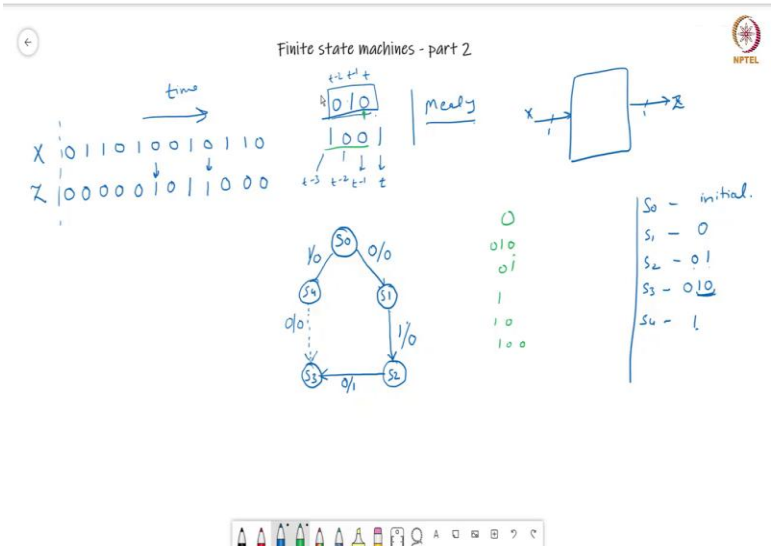
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology, Ropar
Lecture: 01
Sequence Detector: Example 1

Hello, everyone. Today we will continue our discussion on finite state machines and we will take little more sophisticated examples.

(Refer Slide Time 00:30)



Let us assume, the problem we are taking is here, we would like to find two streams. If our number is matching 0 1 0, then we will return 1 or if it is matching 1 0 0 1 then we will return 1.

The machine which we are going to design is a Melay machine. That means that when the output would depend on the input. So when the input will change the output will also change.

So now, how do we start? Like our previous question. The input is a single bit x and output is also single bit z. Now, this input is being given as a serial input. This serial input, as soon as the sequence is 0 1 0, we should get z as output as 0, output as 1. And if the sequence is a, previous inputs are 1 0 0 1, then the output should result in 1. Otherwise my output should be 0.

So, how should we start? First step to understand these problems or to solve these problems should be that we can take a example sequence so that we understand the cases,

what could arise, whenever we are solving them. So let us try to write some sequence x. So let us say this is my x 0 1 1 0 1 0 0 1 0 1 1 0. Now, this should be my z. You remember that first we have to find an initialization condition. Initialization condition, let us assume that before the start of the sequence there was a 0 here.

So, keeping that in mind, let us see if it was 0 then here, we have just started so it would result in 0, this would result in 0, this will also 0, 0, 1, 1, does not match any of the sequence so will be 0.

Now, 1 0 is again not matching anything but yeah, it would result in 0. 1 0 1, again, it should result in 0. But when the sequence is 0 1 0, then here my output should be 1. Now, the next one, as soon as the next input is 0 1 0 0, it is still not matching anything so the result is 0. 1 0 0 1, the result should be 1.

Again, 0 1 0 at this point, output should again be 1 because it is matching this sequence. At this point 1 0 1, it is not matching anything so the putout is 0, here also it is 0 and here also it is 0.

So with this, we have some understanding that how my sequence should behave like. Also, let us understand that as we are going in this direction, time is increasing. So this particular 0 is available at time 0, this is available at time 1, this is available at time 2. At any point of time we are looking in the past that if the sequence is, so when we are saying that we have to match 0 1 0, so basically this should be T this should be at T minus 1, this should be at T minus 2.

Similarly, if we would like to match this at time T, then 1 should be there at T and 0 should be there at T minus 1, this 0 should be there are T minus 2, this is there at T minus 3. What is 1 here? 1 is the clock period.

So because all of these state machines, we are designing as a synchronous state machine so when with the change of the clock period, after one clock period, the input is supposed to changed and then we should, we are supposed to see the output.

