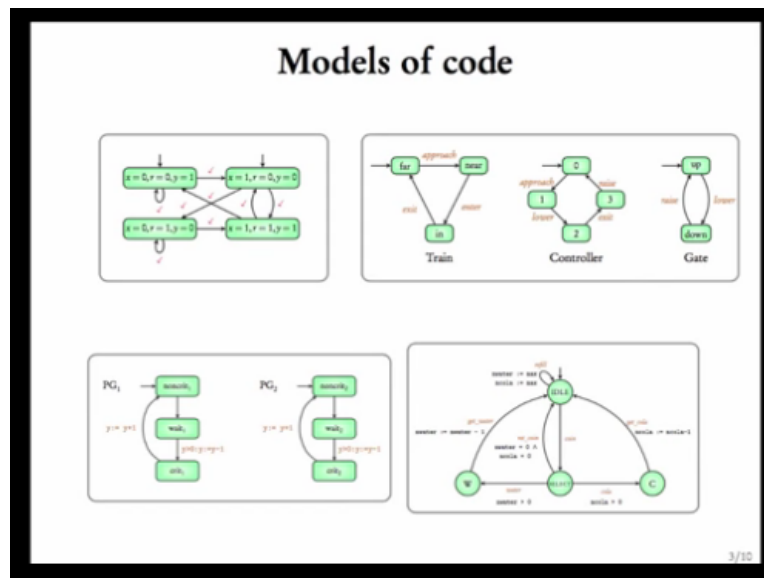


Model Checking
Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology – Madras

Lecture - 07
Model Checking Tools

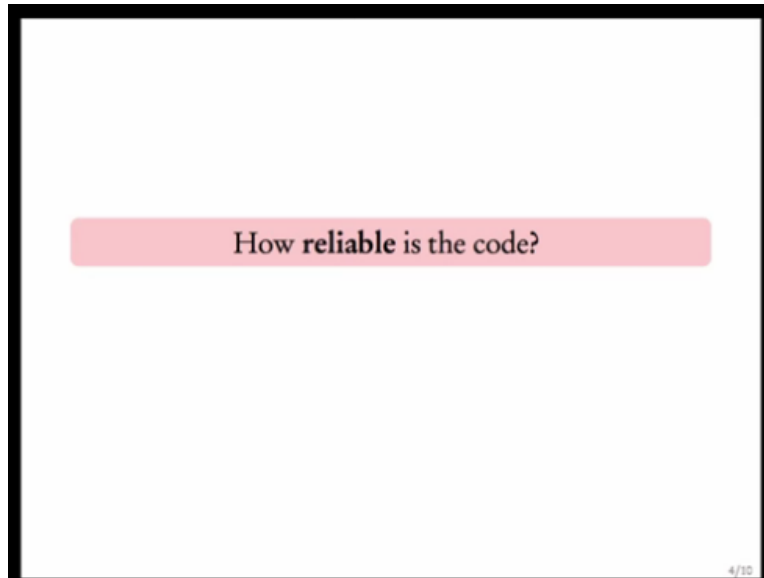
Welcome to the second week of lectures. In this unit we will give an introduction to a module checking software called NuSMV. Let's start with module 1 of this unit, we will see an introduction about general model checking tools.

(Refer Slide Time: 00:22)

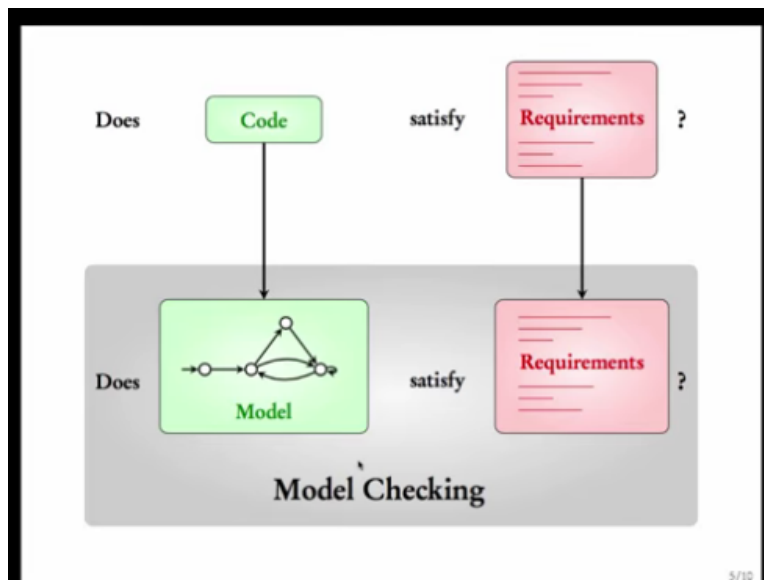


In the last unit we saw how to model different kinds of code as transition systems. In particular we saw how to model hardware circuits, data dependent programs and various kinds of parallel programs as transition systems. What do we do with these models? We will try to use these models to answer the following question.

