation of the set of states; it is represent two state S 0, S 1. Similarly, this is this expression is representing these particular three states: S 0, S 1 and S 2. And if you look this particular right hand side you will find that, these are nothing but Boolean expression.

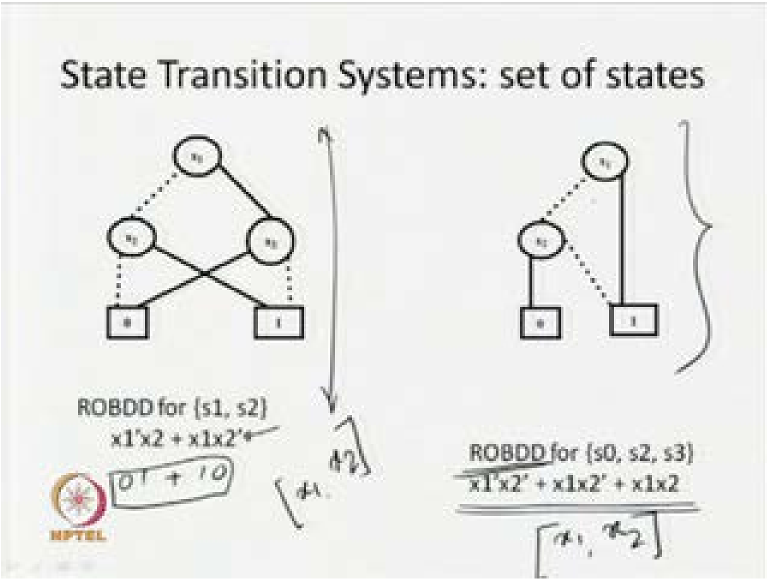
Once you have this particular Boolean expression, then what we can do? We can construct BDD, for those particular Boolean expressions. Now you just see that, the set of state can be a represented with the help of BDDs; or in particular you can say that, ROBDDs, Reduce Ordered Binary Decision Diagram. And the representation we are going to get an unique representation, if you are strict to a particular variable ordering. So now, if you having some states, if you conceder sub set of those particular set of states; then what you can do? We can represent those particular sub set or set of states, with the help of a Boolean expression; and we can convert this particular Boolean expression or we can represent this particular Boolean expression with the help of ROBDDs, that means, ROBDDs can be use to represent set of states.

Now in natural, what we can say that? If we are going to look into a state transition system; state transition system is nothing but the set of state and one transaction relation. So basically we can say that, this is basically state transition diagram; this are two basic components set of states and the transition relation. Now we have seen that the set of states can be represent that with the help of Boolean expression and once you are having the Boolean expression, that Boolean expression can be represented with the help of OBDDs; or we can see that ROBDDs if you strict of particular variable ordering.

(Refer Slide Time: 08:50)



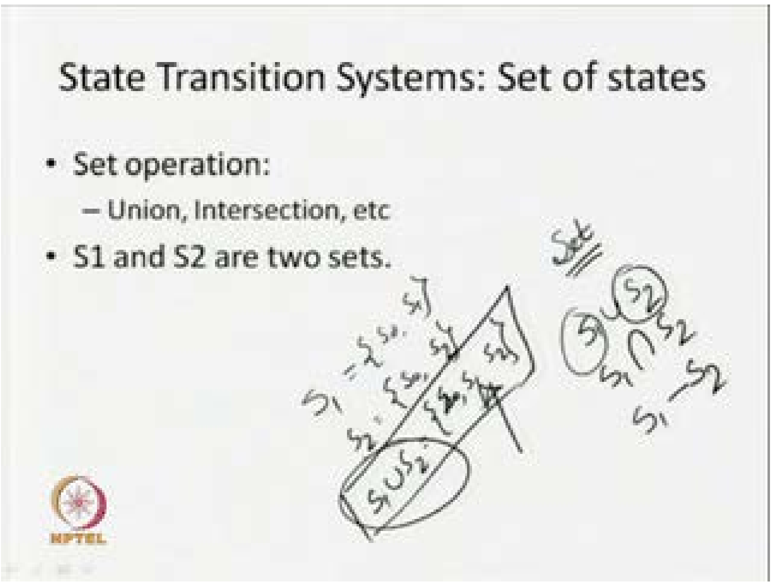
(Refer Slide Time: 09:37)



Next you are going to see how you are going to represent this particular, this thing. So in this particular case a simple example I am showing over here say, this is the set of state I am going to take S 1, S 2. So this is up Boolean expression for this particular sub set S 1 S 2 x 1 prim x two plus x 1 x two this. So this is basically S 1 is your 0 1 and S 2 is your 1 0. Now once I have this particular Boolean expression, we can construct the ROBDDs for this particular Boolean expression and we are going to get this is as our Boolean ROBDDs for this particular Boolean expression. That means, these ROBDDs are going to represent the set of state S 1 S 2, and what is the variable ordering over here? I can say that the variable ordering is your x 1, x 2; this is the variable ordering we are using to construct this particular BDDs. And if you look into this particular BDDs, you will find that no more reduction on can be apply to this particular ROBDDs; so ultimately we say that this is the ROBDD representation for S 1 and S 2 states.

Similarly this is the envelop ROBDDs. This ROBDD represent the subset S 0, S 2, S 3; so which similar encoding that we are having. So we are going to look for the subset which consists of state S 0, S 2 and S 3. Since we know the binary encoding, we can write down the Boolean expression this particular subset 1. We have this particular Boolean expression allows you can construct the ROBDDs for this particular Boolean expression and eventually if you try it. If you try to construct it, then you will going to get this particular ROBDDs for this particular Boolean expression and the variable ordering again we are choosing for this particular order BDDs your x 1, x 2. So with this particular variable ordering we are getting these particular BDDs and this is basically ROBDDs for this particular subset, because we cannot apply any more reduction rules to this particular BDD, so we are getting the ROBDDs.

(Refer Slide Time: 12:35)

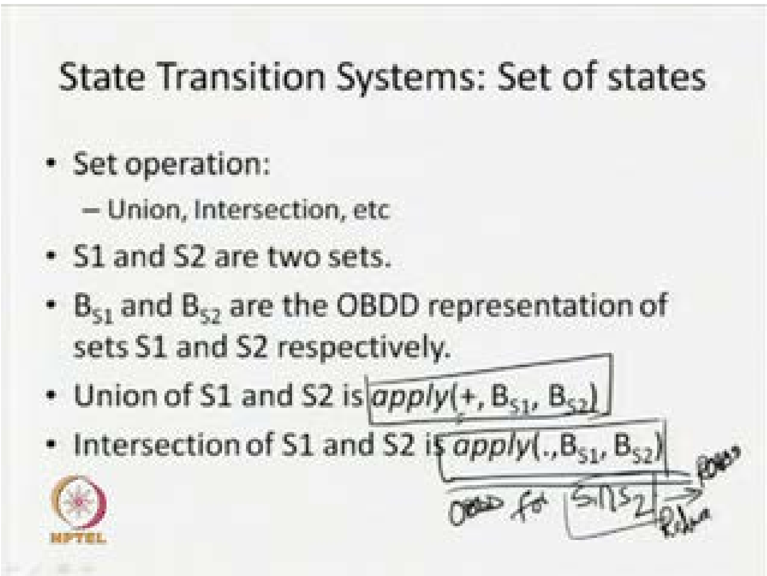


So like that if we are having a state transition system, we are having several states; all those particular states can be represented with the help of ROBDDs, because those particular states can be represented with the help of Boolean expression. Secondly if we take any subset of those particular states, then what will happen? Again we can write the

Boolean expression though for those particular Boolean subsets and this Boolean expression can be represented with the help of your ROBDDs; so we can now say that, the set of states can be represented with the help of ROBDDs.

Now when we are going to have a set, we are talking about the set, when we are having this particular set, then generally you can apply some state theoretic operation on set. Say if I am going to consider two set S 1 and S 2; similarly, you can construct S 1 union S 2, S 1 intersection S 2 like that, set the parenthesis S 1 minus S 2 like that. If we are working with the set, then we can perform some set operation on those particular states. Now these particular states S 1 and S 2, we are representing with the help of ROBDDs; now say if I want to take the union of these two states, now can you work with those particular ROBDD representations. If we want to take the intersection, can we use those particular ROBDDs to get the intersection of these two states; or whether first that to construct the, first that to calculate the union, then construct the ROBDDs. This is also you can go; so we now S 1, you say it is your say S 0, S 1, we know S 2; then we know say this is S 1, S 2.

(Refer Slide Time: 14:22)



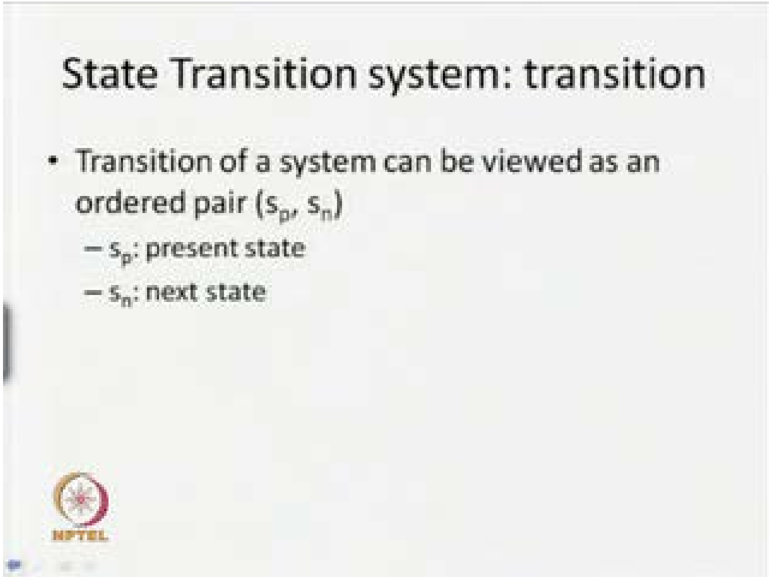
So if I am going to say that a S 1 union S 2, then basically I can say that this is your S 0, S 1, S 2. So first I can calculate this particular union, then I can construct the ROBDDs for this particular union. But, if I am already having the ROBDDs for these two particular sets, whether can I use some algorithm or use some operation; so that I can directly going to get S 1 union S 2. It is possible.

