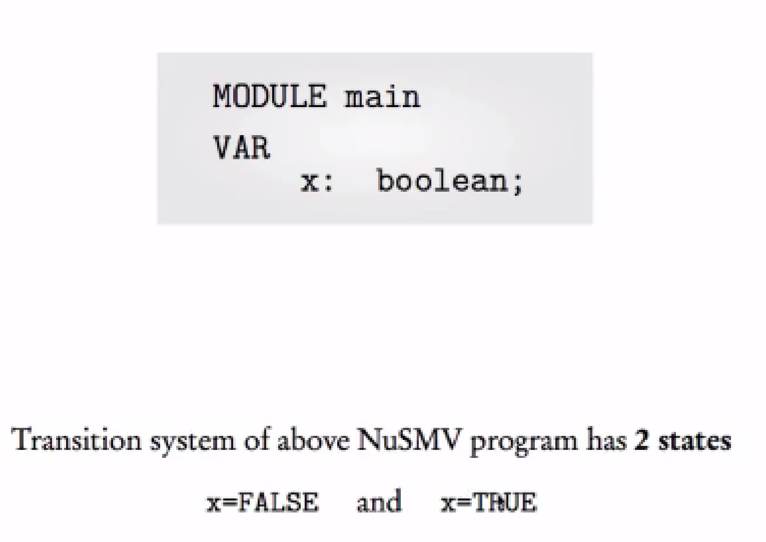
Model Checking

Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology- Madras

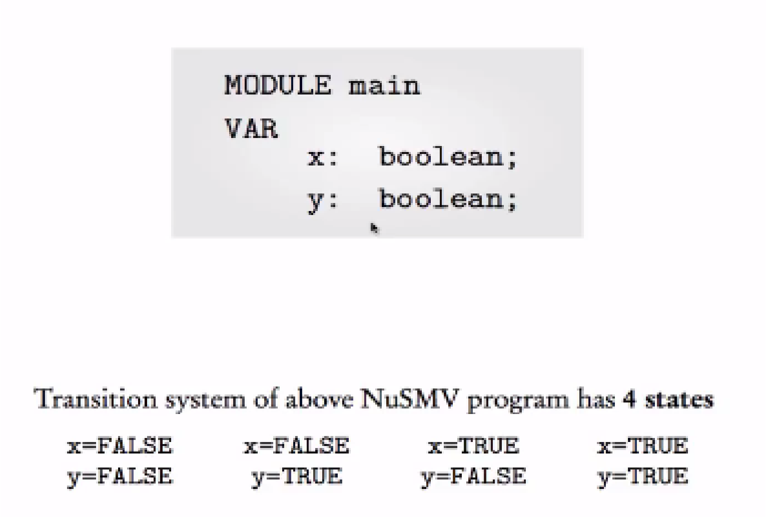
Lecture-48
State- Space Explosion

(Refer Slide Time: 00:18)



We are in unit 10 algorithm for CTL. In this final module of unit 10, I will talk about a challenge faced, while building model checking tools. Look at this NuSMV code. It has a single Boolean variable. The corresponding transition system of the simple NuSMV code will have 2 states; one where x is false, and the other x=true. You can just run this program in NUSMV and check print reachable states.

(Refer Slide Time: 00:57)



Since there is no assign block, both of them would be marked as initial states. Consider this NuSMV program, where you add one more Boolean variable, already you had x to be a Boolean variable, suppose you add a y, how many states would this code produce. There will be 4 states; x=false y false, x=false y true, x=true y false and x= true y true. So for every state here, you have 2 states.

(Refer Slide Time: 01:39)

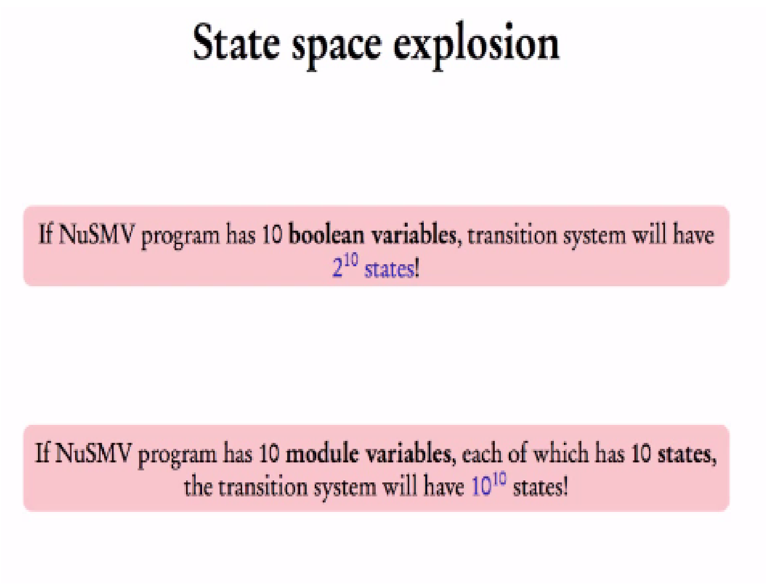
```
MODULE main
VAR
    x:  boolean;
    y:  boolean;
    input:  sys();
MODULE sys()
VAR
    state:  { s1, s2, s3, s4, s5 };
```

