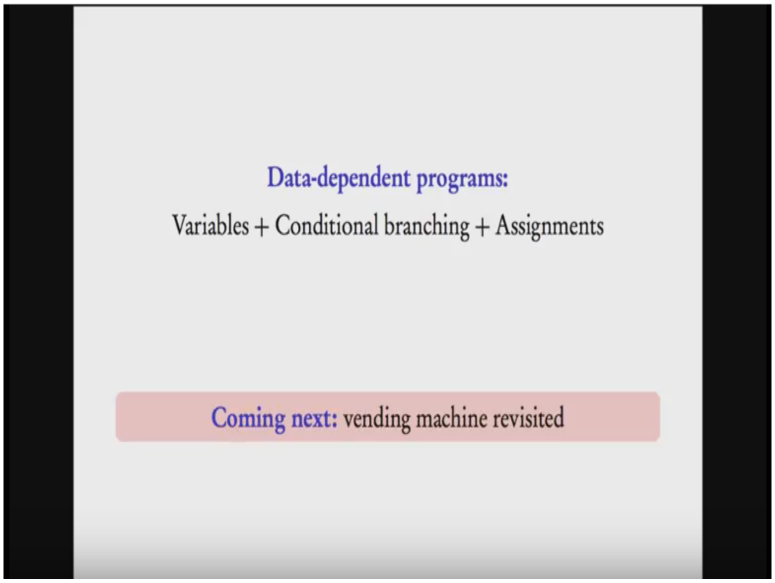
Model Checking
Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology – Madras

Lecture - 04
Modeling Data-dependent Programs

In the previous modules we have seen how to model simple programs and hardware circuits as transition systems. In this module we will be considering slightly more complicated programs which manipulate data such programs are called Data-dependent programs.

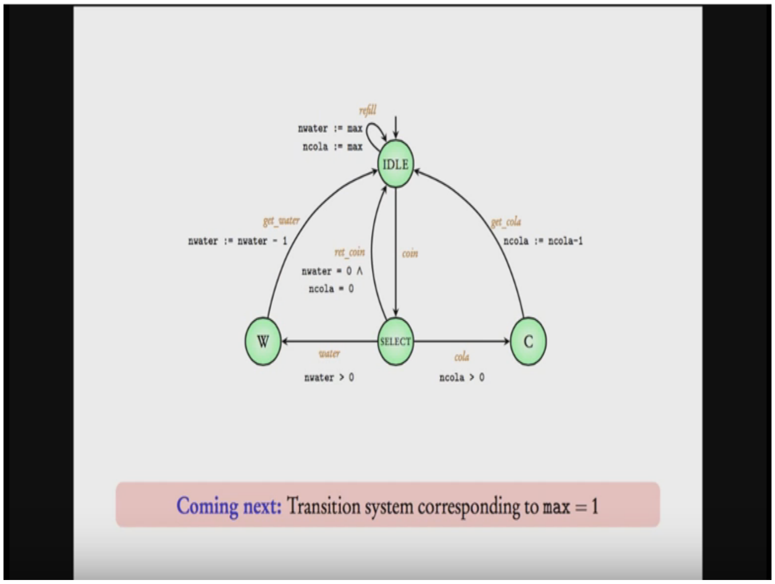
(Refer Slide Time: 00:28)



Essentially the program that we called Data-dependent programs consists of some variables. These variables could be of different types statements with conditional branching on these variables. For example if x is bigger than 0, if y is 5 and so on and statements consisting of assignments to variables like x goes to x-1, y is set to 5 and so on. The goal of this module is to model such Data-dependent programs. We will look at this concept through examples.

In the next slide we will be looking at the model of the vending machine which we had seen in module one.

(Refer Slide Time: 01:35)



Let us recall the transition system which represented the vending machine. The machine is initially in an idle state, when the user inserts a coin the machine moves into a select state. In the select state depending on which option is chosen by the user the vending machine goes into either state w or state c. Once it is in state w it remembers the fact that the user has selected water. Now it ejects a water bottle out and goes into the idle state.