So how we are going to do it? Say S 1 and S 2 are two sets, now we are going to say that B s 1 and B underscore s 2 are the OBDD representation of set S 1 and S 2 respectively. So, B s 1 is the ROBDD representation of S 1 and B s 2 is the ROBDD representation of S 2. Now if we are going to take the union of these S 1 and S 2, then we can construct with the help of these operation. Already we know, the apply algorithm, apply with which some operator can be applied with the help of this apply operation; so we are going to apply the operator is your class B s 1 and B s 2. Now this operation after constructing this, after applying this particular operator what will happen? We are going to get an OBDDs; and that OBDDs are going to give me the set of state basically and these going to said a union of S 1 and S 2. Because it is going to take the common position of form B s 1 and B s 2, you can construct this.

Similarly if I am going to perform the intersection of S 1 and S 2, then we can apply these particular operator apply dot B s 1 and B s 2. So by application of this thing, we are going to get the OBDDs for S 1 intersection S 2. So you just see that if I am in the BDD representation of two sets, then I can use this particular apply operation on that to get the union and intersection of these two states. Already I have mention that when I use this particular apply operation, I am going to get an OBDDs for this particular operation S 1 and S 2 and the variable ordering of these particular result in OBDDs will be the same which are B s 1 and B s 2.

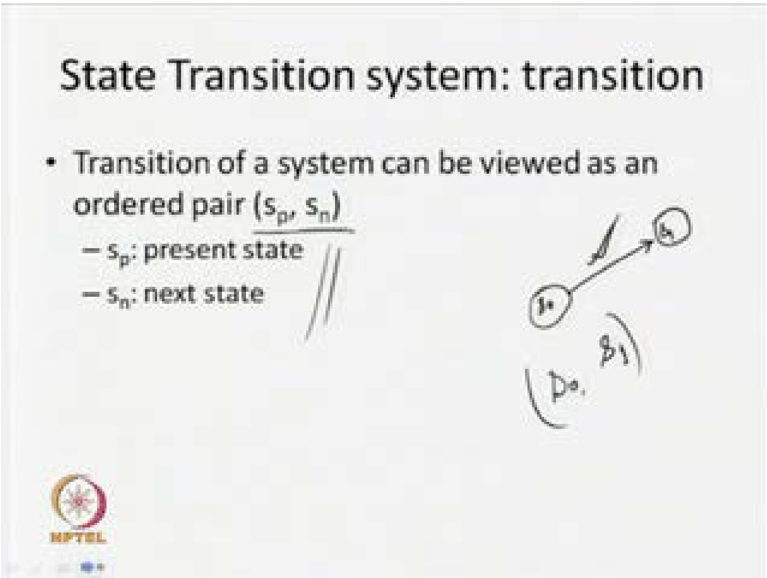
Again you should remember one thing, when we are going to use this particular apply algorithm, both the BDDs must a computable variable ordering; that means you should have the same variable ordering. And the variable ordering of our resulting BDDs also same with those particular input BDDs but what if a OBDDs we are getting it may not be reduce one. So after that what happen? I can use this particular reduce algorithm to get ROBDDs for this particular intersection operation. So if we are having BDD representation of also in particular I can say that, order BDD representation of two states, then I can use this particular apply operation, apply algorithm to construct union of these two states or in the section of these two states like that we can construct other operation also; so this is one advantage we are getting.

(Refer Slide Time: 17:15)



Now what will happen? We have discus we have talk about the state transition system; basically we have seen that in state transition system, it is a collection of some states and we have discus how to represent those particular states with ROBDDs, how to represent the set of states with the help ROBDDs.

(Refer Slide Time: 17:35)

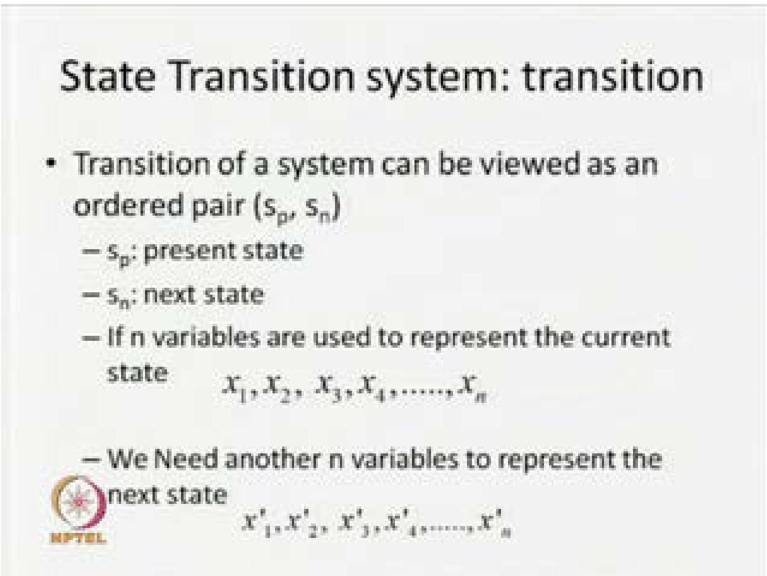


Now when we talk about the state transition system, then we are having another component which is your transition. Basically say this is your state S 0 and this is your state S 1, now we are having these particular transition, we are having a transition from S

0 to S 1 in our state transition graph. Now our aim is to, we should look for how to represent this particular transition, then if we can represent this particular transition. Then what we can do? We can represent the entire state transition diagram, because that state transition diagram is nothing but the collection of those particular transition.

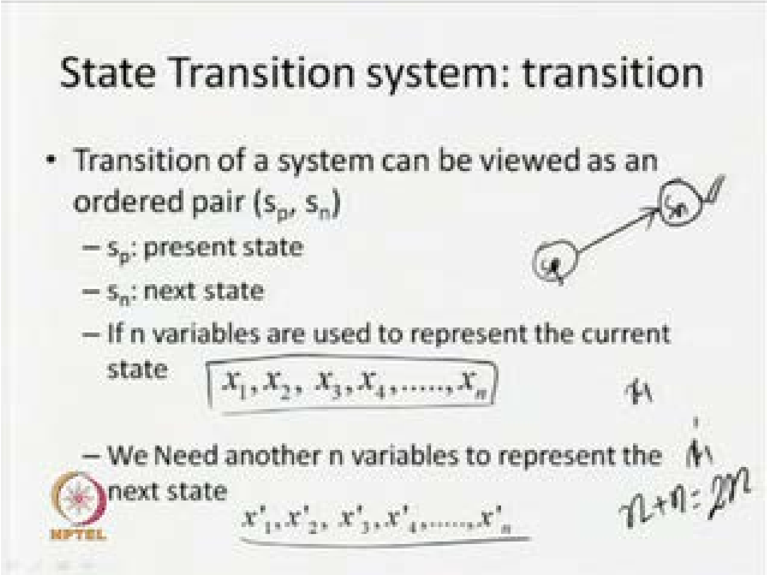
Now when we are going to look into this particular transition, then it can be viewed as an order of your S 0 and S 1. We can say, this particular transition is an order of your which is a collection of two states and here S 0, it is having a transition from S 0 to S 1. So we can say that S 0 is the current state or present state of my system; once it make a transition then it will go to the S 1, then I will say that this is a next state of my transition. So that is why I am saying that, this transition can be represented with the help of an order of your S p and S n. Where S p is the present state and S n is the next state. So with the help of these particular things, we can represent that transition and once we collect all the transition, then we are going to get the entire state transition diagram or behavior of state transition diagram. Now we are going to see, how these particular transitions are going to represented or how we can capture this particular transition.

(Refer Slide Time: 19:11)



So in this particular case say, we already I have said that if we are having some n number of states then we need total or minimum number of state variable as your log n to device too. Now you just see that, we are having a state transition system where n variables are use to represents the states.

(Refer Slide Time: 19:35)

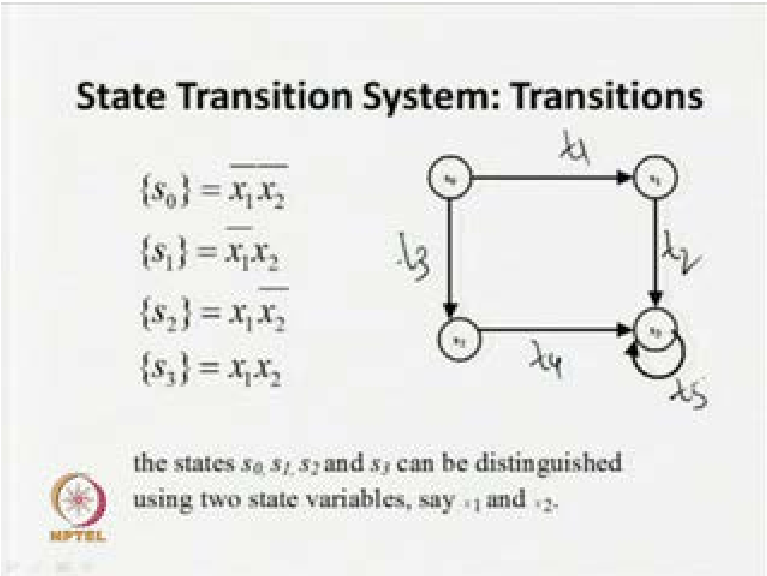


So we can say that, these are the state variable x 1 to x n. Say these are the state variable require to represent the state; and I will say that, these particular state variable is representing the current state or present state. My system is in a particular state in a particular instant of time and I am going to say that in which state I am or in which state my transition system is. I am going to represent this with the help of those particular state variables. By the evaluation of this particular state variable, it is going to give me the current state. Now when I am having a transition, say from one state S 0 or say S p to S n present state. Then we have to represent this particular next state also, where we are going to after making this particular transition. I say that these particular next state behavior will be again represent next state will be again represented with the help of n state variable, because this is n state variable is required to present this particular present state.