Transition system of above NuSMV program has $2 * 2 * 5$ states

So the number of states gets doubled, when you add a Boolean variable. Let us now look at a slightly complicated code, so there is a module which has one variable of the enumerated type, enum type and this variable can take, s1, s2, s3, s4, s5. Remember when we were doing NuSMV programs, we had variables like locations l1, l2, l3, similar to that. This says that this module has 5 values for the variable state.

Suppose in the main module, you have 2 Boolean variables and a variable of this kind of this module type. How many states would the final NuSMV code produce? With x and y, there were already 4 states, now with each of the states, you can associate one value for input, it could be either s1, s2, s3, s4 or s5, input dot state could be either s1, s2, s3, s4 or s5. So the total number of states would be 2 times 2 times 5.

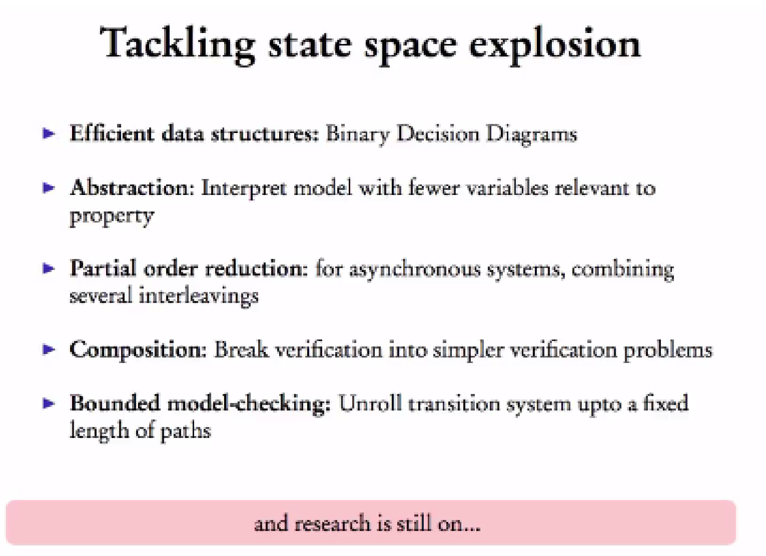
(Refer Slide Time: 03:11)



Adding a variable increases the number of states of the transition system by a multiplicative factor. So if we add one more Boolean, it will be 2 x 2 x 5 x 2 states, you can check this out in NuSMV. If a NuSMV program has 10 Boolean variables, then the transition system will already have 2 power 10 states. Now if the NuSMV program has 10 module variables, each of which has 10 states individually, then the transition system will have 10 raised to 10 states.

This is to say that if we have 10 transition systems, each with 10 states, then the joint behaviour is represented by a transition system with 10 raised to 10 states, it will be a product. If you remember our previous units, we did a product construction, so it will have 10 power 10 states. A similar thing is happening in this NuSMV program, each time we add a variable, we multiply the number of states by a factor. This is known as state space explosion.

(Refer Slide Time: 04:37)



When we are building model checking tools, we need to overcome this challenge somehow. We need to come up with ways of handling big transition systems. Usually industrial models are huge. So we need ways of tackling state space explosion. This has been an area of research for the last 20 to 30 years. In this line, I will give a brief sketch of various techniques used for tackling the state space explosion problem.

Firstly, we need to have efficient data structures, so there is a data structure called binary decision diagrams, which is used to represent the sets of states, that NuSMV produces. So this is implemented inside NuSMV. Many states are clubbed together and represented using one structure. An another method is to interpret your model with fewer variables relevant to the property. This is called abstraction.

Instead of looking at the entire model itself, we need to come up with a view of the model with fewer variables and we need to choose these variables in such a way that they are the ones which will affect the property that we have in mind and another method is partial order reduction. It is used, when we do a product of asynchronous systems, remember that when we do a product of asynchronous systems, in every state, only one of the component moves.

We had given a distinction between synchronous verses asynchronous composition, to that in case, only one of the components move, this will give rise to many possible states and partial order reduction is the way of eliminating some of those states. An another method is to break the verification problem into simpler verification problem. This is called compositional verification; you verify certain components of the system individually.

One more method is bounded model checking, where you look at the transition system up to a fixed length, you do not see the transition system in its entirety, you look at the evolution of the transition system up to a fixed length. These are some of the methods, research is still on to tackle the state space explosion problem. Model checking is an active area of research and a big challenge in model checking is tackling state space explosion.