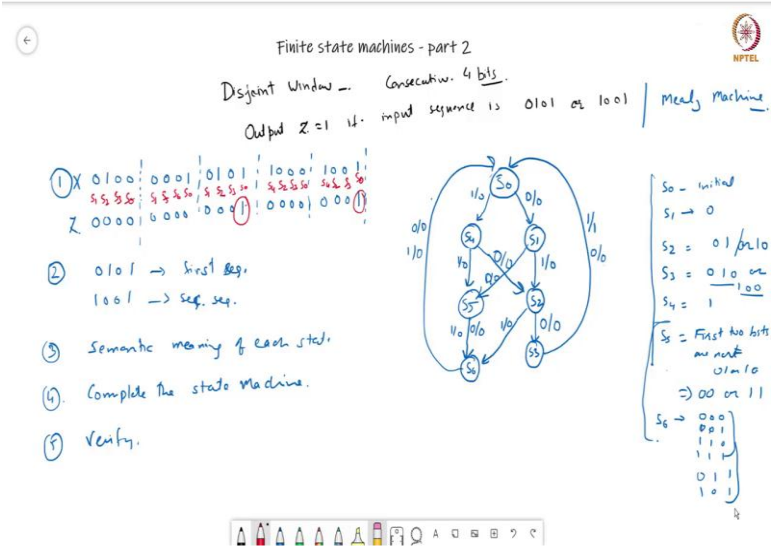
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology Ropar
Lecture 53
Sequence Detector: Example 2

(Refer Slide Time 00:17)



Now, let us work on one more problem. After understanding the basic structure of our examples and how to create state machines, let us construct one more state machine. So far, whatever problems we have been considering, they were all overlapping as well as continuous windows. Now, we will talk about one problem where we are talking about, it is non- sliding window. It is, actually disjoint window programs which now we will try to solve.

So far, all the problems were sliding window as well as overlapping patterns. Now, because the window is going to be disjoint. Their consecutive 4 bits which we have to match, there is no question of overlapping. So, the question is that, there are consecutive 4 bits and output would be 1 if the input sequence is 0 1 0 1 or 1 0 0 1.

So, to understand this problem, we will go to the step number 1. Step number 1 says that we have to give some input sequence. So, let us say the input sequence is 0 1 0 0. Because it is a disjoint, I will write a, maybe I will put a dotted like and then there is a 0 0 0 1 again another dotted line, then 0 1 0 1 another dotted line, 1 0 0 0 another dotted line, 1 0 0 1 another dotted line.

So, this is my input sequence x and what is the output corresponding to that? Because this is not matching, all the output is going to be 0. Here again, all the outputs are going to be 0, here 0 1 0 1, again, yeah, it is matching, so, 0 0 0 but only when the fourth bit will come then the output is going to be 1. Here also, the pattern is not matching, the output is 0. Here, 1 0 0 1, yes, pattern is matching so, the output would be 1 only in the last bit.

So, yes. The machine, because my output would be 1 when my input is 1 so, it is going to be a Mealy machine. Now, let us try to draw the state graph. So, every time we are staring with the first bit so, this S0 state is going to be always the initial and starting state. So, for the second step, let us say, we will take this 0 1 0 1 as the first sequence. So, now, if input is 0, we will go to S1 state. If input is 1, we will go to, there is an arrow like this, so, we will go to S2 state, if the input is 1.

And then if input is 0, so, 0 is there, 1 is there, the next state would be S3. That means the input is 0. And now, let us say we have another state S4. So, because we will again go back to the starting state, so, basically the pattern would finish as soon as the fourth bit is there so, we can say that if output is 1, then we will, sorry. So, if the next input is 1, then the output is 1 and we are there again in the S0 state so, that the next pattern can start. So, we are finished with the first sequence and we have finished all the bits so, that the output is 1.

Now let us go with the second sequence. 1 0 0 1 is the second sequence. Now, in the second sequence, we see that if the input is 1, then certainly this is the state which is different, S4. S4 means 1 so, we will also, write the semantic meaning here. Now, the next input is 0 so, when it is 0, then you can see something which is common between here, so, we can still say that, so, if it is 0, then we move to the S2 state. So, with this, S2 state would mean that either it is 0 1 or it is 1 0, 1 0.

So, you see, this pattern is common. 0 1 and 1 0 is different pattern, after that 0 1 is the same pattern in both the cases so, we can reutilize this S2 state. So, S2 state will have either 0 1 or 1 0. Now, what would be the meaning of S3 state in that case? S3 state is either 0 1 0 or it is 1 0 0. So, after S3 state, if we are going back to S0, so, that means either we are matching 0 1 0 1, or 1 0 0 1. So, that mans this, this whole pattern could be matched with this S0, S1, S2, S3, S4, five states.

Now, let us try to go to the third step, where we are going to. We here, we also, know the meaning and semantic meaning of these states. So, step 3 is also, there, we know the semantic meaning of each state. So, actually this step 2 and step 3 are going in parallel. Now, the step 4 means we will complete the state machine.

So, when we are completing the state machine, now, let us look at all the possibilities. So, if we are there in S0, S0, S0, it is already complete because 0, we know the transition for both input 0 as well as 1. Now, if the state is S1 and the input is 0, then we have to move to some other state. So, this particular state would resemble that we have received two bits. But none of the bit is matching this pattern, so, that means we have to create another state, let us say S5. This S5 is, if the input is 0. Output is certainly going to be 0.

So, this S5 state would signify that the two bits, first two bits are not 0 1 or 1 0. So, that means either they are 1 1 or 0 0. So, 0 0, in the case of 0 0, it is this or in case of 1 1, it is going to be this. So, that means first two bits are not 0 1 0 so, that means either they are 0 0 or they are 1 1. This is the S5 state and similarly, this means that we are able to finish, this S1 state is complete because we know that what is the transition for 0 input, we know what is the transition for 1 input.