I need another n state, n variable to represent this particular next state and I will say that the prime version of the original variables, are used to represent this particular next state behavior. So if I am having x 1 variable, which is related to your present state S p, then corresponding to S 1. I am going to one variable I will say that, this is one x 1 prime which will represent or which used to represent to this particular next state. So that means, we need another set of state variable to represent this particular next state. So we are now having the representation of the present state, we need n variables, we are having the representation of the next state which involve another n variable, so that means we need to tell 2 n variable.

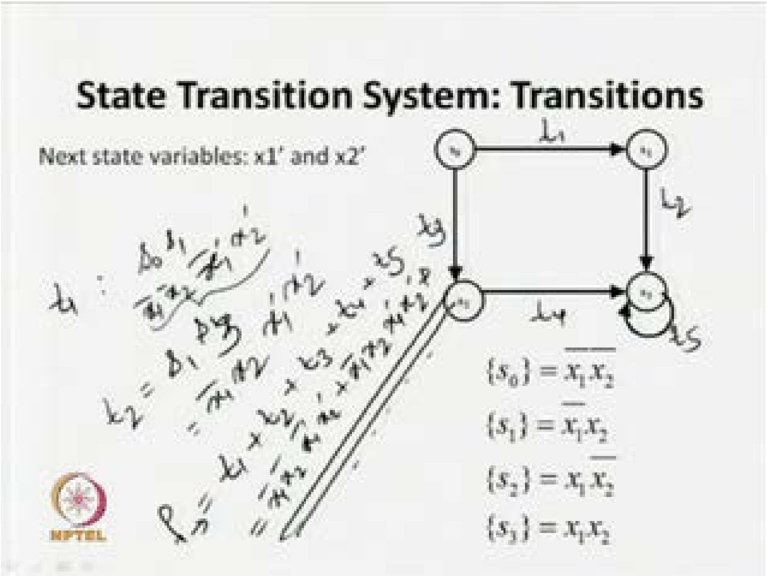
So that means, if we are going to use n variable to encode our state, to represent this particular transition we need 2 n variables; because this is nothing but the combination of present state and next state. n variables are use to represent the present state, another n variable will be used to present the next state. So we need 2 n variables. So if we are having x 1 x two x 3 like that x n as the present state variable, then the prime version will be use to represent the next state. And combination of these particular variables are going to use or going to represent our, that particular transition.

(Refer Slide Time: 22:06)



Now we are going to see this particular representation with the help of our earlier example. What we have seen? This is the state transition diagram, it is having four states: S 0, S 1, S 2, S 3; and these particular states are represented with the help of this particular state variable and this is the binary encoding of this particular states. Now we are having some transitions, so 1, 2, 3, 4, 5; we are having five transitions: transition t 1, transition t 2, transition t 3, transition t 4 and transition t 5. When I am going to represent this particular state transition system, we have to represent those particular transitions now. How we are going to represent it?

(Refer Slide Time: 22:41)



So you just see that this is are the states, and x 1 and x two are use to represent the states and you can say that these are use to represent the present state. Then we are going to use next state variable as you on x 1 prime and x 2 prime, x 1 prime is corresponds to variable x 1 and x 2 correspond to x 2 prime corresponds to the variable x 2. Now with the help of these particular four variables, we are going to represent those particular transitions four.

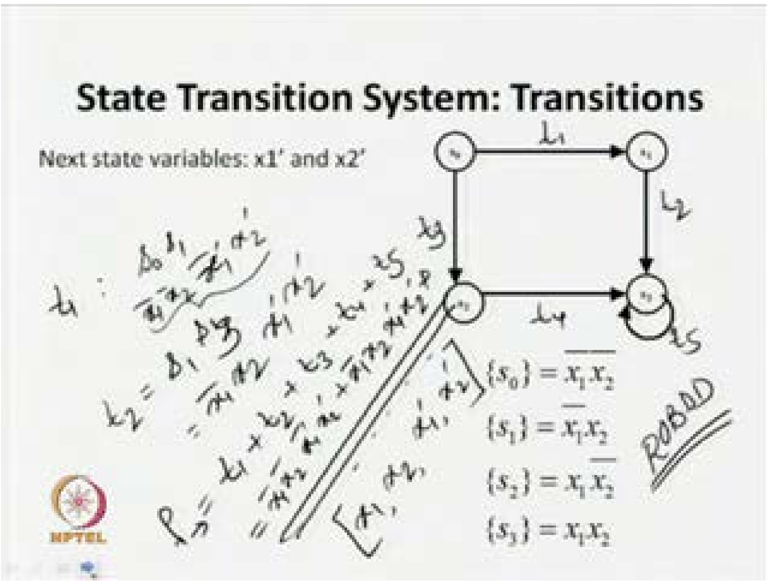
So how we are to say these things? Now just say that, now already I have said that this is your transition t 1, so this is transition t 1 is nothing but from S 0 to S 1. Now what is your S 0? I am saying that S 0 is your x 1 prime x 1 bar x 2 bar and it is going to next state as your S 1. S 1 is represented by x 1 bar x 2. So that means, we can say that this is your transition I am going to represent x 1 prime bar and x 2 prime bar.

So you just see that, with the help of this combination I am going to represent this particular transition t 1, so this is transition t 2. Now I can say that transition t 2 is nothing but it is going from S 1 to S 4, S 1 to S 4. Now what is your S 1? This is your x 1 bar x 2 and x 4 I am representing, sorry. This is your S 1 to S 3, S 3 is represented by x 1 and x 2; that means this is the next state variable will be represented by prime version, x 1 prime and x two prime.

So like that I can have a transition like S t 3, t 4 and t 5. Now all those particular transition, all those five transition will be represent that with the help of summation of those particular thing; that means t 1 plus t 2 plus t 3 plus t 4 plus t 5. So these are the five transition, that we are having in this particular transition; so these expression is going to represent this particular f transition behavior. And already I have seen that now, t 1 is represent that, transition t 1 will be represented by x 1 bar x 2 bar x 1 prime bar x 2 prime plus x 1 bar x 2 x 1 prime x 2 prime, like that I can write the expression. Now you just see that collection of those particular transition, is going to give me the behavior of this particular transition system.

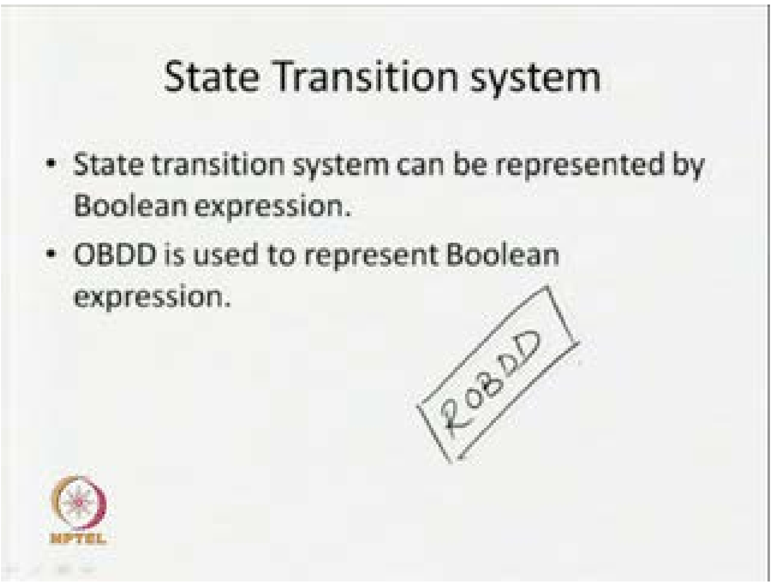
That means now see, what we have seen? Then again those all those particular transition can be represented with the help of a Boolean expression, you are getting a Boolean expression. Once we have this particular Boolean expression, then what we can do? We can construct a BDDs for this particular Boolean expression and once we have this particular Boolean expression, we can use the reduce algorithm to get the reduce binary decision diagram. And if we stick your particular variable ordering, then we have going to get an order binary decision diagram; so ultimately you have to get the reduce order binary decision diagram of this particular state transition system.

(Refer Slide Time: 26:24)



So in this particular case I can say that, I can use the variable ordering as your x 1, x 2, x 1 prime, x 2 prime. I can take this particular variable ordering also. If I am going to take this particular variable ordering, then what will get? Will get an order binary decision diagram of this particular Boolean expression and once we are getting they are order binary decision of this particular Boolean expression, then what we can say that? We are representing the state transition behavior of my system with the help of this particular ordered binary decision diagram. After that if we apply the reduce algorithm to that particular BDDs, order BDDs we have going to get ROBDDs of that particular transition system; that means we have to you can say that, ROBDD can be used to represents state transition diagram.

(Refer Slide Time: 27:16)

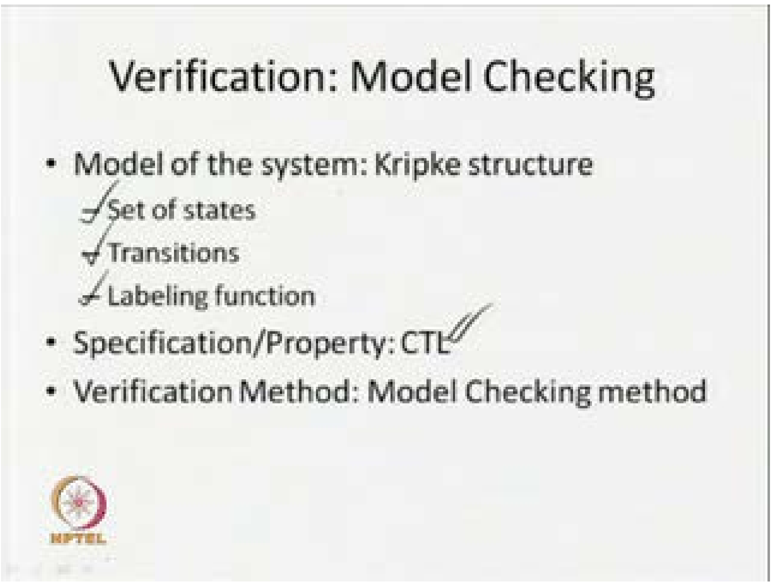


So that is why you are saying that, state transition system can be represented by Boolean expression. Already we have seen, how we are going to represent this particular state transition system with the help of Boolean expression; this is nothing but the combination of all those particular transition and once we are having that particular Boolean expression, then OBDDs can be used to represents those particular Boolean expression. That means eventually we can say that, OBDDs can be use to represent the state transition diagram, that means the behavior of our state transition system has been represented by our Order Binary Decision Diagram; and once we reduce it, then we can say that ROBDDs is used to represent this particular state transition diagram. And if we stick to one particular variable ordering, then we have going to get an unique representation of that particular state transition diagram.