(Refer Slide Time: 01:00)



(Refer Slide Time: 01:05)



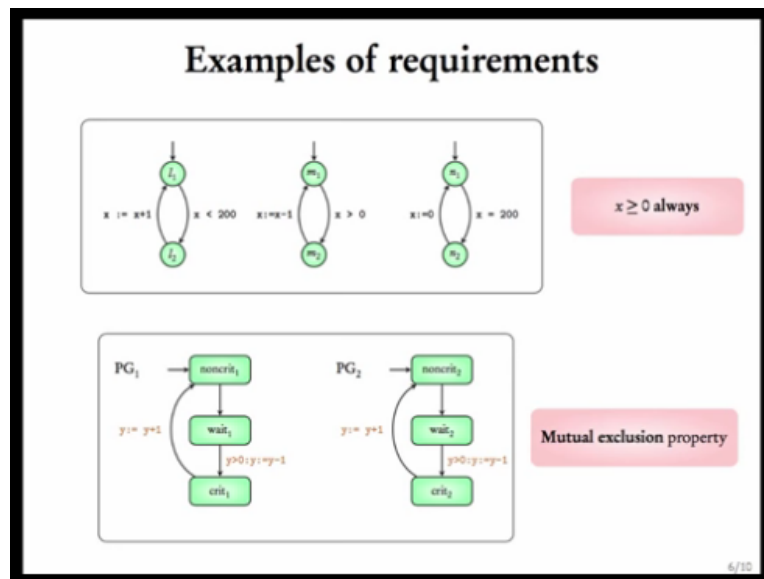
How reliable is the code, normally a code comes with a set of requirements that it needs to satisfy. For example if x is a variable in the code one of the requirements would be that the value of x is always between 0 and 50, another example if x and y are variables one requirement would be that x is less than y always during the execution of the code, yet another example whenever x becomes bigger than 50, y should become bigger than 200 these are some requirements.

One way of checking if the requirements hold on the code is to give test cases and check if the solutions meet the requirements. When the size of the code or the number of

parallel interacting components is large doing a manual verification is difficult. It is also possible that certain errors go unnoticed.

This is where we could need use of our models for the code. Checking the requirements can be done on the model of the code. As we saw before this technique of checking requirements on a model of the code is said to be model checking.

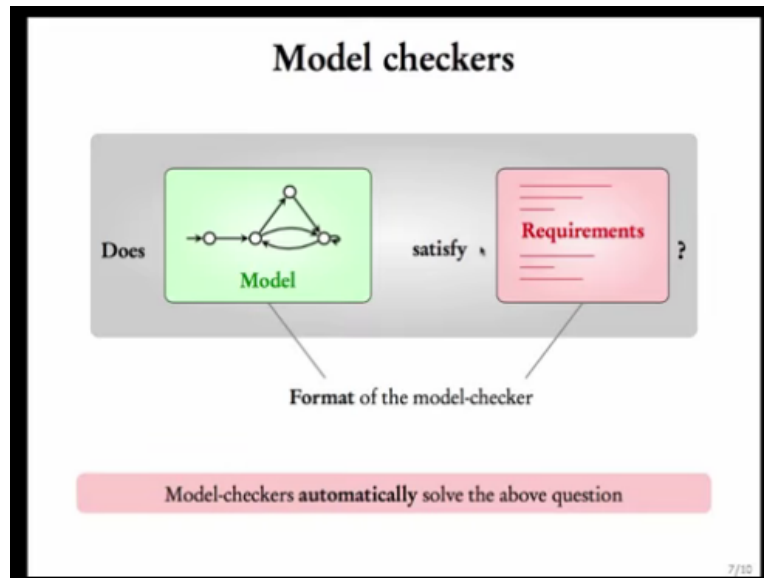
(Refer Slide Time: 02:50)



I have listed some examples of requirements; here is a system of 3 parallel programs. One of the requirements is x greater than or equal to 0 always. These are 2 parallel programs with an inbuilt logic for maintaining mutual exclusion one of the requirement could be to check if both the programs are not in their critical section simultaneously.