Similarly, S4 state is also, complete because we know what is the transition for 0 state and what is the transition for 1 state. Now, let us look at this. At S5, the input is 0 or 1, then either it is 0, then also output is going to be 0, 1 also it is 0. So, this has to be a new state. This S6 states means that the input could be 0 0 0 or 0 0 1 or 1 1 0 or 1 1 1.

So, essentially any, any 3 bit pattern which is not matching with these two patterns is S6 state. So, that means if the input here is 1, that also would mean that this is a S6 state. S6 state could be 0 1 1. So, this is my S2 state. So, 0 1 1 or 1 0 1. So, any of the pattern would mean S6 state which, the way we are trying to put it is that S6 is a new state which does not meet the pattern but it has 3 bits.

Now, what to complete the state machine here. If S6 would result, would be given any input 0, then also, output is 0, if it is given 1, then also, output is 0. Now, the only state which is not complete is S3. What would happen is S3 is given a 0 input. So, if it is a 0 input then also, the next state is going to be S0 but the output is going to be 0. So, with this, we are able to complete this particular example using the procedures which we have followed.

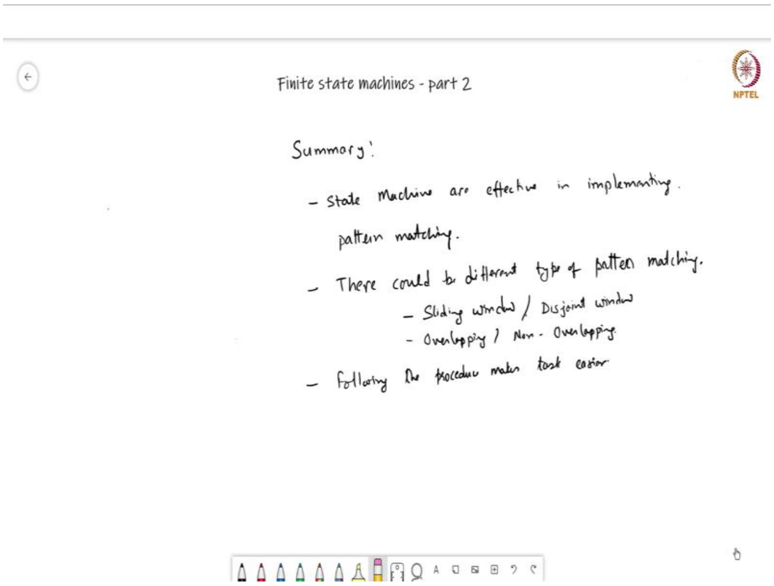
Now, we can go to the fifth step to verify whether we are correct or not. So, this verification, we can do using the same sequence. The whole idea here in this state machine is that we are essentially trying to find the semantic meaning here and we require S5 and S6 state specifically that because they are essentially a counting bits which are saying that whether two bits has elapsed or whether three bits has elapsed, three bits has been read. Although, none of them is correct.

So, this means that S6 is, that matching 3 bits but none of them is the correct pattern which we would like to meet. So, if you want to do verification, you can do it offline and we can quickly see for these five, six patterns. For this pattern, the first state is, let us write the state. This state is S1, is the first state and after that the next state here is 0 and then 1, it is S2. After S2, it became 0, it is S3. At S3, the output became 0 but the next state is again S0 because the input, the next input is 0.

So, from S0, again we have received 0, so, the next state is going to be S1 and then we have received 0, so, that means next state is S5. From S5, the next state is always S6 and from S6 also, the next state is always S0, the output is going to be 0. So, again, we have received 0, so, that means the next state is S1. From S1, we have seen the 1, so, that means the next state is S2. From S2, the input is 0, so, the next state is S3 and from S3, the next state is S0 but because input is 1, the output is 1.

So, this time, the next state from S0, the input is 1, the next state is S4 and after S4, we have received 0, so, the next state is S2. From S2, the next state is 0, so, it is, next input is 0, so, the next state is S3, and from S3, the input is 0 so, the next state is S0 but output is 0. Here also, we are following the same pattern. So, from S4, because the input is 1 and from S4, we are going to S2 and from S2, we are going to S3 and from S3 we are again going back to S0 and the output is 1. So, this verified that our input, this particular state diagram is correct.

(Refer Slide Time 15:13)



Now, we can summarize today's lecture. So, in summary, we have seen with couple of examples that these state machines are quite effective in implementing pattern matching and specifically when input is coming as a serial or a single bit. So, when input is used bit by bit then the state machines can be very effective in matching the pattern.

And for pattern matching also, we have seen with different examples that there could be two types of patterns. We have to specifically look in the problem statement. Rather, the pattern is sliding window or disjoint window. Usually when it is not specified, it is always sliding window but if it is disjoint window, then it would be specified in the question very specifically that it is a disjoint window.

And by default, we can assume that it is a, the pattern could be overlapping but it could be non-overlapping as well. So, non-overlapping means that once a pattern has finished, then that pattern cannot be a part of the other pattern.

So, the other thing which we have observed with today's example that if we follow a systematic approach, if we follow all these 5 steps, step 1, that writing the example sequence and then step 2, we start with one of the sequence and the other sequence so, that the result is 1 and then the step, along with that we also, have the semantic meaning of each and every state and then the next step is that we complete this state machine, we try to have all the possible transitions which are there. And the last step is definitely more important that we have to verify whether things are correct or not. So, with this, I would like to close this lecture. Thank you very much.