Ok now we have seen or we have discuss a particular data structure call BDD, Binary Decision Diagram and eventually you have come up to reduce order binary decision diagram. With the help of these particular BDDs, what we can do? You can represent any Boolean expression and eventually you have seen that these particular ROBDDs can be use to represent any state transition diagram. Because if you are having a sequential circuit, the sequential circuit can be represented with the help of your state transition diagram; and the behavior of state condition diagram can be represented with the help of ROBDD.

Now before that, what we are discussing about? We are talking about verification technique, and the name of this particular verification technique is your model checking. We have seen this particular verification technique it is nothing but a property verification technique; where we are going to represent or particular state transition diagram or we say that this is a Kripke structure. The property will be represented that with the help of CTL, because we have talk about CTL model checker only; we are having other model checker also.

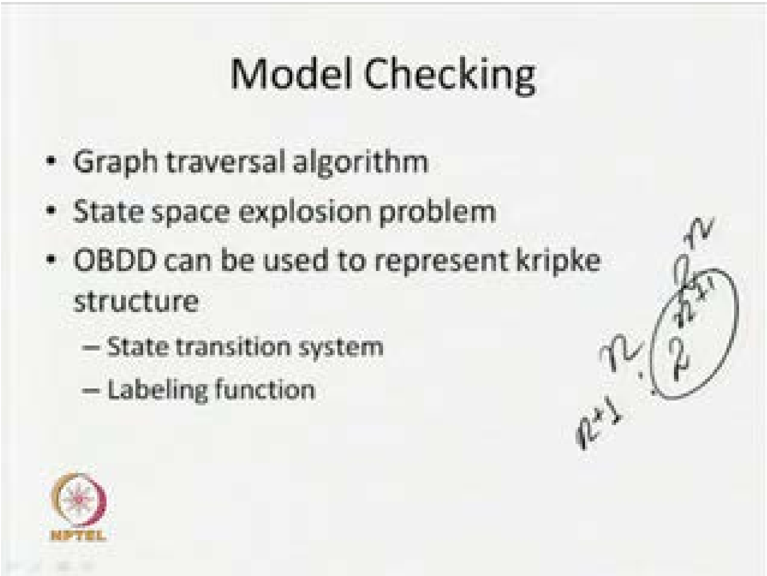
(Refer Slide Time: 29:31)



So in model checking system, we are going to use the Kripke structure to represent the states or represent the transition system or the model award system. And the Kripke structure is having three components, one is your set of state, second one is your transition and third one is your labeling function. So these are the three components that we are having.

Now what we are having? The specification and property, we are going to represent with the help of your CTL: Computational Tree Logic, because we have discuss about CTL model checker only. And after that, we having a verification method; in that verification method, what we are going to see basically? What we are doing is, we are going to check whether this particular given CTL property is true in this particular model of the system on not.

(Refer Slide Time: 30:07)



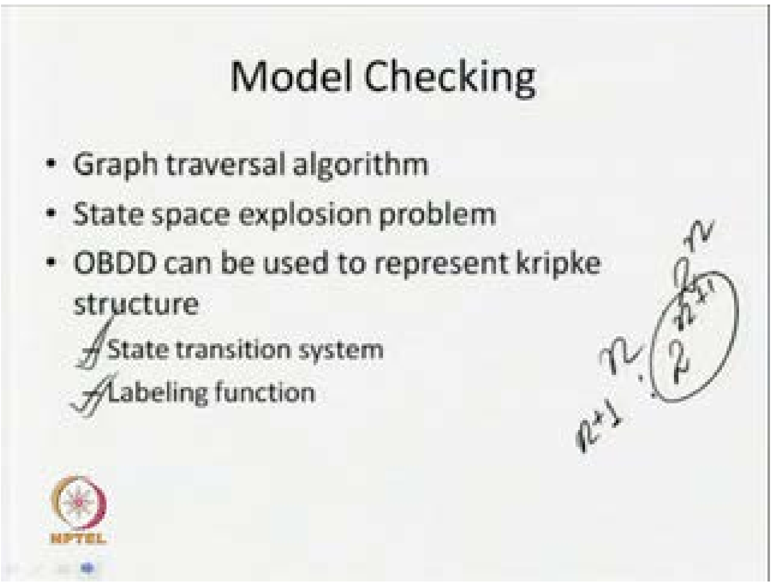
Now when we talk about this particular model checking algorithm, what we have seen that? This is nothing but say graph traversal algorithm. Because we are having the entire state space Kripke structure and Kripke structure can be few it as a graph, it is a simple graph only. And we are having a graph traversal algorithm to find out the set of states where s particular given properties is true. But, we have seen one problem with this particular model checking, which is your graph base model checker. It is having the problem with a state explosion problem, because already you have seen that. The number of states a related to the number of states variable we have. So we are having n variables n state variables, then we are going to get 2 to the power n different states. These are deferent possible combination that we may have and these are the different states that may have.

But in particular all states may not be disable; But in my design say, it requires another one variable that means, if I say that I cannot do it n variable I need n plus 1 variable. So in that particular case, the number of states we have to going to get about 2 to the power n plus 1; that means, the number of states is exponential in nature with respect to the state variables. So this is the measure problem if you go for a bigger system that this state is will be very higher.

Now how to contain this particular state space? So these BDDs can be used to contain this particular state space explosion problem, because already you have seen that the state transition system can be represented with the help of ROBDDs; and in most of the cases you have seen that, most of the Boolean expression can be represented with the help of a compact ROBDDs, also I have mention.

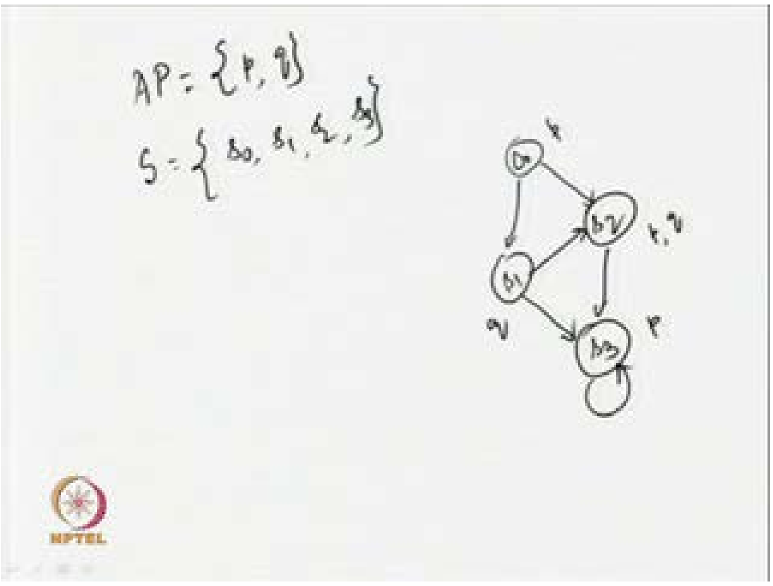
That size of this ROBDD depends on the variable ordering. For a particular variable ordering we may get very compact representation but we for the same function we defined variable ordering, we may not get compact representation it may be a bigger one. Also to find out the appropriate variable ordering is a hard problem; we do not have any algorithm which will say that, this particular variable ordering we will give to the best representation, best ROBDDs representation for this particular Boolean function.

(Refer Slide Time: 32:45)



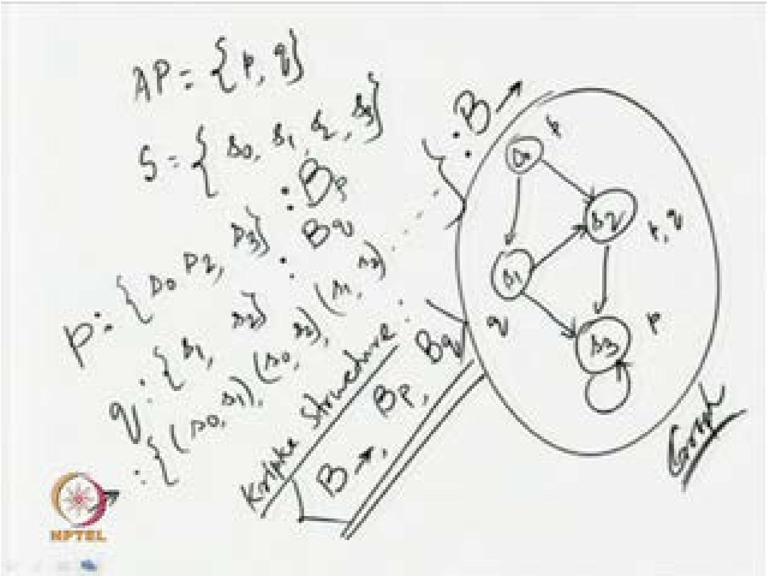
So this is the hard problem; so you use some heuristic to get some reasonable, some variable ordering, which will give as a reasonable size of a ROBDDs. So that OBDDs can be use to state transition system but in our Kripke structure we are having another component which is your leveling function. Now we are how we are going to represent this particular leveling function. Just see is better I have having slide or not.

(Refer Slide Time: 33:02)



So in this particular case what will happen? You just see that, how I am going to keep this particular leveling function. So conceder one particular state transition system, say this is s 0. Say just I am define this thing, so this is Kripke structure say p, q. Now you just see that, in this particular state transition system as a Kripke structure, I am having two these things. So atomic proposition, that set of atomic position is your p n q and it is having four states, S is having four states: s 0, s 1, s 2, s 3. And on the other hand these are level with this particular atomic proposition, when you are talk about Kripke structure, we need the state consign system along with this particular leveling function. Now when you are using OBDDs to represent the state transition system along with that, we must have the information of this particular labeling function. Then on way model checking or you can say that ROBDDs can be use for model checking algorithm.