Clearly the functioning of this vending machine depends on the numbers of water bottles and the numbers of cola cans currently available in the machine. These are the variables on which the vending machine depends on. Let nwater be the current number of water bottles available. Let ncola be the current number of cola cans available. Let us assume that the maximum number of water bottles that can be present in the machine is max.

Similarly the maximum number of cola cans that it can keep which also max these are the variables. How does the machine change these variables. Rather how does the code representing the machine change these variables. Each time a water bottle is given out the value of the variable is decremented by one. So along this transition nwater becomes nwater-1this is an assignment. Similarly when a cola can is given out the effect of this transition is to reduce the value of ncola by 1.

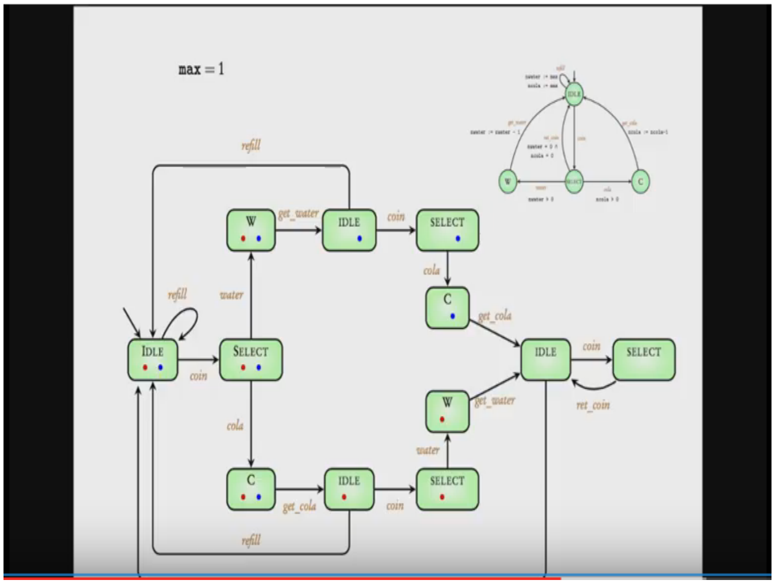
Note that from the select state the machine can go to state w only if the number of the water bottles currently available is bigger than 0. Similarly from select it can go to c only if ncola is bigger than 0. So this is a condition on the transition

if both of them are 0 let us add this transition which returns the coin. So nwater bigger than 0, ncola bigger than 0, nwater equal to 0 and ncola equal to 0 these are conditions on this transition nwater set to nwater-1 ncola set to ncola-1 are assignments on these transitions. These are the effects of these transitions to make it more interesting let us also add this refill option.

When the machine is in the idle state from time to time it can do a refill and what is the effect of this refill the number of water bottles become equal to the max and the number of cola cans becomes equal to the max. In addition to this diagram representing the working of the code we also need to know the initial condition. Let us assume that initially the number of water bottles is max and the number of cola cans is max. This kind of a picture along with a initial condition is set to be a program graph.

As you might have noticed a program graph looks different from a transition system. A transition system can have only action names on its edges. In a program graph in addition to the action name there are conditions and assignments. Given the initial values of the variables one can associate a transition system that represents the working of this program graph starting from these values of the variables. We will explain this taking the initial value of max to be equal to 1.

(Refer Slide Time: 07:32)



Starting from the value of max to be 1 the program graph performs different executions. We want a transition system that represents these executions. We will now give this transition system. To give the transition system we need to define its state and its transitions. Just to avoid confusion let me call the states of the program graph as locations. Now what are the states of the transition system that we are talking about.