Model checkers are tools which can perform this task one needs to enter the model of the code in the format of the model checker and the requirements yet again in the format of the model checker. These 2 are the inputs, once these 2 inputs are given the model checkers can automatically solve the question whether the model satisfies the requirements.

(Refer Slide Time: 03:55)



Some of the open source model checkers are SPIN and NuSMV these are the most famous ones. SPIN is particularly suited for modeling and verification of concurrent systems NuSMV is most suited for hardware circuits the kind of requirements that NuSMV can check is richer as compared to the model checker spin.

One cannot make a generic statement saying that one model checker is better than the other it depends on the kind of questions that you are trying to solve. In this course, we would try to learn how to use the model checker NuSMV. The format of SPIN is different from that of NuSMV, however if you learn NuSMV it would very well be easier for you to learn SPIN.

(Refer Slide Time: 05:05)

Some open source model-checkers

- ▶ SPIN
 - ▶ Well-suited for **concurrent** systems
- ▶ NuSMV
 - ▶ Well-suited for **hardware** circuits
 - ▶ More requirements can be checked compared to SPIN

In this course: NuSMV

9/10

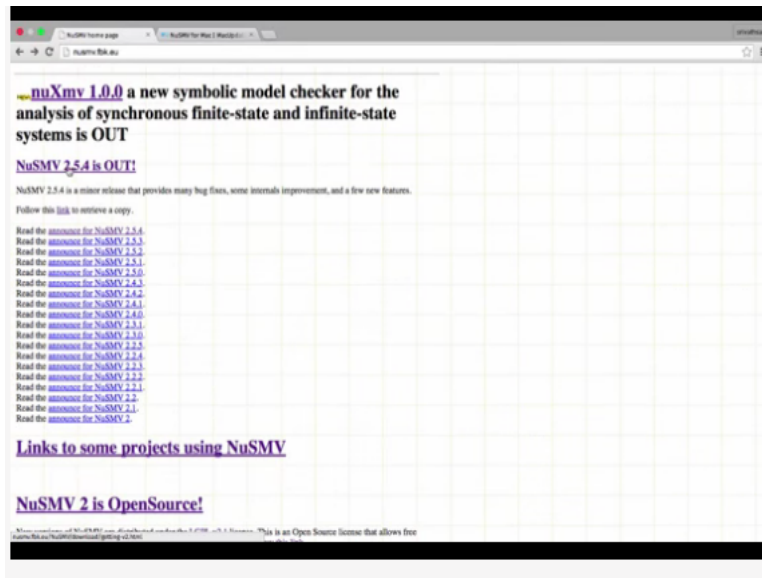
(Refer Slide Time: 05:06)

Installing NuSMV

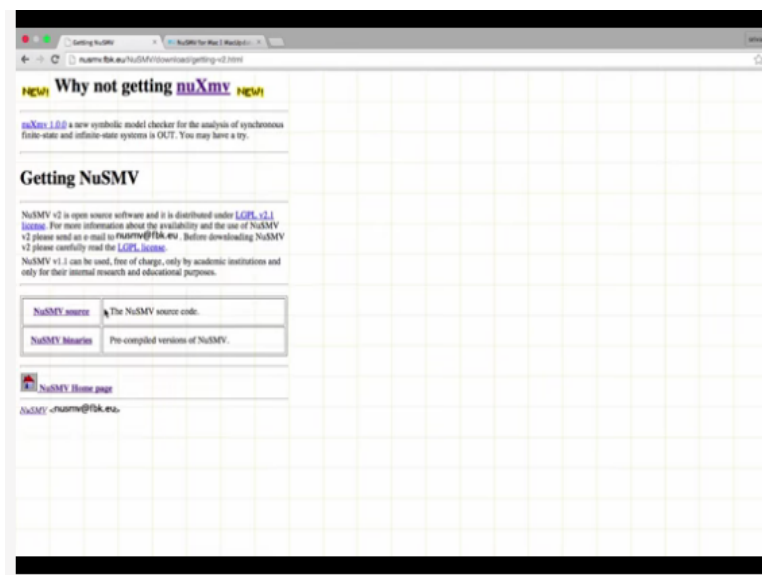
9/10

There are lot of online tutorials available to learn SPIN. I will now give you the websites from where you can download and install NuSMV. Go to this webpage NuSMV dot fbk dot eu. In this webpage you will find a lot of details about NuSMV. To install click on this link NuSMV 2 point 5 point 4.