(Refer Slide Time: 34:05)



So in this particular case now you just see that. That which are the state, where this particular atomic proposition p is true; so if you see this thing, then this is your s 0, s 2 and s 3. And where which say the state, where this particular atomic proposition Q is true; then in this particular case we will find that this is your s 1 and s 2. And we know the state, that is state transition system that here, we are having said this is a transition and what are a transition, I can say that this is s 0 to s 1, s 0 to s 2, like that s 1 to s 2. Like that we can list all those particular transition 1 2 3 4 5 6; so these are the information we have.

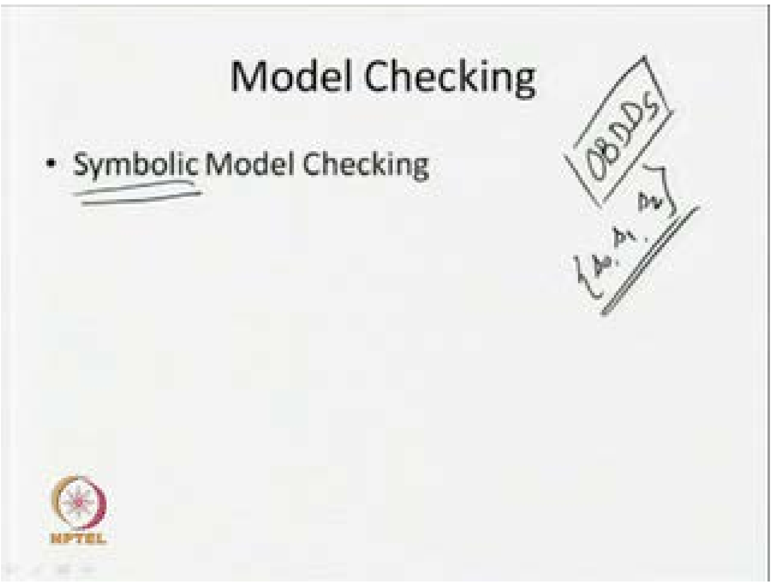
Now this is a subsets, this subsets can be represented with the help of BDDs; that I can say that, this is the BDD representation of those particular states, where this particular atomic preposition phase two. Now I can represent these particular states with the help of another BDD. I can say this is the BDD B q which is represent the set of states, where this particular atomic preposition phase two. And these are the state, these are the state transition that we have in our system and already we have seen how to represent this particular transition behavior of the system with the help of ROBDDs.

So I can say that, this is your B arrow is the ROBDD representation of this particular transition relation. And you just see that it is if I am looking over the order BDDs, all these BDDs are having comfortable variable order. Now, what I can say that for Kripke structure, we should have the state transaction behavior; and along with that, we must the labeling function. So that means, we can say that B arrow B p and B q. These three BDDs is going to represent by this Kripke structure.

So if I am having this particular Kripke structure, that state transition diagram along with this particular labeling function. So this information can be captured with the help of these three BDDs. Now you just see that that means, ROBDDs can be use to represent Kripke structure also, so once we represent the Kripke structure with the help of ROBDD.

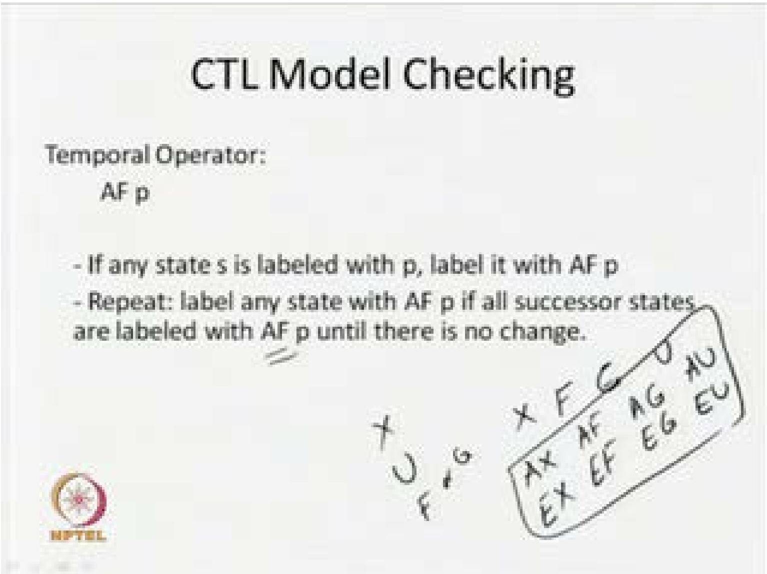
Now you see, whether that models checking can be perform with the help of this ROBDDs or not. Because already we know that if I am representing this Kripke structure with the help of a graph, then I can use some graph traversal algorithm to implement of model checking algorithm; and to find out which are the states that a particular property is true, so this is the graph theoretical algorithm elaborating we are using. So since we are using this particular BDDs to represent the Kripke structure. Now whether can I use this particular ROBDD representation through come up with some model checking algorithm.

(Refer Slide Time: 37:41)



So in this particular case, it is possible and we are going to say, this is your symbolic model checking. So if OBDDs are use to implement our model checking algorithm, then we say this is the symbolic model checking algorithm. And we are giving the term symbolic, because here the state transition diagram has been represented symbolically with the help of ROBDDs; and set of states I am going to have a combination of set of states s 0, s 1, s 2 like that. This particular set of state again represented with the help of an ROBDD, you can say this is a symbolic representation of this particular subset; that is why you have using this particular terms symbolic, symbolic model checking. So symbolic model checking is nothing but your model checking algorithm only. But, here we are using the data structure BDDs to represent our entire state transition diagram, that means we are using the ROBDDs to represent our Kripke structure. Now we are going to see, how we are going to implement of model checking algorithm with the help of those particular ROBDDs.

(Refer Slide Time: 38:49)



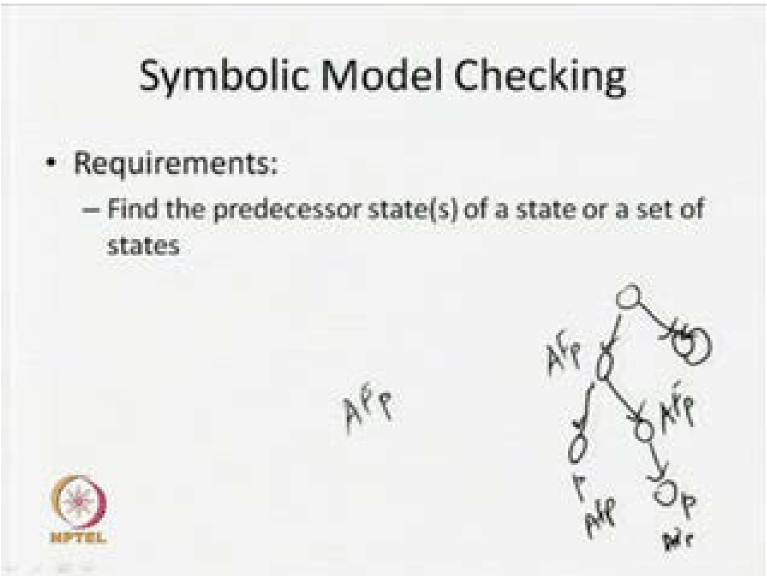
Now you just see that, I am coupling of these particular already; we have discussed about this particular temporal operator A F p, because already we know that. We are having some operators say next state, future, globally and until; and along with that we are having two part quantifier; so in all part A X, A F, A G and A U. Until operator is a binary operator, but X F and G are your inner operator. And another part quantifier we are having E X, E F, E G, E U.

Now we need the procedure to evaluate this particular operator. But already we have seen that, we need three operators, which are the adequate set of the operators. At least it should contain one next state operator, one until operator and we can choose one of the operators like F or G either A G. So already we have discussed this particular operator A

F p and we have see n how we are going to evaluate this particular A F p. We are having the algorithm to evaluate this particular temporal operator A F p. So in a lecture, how you have going to do? If any state s is labeled it p, then labeled with A F p, then we are going to repeat this procedural label any state with A F p, if all successor states can be are labeled with A F p until there is no change.

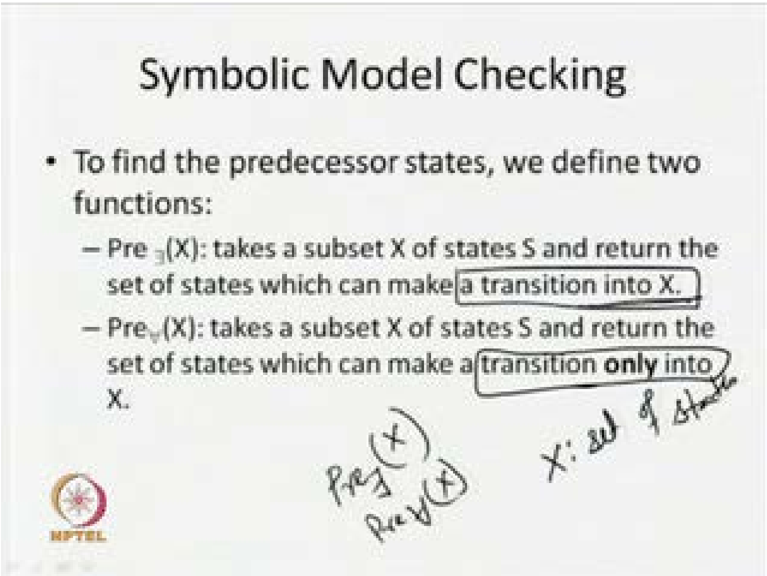
Now you just see that, we are starting from some state, if it is labeled with p; I will going to say that, A F p is to over there. Then we are going to find out all the predecessor of this particular state. If these all the predecessor state also you say that A F p is true, because in future we are going to get some state, if all the successor are having labeled with A F p. So this is the way that we are going to find out this particular operator A F p which is graph traversal algorithm.