The states of the transition system would include the location of the program graph and the values of the variables the current values of the variables. The red dot represents the fact that the value of nwater is 1 the number of red dots gives us the value nwater the number of blue dots gives us the value of ncola. Initially my program graph is in the idle location with nwater and ncola being 1. Let us look at the transitions,

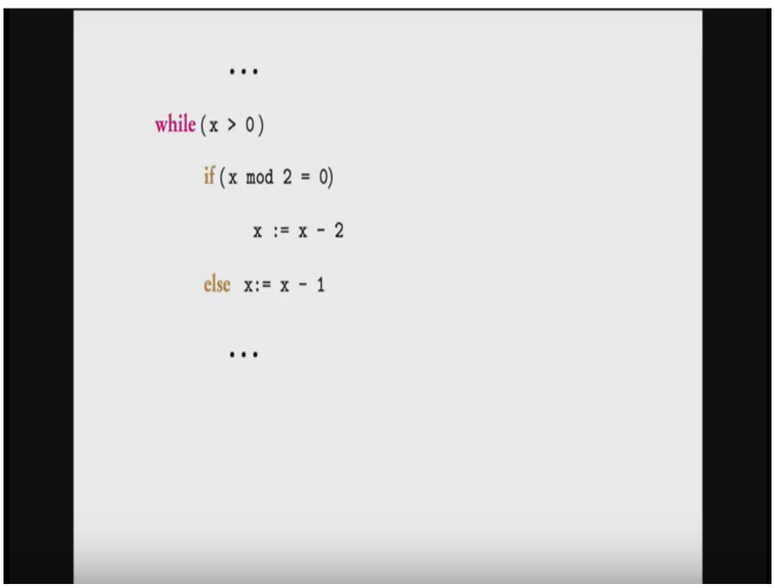
when a coin is inserted the program graph goes from idle to select there are no changes to the variables in this transition. In the transition system there is a transition from idle to select with no change to the variables. From select, there is a transition to w on water. This is possible provider the value of nwater is bigger than 0. Look at this state since the value of nwater is 1 this transition is possible. This transition does not change the values of the variables. So we go into the state with location w and nwater equal to 1 and ncola equal to 1.

Similarly from select we can take the transition that asks for the cola since ncola is bigger than 0 and ask this transition does not change the variables we go to state see with the same values for the variables. Let us get back to state w, when a water bottle is ejected the values of nwater becomes and nwater -1. This is this transitions from w on get water going to idle the effect of the transition is to reduce the value of nwater by 1. So in the transition system we go from w to idle. However, we do not go to this idle state we go to the idle state which has the value of nwater to be 0.

Similarly from cola this transition to idle will reduce the value of ncola by 1. So we go to a state where n water is unchanged. However the value of ncola is reduced by 1. Note that from here we do not go to this idle state or this idle state. We go to the idle state that reduces the current values of ncola by 1. Now in this idle state if a refill transition is taken you get back to the idle location. However, with nwater and ncola being set to max. Hence in the transition system no matter which idle state you take if you take this refill transition you go to the idle state which has n water and n cola being equal to 1.

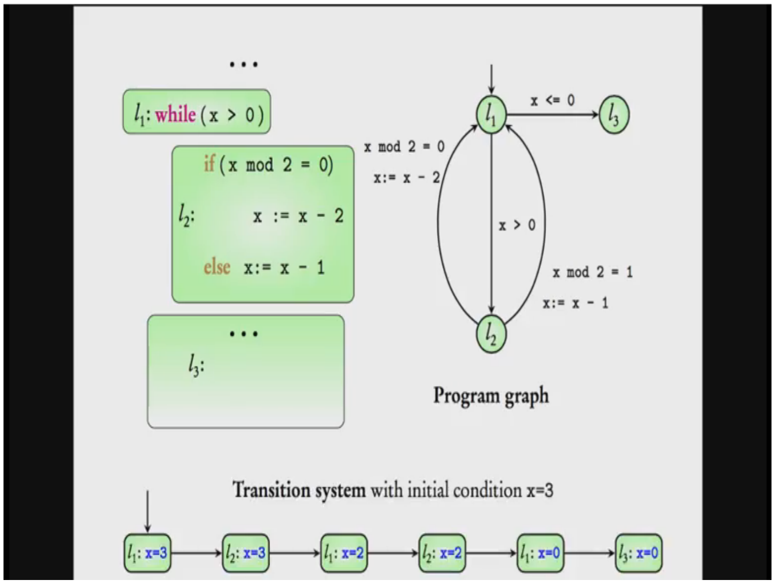
Similarly we can finish the picture this represents the transition system corresponding to this program graph with the initial value of the max equal to 1.