(Refer Slide Time: 05:35)



(Refer Slide Time: 05:37)



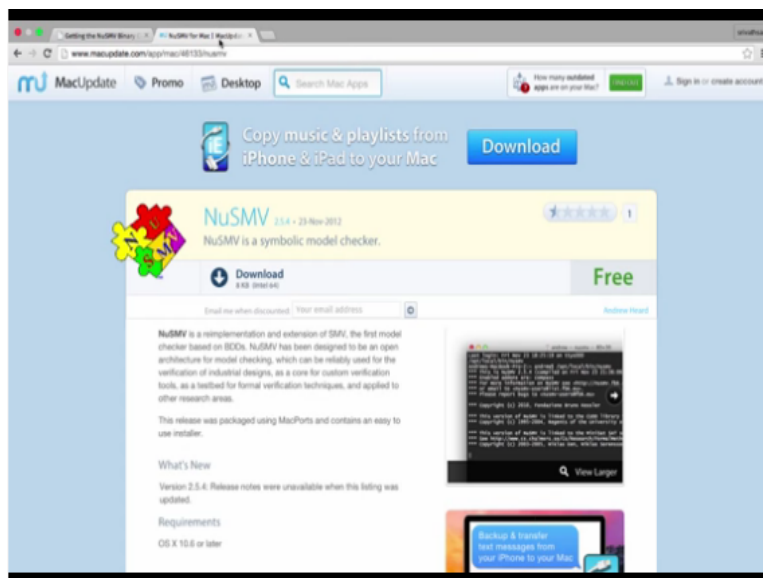
There are 2 possibilities either you can download the NuSMV source code or you can download precompiled binaries of NuSMV. For this course, we do not care about the source code so you could directly download the NuSMV binary files. Clicking on that link brings you to registration page.

(Refer Slide Time: 06:05)



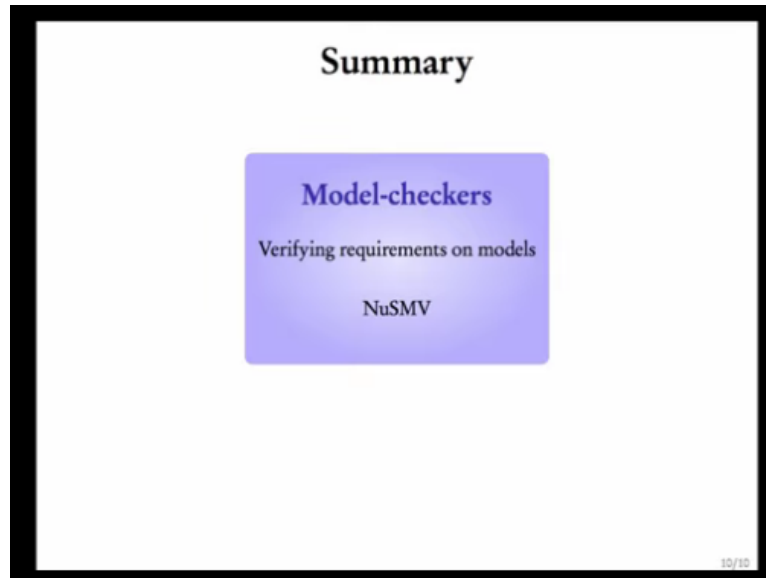
Once you register you will get to a page from where you can download the binary corresponding to your architecture. Mac users can download NuSMV from this page.

(Refer Slide Time: 06:24)



This brings us to the end of this module. Let me summarize what we have seen in this module.

(Refer Slide Time: 06:35)



We claimed that models of code can be used for verifying requirements. Model checkers are tools which can take both the model of the code and the requirements in a certain format and can check if the requirements are true on this model. We introduced the model checker NuSMV. NuSMV stands for new symbolic model verifier. Please install NuSMV on your machines because this week we will learn to use the model checker NuSMV.