(Refer Slide Time: 40:49)



So basically, what is our basic requirements if you look into nature, find a predecessors state of a state or a set of states. So basically you just see that, if I am going to have this particular transition system. Now say, if this is your p and p is true, then in these two case, I am going to say that A F p is true. Now what I am going to see, I will going to see the predecessor. So form this particular predecessor I will say that, A F p is true over here; A f p is true over here. Now in this particular this predecessor is having only one successor, A f p is true; so I am going to labeled it with A f p. Now what is particular states in both the successor I having A F p so I am going to labeled it with A F p. Now when I come to this thing, I will find that one is labeled with A F p but second one is not labeled with A F p, so we are not going to labeled with A F p. So you just see that in this particular method what happen? We have to find out the predecessor states of a particular state of states or a particular state, and depending on operator I am going to say that one particular CTL formula is true on this particular states or not. So in our graph traversal algorithm, so basic the common is the find out the predecessor states or particular states or a particular set of states.

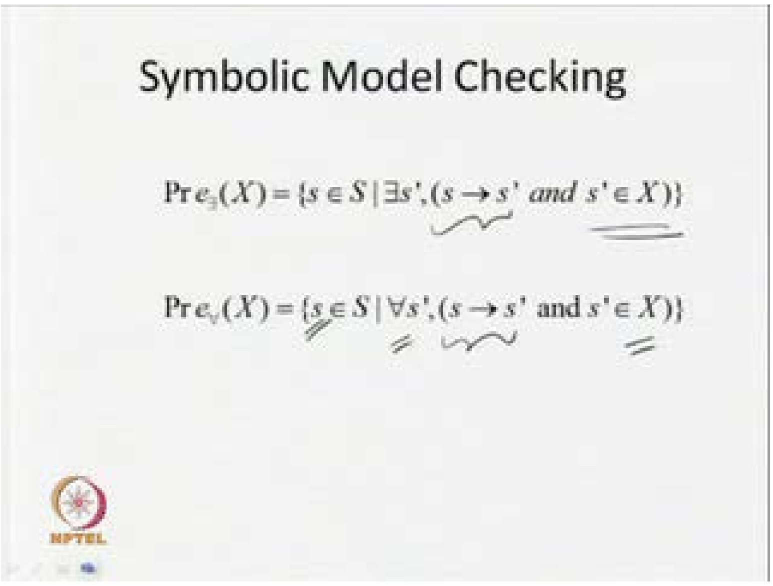
(Refer Slide Time: 42:15)



Now for that, now when we are going to look for this particular symbolic model checking. That means, we need two operators: one we are going to say that pre they are exist X and Pre for all X. So these are the two operator will be Pre they are exist X and Pre for all X; where X is a set of state.

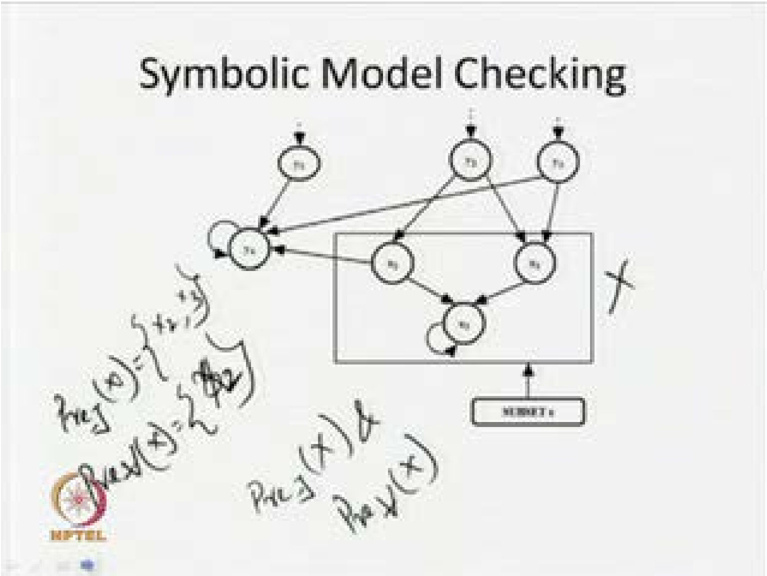
So what we are going to say that, Pre they are exist X, now we take a subset x of state s and return the set of state; which can make a transition into X. So, we are having a subset X, we are going to set of state which can make a transition to this particular subset. And Pre for X, it is similar to Pre they are exist X; but in this particular case, we are going to take those particular states which can make a transition only into. This is the graphs, transition only into and here we can say that make a transition into X, so they are exist. At least one transition is coming to X, I can say that we are exist a transition, which is going to this particular subset X; that means if I am giving a subset X, then we are going to find out a predecessor state of those particular subset X; well at least one transition is coming to this particular state.

(Refer Slide Time: 44:19)



But, here we talk about Pre for all X; that means, we are going to considered those states which are going to make a transition to this particular subset X only. That means all transitions with go to this particular X. So that is why I have saying that, make a transition only on two X it should not go to this. So our in model checking algorithm all basic requirement is to find out the predecessor state, and we can find out this particular predecessor state with the help of this two operator Pre they are exist X and Pre for all X. Now this is basically in a metrical (( )), we can say that Pre they are exist X; you have going to collect all those particular states belongs to X; such that you have existed state as that and we are having a transition from s to s dash and s dash belongs to X. and for Pre for all X, we are going to collect all states from s; which are member of this given entire state space. Were for all s dash if we are having a transition from s to s dash then s dash must belongs to X. So this is your for all transition must go to X be a Pre they are exist at least one transition must go to X. So, these are the Pre for all X and Pre they are exist X and basically these two operators going to give me the predecessor state or the given set of states.

(Refer Slide Time: 45:07)



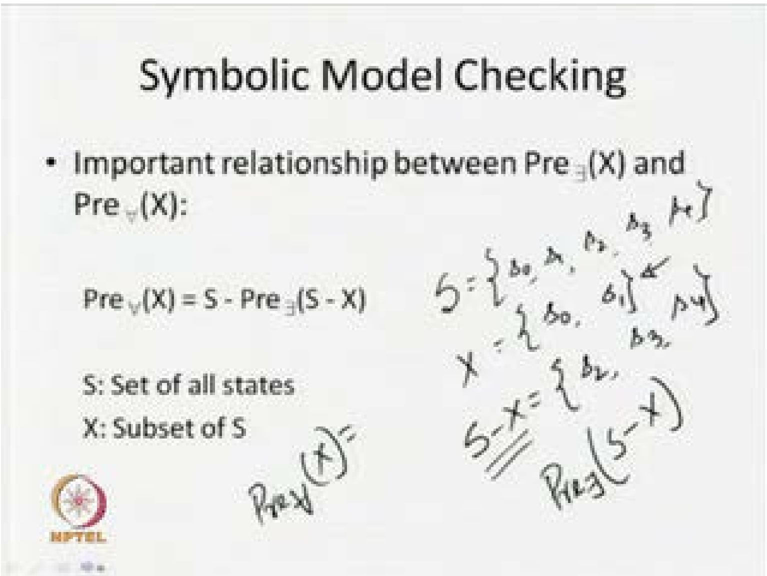
Now just see this is a symbol. Example just whatever we are talking about Pre they are exist X and Pre for all X. Now in this particular case, this is a subset we are considering say this is your X, X it is having this particular three state; now we are going to look for, basically we are going to look, we are going to envelope the predecessor state in such way that at least one transition will come to this particular X, subset X or Pre for all X; all the transition will come to this particular state.

So in this particular case, if you say that pre for all X then will find this particular state say s 1 s 2 s 3. So for s 2 we are having two transitions, both that transition are coming to this particular subset x. So basically it is going to say me, if this is your this X I am writing, say X 2 is the states which are all the transition coming to this (( )). And Pre they are exist at least one transition must come to this things, so in this particular case since all are coming; so pre they are exist X also I am having that X 2. When I come to this particular X 3 state then we will find that, it is having two transition; at least one is coming to this particular subset X, so X 2 X 3 is going to give me the set of states, where it is satisfying that Pre they are exist X. So, these are things that we will be ready over here.

Now you just see that we are having these particular two operator, Pre they are exist X and Pre for all X; with the help of this thing we can find out the predecessor states. Now here when we are going to look, calculating this particular predecessor state and we know that these are required to if all that a model checking algorithm; because already we have seen this particular process, where we are going to say that in such type of scenario. That at least we have to calculate the predecessor state and we are going to collect all this particular things.

Now we are seeing that Pre for all X and Pre they are exist X, we can evaluate those particular states, where in Pre for all X all the things are coming to this particular subset X; and Pre they are exist X at least one transition is coming to this particular set of states. Now we are having a relationship between these two things: Pre they are exist X and Pre for all X.

(Refer Slide Time: 47:50)

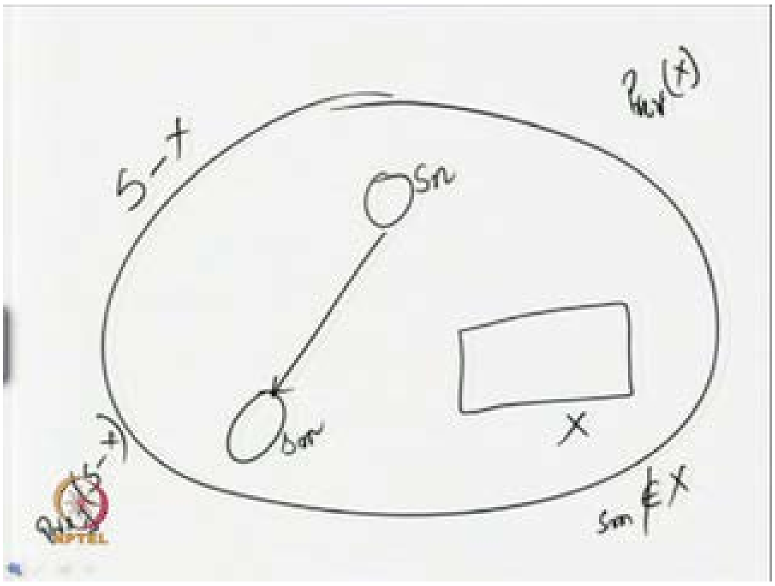


Now this is the imported relationship between Pre they are exist X and Pre for all X. Now this is now Pre for all X can be, if represent that with the help of Pre they are exist X and this is the relationship Pre for all X equal to S minus Pre they are exist S minus X. Where S is the set of all state and X is the sub of S. Now what we can say that, how we are going to have this things, Pre for all X is equal to S minus Pre they are exist S minus X. Now say, I am having a given set of state say S is equal to say I am having s 0, s 1, s 2, s 3, s 4. Now say that I am having a subset X which is having say s 0 and s 1.