(Refer Slide Time: 12:56)



Let us now look at another example consider this program fragment manipulating a variable x there is a while loop with a condition on x. If x is bigger than 0 the loop is entered. If x is even the value its reduced by 2. If x is odd its value is reduced by 1 and yet again this condition is checked. Let us try to model this program fragment as a program graph.

(Refer Slide Time: 13:45)

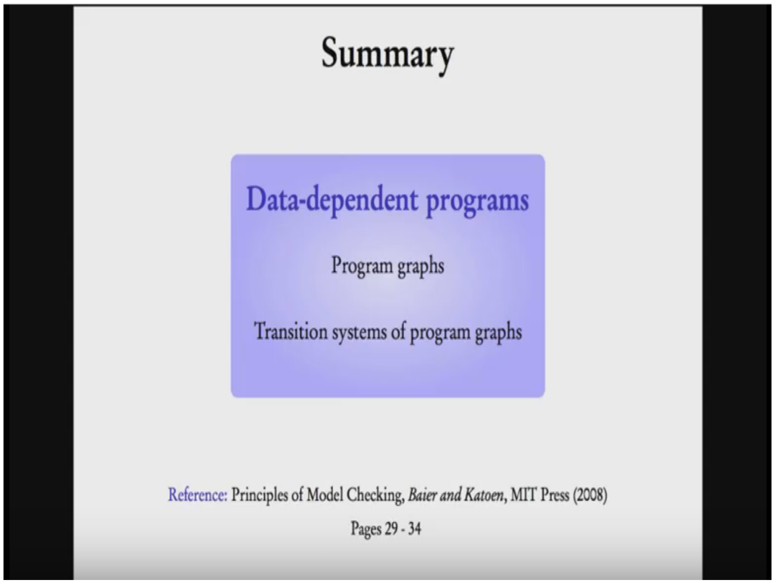


The locations of the program graph are as follows the place where the value of x is checked before entering the while loop is location l1. The contents inside the while loop form location l2. The rest of the program below the while loop forms location l3. At l1 there are two transitions if x is bigger than 0 the while loop is entered otherwise the program jumps to l3. At l2 if x is even the value of x is reduced by 2 and the controls jumps back to l1 this is represented by this transition else if x is odd the value of x is reduced by 1 and the controls jumps back to l1 this is given by this transition.

This is the program graph representing this simple while loop. To see the specific behavior of the program starting from a certain value of the variable we can look at the transition system corresponding to the program graph and this starting value of the variable. For example, let us take the initial value of x to be equal to 3. The transition system with the initial condition x equal to 3 is as follows. The program is initially in location l1 with x equal to 3. since x is bigger than 0 this transition is taken and the program goes to location l2 with the same value of the variable.

At l2 since 3 is odd this transition is taken. Hence the next state would be l1 with x equal to 2. Once again the while loop is entered since 2 is even this transition is taken and hence the next state would be l1 with x equal to 0. Now as x equal to 0 the program does not enter the while loop instead it goes to l3.

(Refer Slide Time: 16:56)



This brings us to the end of this module. In this module we considered programs that manipulate variables. There are two notions that you need to be understand one program graphs these are representations of the control flow of the program. Two transition system of program graphs these represent the behavior of the program on a certain initial valuation of the variables. If these two notions are clear you are ready to jump to the next module