Now in this particular case if I am having s 0 and s 1 x like that then what is my S minus X? S minus X is basically s 2, s 3, s 4. Now you just see that, if I am going to evaluate for Pre for all X, then what will happen? I am going to consider all those Pre predecessor states where all that transition is coming to this particular two states S were s 0 and s 1. This S is the set of all states; s 0 and s 1. So Pre for all X in this particular case, all transition must come to this particular states.

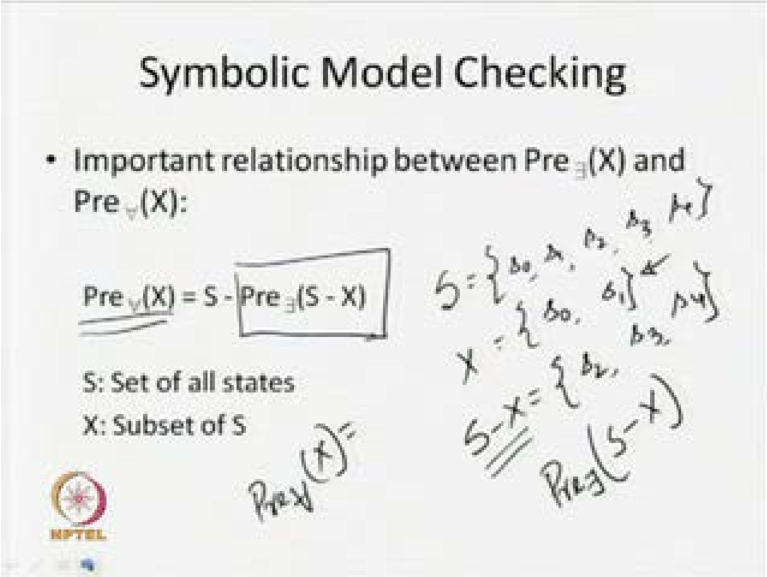
Now I am going to take the set different, I am going to look for S minus X. So s 2, s 3, s 4 is the set of states which represent S minus X. Now when I am going to consider that Pre they are exist S minus X. So in this particular case what will happen? From a particular state at least one transition is coming to this particular S minus X. So you just see that, if for a particular states, if one transition is going to S minus X; that means all transaction will got cannot go to this particular X, so see that.

(Refer Slide Time: 50:05)



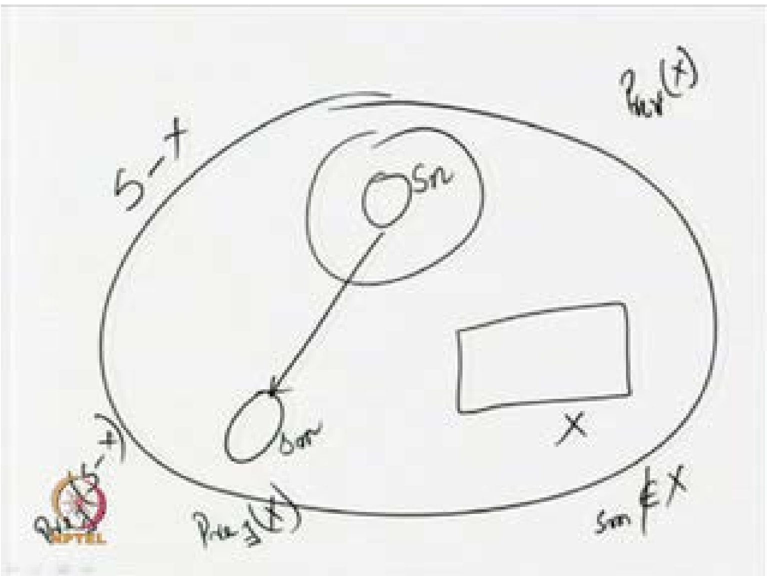
So this is my set of state X, and other person is basically I am viewing this is one as S minus X. Now if I consider one particular state say S n. Now if one of the transition coming to say this is say S m which is not belongs to your X that means, S m is not belongs to your X. That means at least you are having one transition, which is not going to this particular X. So this is the things that I am getting this transition, I am going to look for Pre they are exist S minus X. So this particular states will not come under Pre for all X, because all transitions are not going to X. So that is why what we are doing? First we are everything Pre they are exist S minus X, so such type of things we are getting and that means such type of state S n will come in to picture and it should not be included in Pre for all X.

(Refer Slide Time: 51:12)



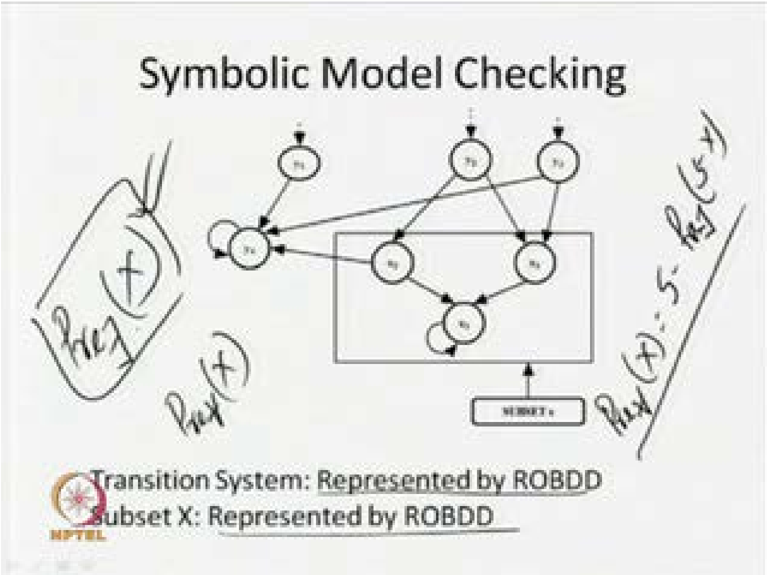
So in this particular case I am evaluating this particular things and eventually I am removing those particular states from S. At least where transition is at least one transition is going outside of this particular X. So the remaining state will be, a remaining state is going to give a Pre for all X. So in this particular case, what will happen? You just see that here, we are evaluating Pre for all X with the help of Pre they are exist S minus X. Ok, this is the way that we are going to evaluate; that means, if we are having a procedural power Pre they are axis X. Then form here we can allow evaluate Pre for all X.

(Refer Slide Time: 51:55)



That means you just see that, we need one procedural which is going to evaluate Pre they are exist X, then very well we can calculate Pre for all X.

(Refer Slide Time: 52:05)

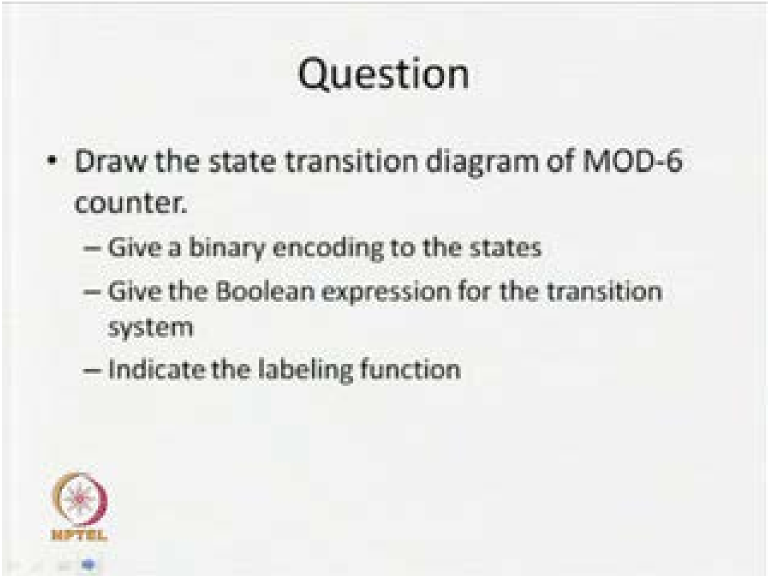


So that is why what happens? Now you just see that the whole transition system we can represent with the help of ROBDDs, if we are having a subset of X it can also represented by ROBDDs. Now we have to see how we are going to evaluate these particular predecessors there, already we have seen that. We can, if I can get a procedural of Pre they are exist X it is going to give me all the states from which at least one transition is coming to this particular subset X. Now once we have this particular procedural I can always calculate Pre for all X, because we know the relationship Pre for all X is equal to S minus Pre they are exist S minus X. So if I am remain a procedural for Pre they are exist X we can evaluate Pre for all X.

Now how to evaluate these things? Now in this particular case, now we are going to, today I am going to stop here. In next class I am going to say how we are going to evaluate Pre they are exist X and once we can have the procedural for this particular Pre they exist X. Then what happens? We can evaluate Pre for all X also and once we have the particular procedural, then see how we are going to implement of model checking algorithm with the help of this particular ROBDDs. That means we are going to represent our transition system with the help of ROBDDs. And now we are going to implement those particular model checking algorithm, with the help of this particular

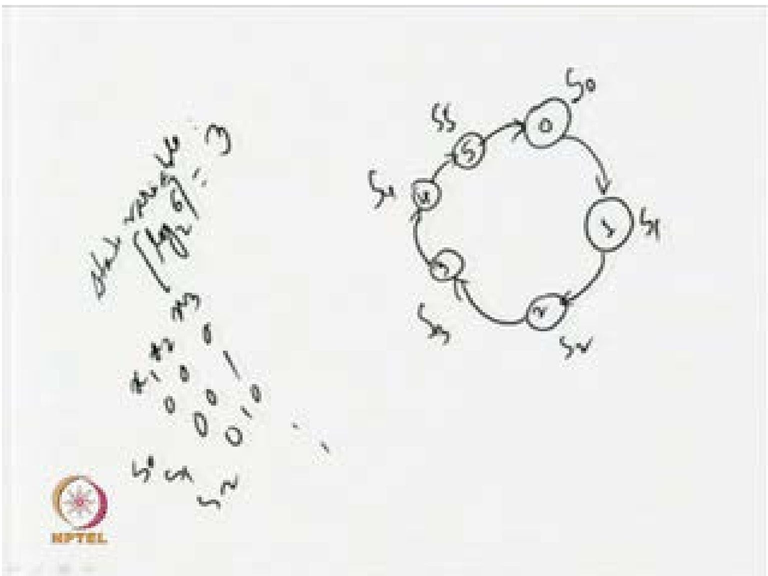
ROBDDs and in this particular case, we are going to say this is the symbolic model checking algorithm.

(Refer Slide Time: 53:52)



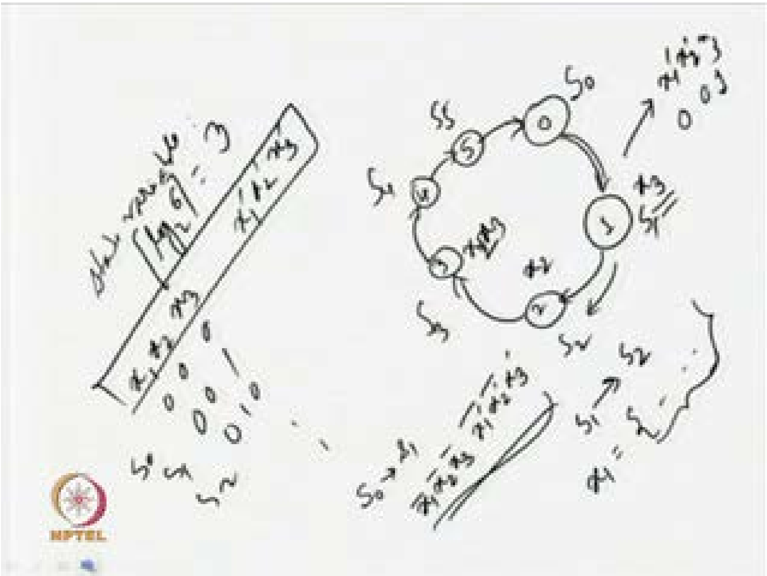
Now just look in to the question, that I am saying that draw the state transition diagram of MOD-6 counter. So what we have today? Give a binary encoding of the states, give the Boolean expression for the transition system and indicate the labeling function. We have to indicate the labeling function, which are the labeling function.

(Refer Slide Time: 54:18)



Now how we are going to say? I am talking about the MOD-6 functions. So MOD-6 counter basically, so it will come from 0 to 5 again reset to 0; that means what happen? I can say that, if this is your 0 then I am going to have 1, then I am going to have 2, then I am going to have 3, 4, 5; then again group of back to 0. That means how many states we are having? 6 states we are having. That means you will need, how many states variable will be needed for it? This is your log 6 to debase 2 ceiling, so that means we need three variables. So I can say that x 1 x 2 x 3 other three variables; and what that I say, if I say that this my state S 0, S 1, S 2, S 3, S 4, S 5. Then I can say that S 0 state at putting will be 0 0 0, S 1 will be 0 0 1, S 2 will be 0 1 0 like that these are the state encoding.

(Refer Slide Time: 55:35)

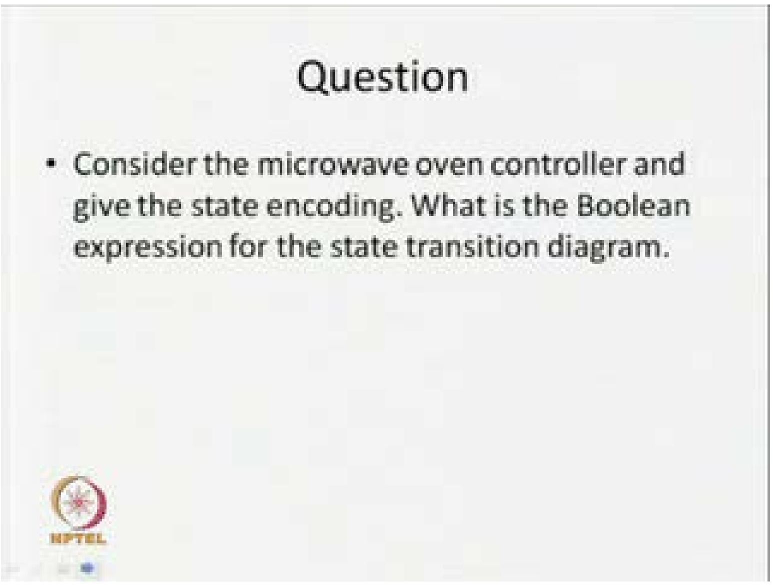


Now what I am saying, that first we have to say give the binary encoding of the states, this is the binary encoding give the Boolean expression for the state transition system. Now what will happen? Now what are the state transition? We can say that, I am having a transition from S 0 to s 1, now since to represent this particular transition there where are I need another set of variables. Then I can say that x 1 prime, x two prime, x 3 prime. So these transition from S 0 to s 1 can be represented with the help of x 1 bar, x 2 bar, x 3 bar; this is 0 0 to 0 1. x 1 prime bar, x 2 prime bar, x 3 prime.

So this is the transition behavior for this particular expression for this particular transition similarly, for the second transition also I can represent it is going from S 0, S 1 to S 2. Like that I can represent all those particular six transition, with the help of the six particular variables x 1, x 2, x 3, x 1 prime, x 2 prime, x 3 prime and we can having now Boolean expression for those particular state transition write down.

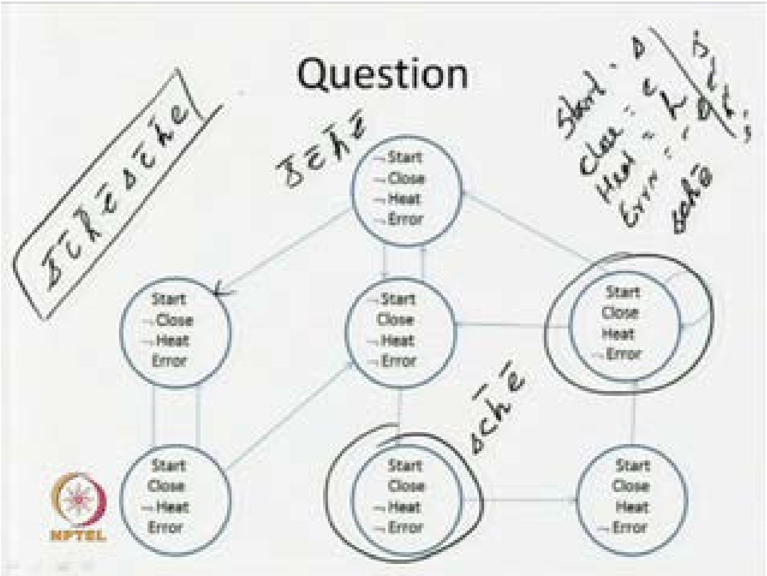
Now indicate the labeling function. Now where I am going to talk about this particular transition behavior, then you will find that x 1 x 2 x 3 all are 0, that means all of false. Here in S 1 I am going to say that, this is labeled to it x 3, this labeled to it x 2 because; what happen? I can say that this is nothing but x 1 prime x 2 prime x 3; this is 0 0 1. That means x 3 is true in this particular state S 1. So that means what I can say that? It is labeled to it x 3 similarly, it is label to your x 2 and I can say that, this is labeled to with your x 2 x 3. Like that we can have development and we already have seen the if I have the label now. I will be doing which are the states that x 1 is true, now we can collect those particular states; and we can represent the this particular state with the help of ROBDDs. So that is why, I am saying that indicate the labeling function. So this is the way we are going to do.

(Refer Slide Time: 57:50)



Secondly we are going to say that, consider a microwave oven controller and give the state encoding. What is the Boolean expression for the state transition diagram? So this is another question that I already discuss about this particular microwave oven controller and we are having that state transition diagram. This is the state transition diagram that we have.

(Refer Slide Time: 58:06)



In this particular case, I am here in four state variables. So I can say that Start, so I am going to say that represent state by ‘s’. Then I am going close, I am going to have this particular ‘c’ representing this particular close. Heat, I can say that ‘h’ is my heat; representing I am say Error ‘e’, Say ‘e’. I am going to represent error by ‘e’. So these are the four state variable, I will be leading to represent this particular state.

Now when I say that, what is the binary encoding of this particular state? I can say that it is labeled to it basically s bar, c bar, h bar and e bar. Similarly, when I say that, what is the state at encoding of this particular state? I can say that this is Start is true, Close is true, Heat is true but Error is false. So these are the state at coding so similarly, when I come to this particular state, what is this state encoding? And I can say that this is Start is true, Close is true, Heat is false, Error is false; so these are the state at coding. So what I am saying that? Consider the microwave oven controller and give the state encoding, so we can have that state encoding. What is the Boolean expression for the state transition diagram? Now in this particular case, now we have to consider all those particular transition, say I am having this particular transition it is going from.

Now we need another set of variables where I can say that the next state variable will be your s prime, c prime, h prime and e prime. So this particular transition I can represent this like that, this going from s bar, c bar, h bar, e bar; o s, close bar, h bar and e. This is the present state, this is the next state; so this particular transition we will represent that with the help of this particular Boolean expression. Like that I can write the expression for all those particular transition and eventually I can getting Boolean expression. That is why I am saying that, what is the Boolean expression for the state transition diagram? We can get the Boolean expression for the state transition diagram and once we get it, then what we can do? We can represent this particular Boolean expression with the help of ROBDDs. Then I will stop here. In next class we are going to say how we are going to evaluate Pre they are exist x; and with the help of that operator we are going to say, how we are going to implement the model checking algorithm using BDDs and that means, we are going to get the algorithm movement, of course symbolic model checking algorithm.

Thank you.