So, how should we start? To start this, let us try to derive a state machine so then we should like to derive how should we start? We always, it would be good to start with an initial state. Let us call this as S0. So S0 is just an initial state which also somehow

represents this dotted line which means that we do not know what has happened before that. So it could be a reset state, it could be an initial state.

And there, if something, so we always have to start with some state which is an M T or would be an initial state. Now, then what should be the next step? There are two sequences which we have to match. So what we can do is, we can initially start with one of it and then we can take on the second one. So let us take the first, let us take first this particular string.

Now, remember, we are going to design a Melay machine. So S0 is my initial state. I will write here what is my S0. S0 is my initial state. Now, if there a, input is 0, let us say input is 0, then I should move to the next state. Why we are moving to the next state? Because we are, we have to remember that there is one 0 here.

Now, what should be the output? Because there is only one 0 which has matched so the output is 0. Now, after that, let us say the next input is 1. The input is 1. We should again move to the next state S2. Why? Because this is unique. This is different that 0 and 1 so the output should again be 0. There is an arrow here. The transition represents that there is a 1 is the input and 0 is the output. S2 is a distinct state.

So I can write here also S1 means that there, we have found at least 0. And S2 says that 0 as well as 1 is there. So this is my interpretation of my states which may also keep in varying as we are moving. So this interpretation will always help us to remember that what is the significance of any particular state.

Now, after that if I receive another 0. So that means 0 1 0. So if I receive another 0. The question would be, where should I move to? For now, let us. let us keep another state. For now means that we may see that in future do we require that particular state or not but we will keep that state in our mind.

So here, let us have this as S3. S3 means that there is, if there is a 0 here, the output is going to be 1. Now, what does this S3 mean? S3 could mean multiple things. It could mean 0 1 0. Or it could also mean that it is 1 0. So, here we are sure that the output is going to be this. 1. Because we have matched this particular sequence.

What would happen, now, shall we, so there are two possible ways to move forward. One possibility is that we will complete this state machine. Right now, this state machine is incomplete because we have not seen for this S0 state, what would be the transition if the input is 1, similarly for the S1 state, what would be the transition if input is 0. So State machine completion is 1 task which is remaining and the other thing is, we can start with the second sequence also.

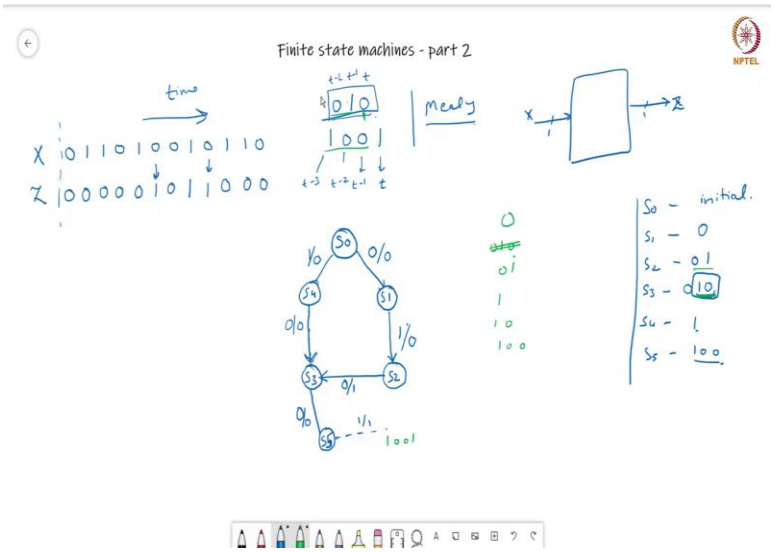
So let us take the second sequence first and then we will complete the state machine. Now, for the second sequence, we see, if it is 1, if the input is 1, then we have to have, let us say, it is the S3, I have already written, so let us create another state S4 which means that input is gone and output is going to be 0. So this S4 essentially means that there is only one which has been recognized. So this 1 is different than this 1 because this one says that there is a 0 preceding this but this 1 means that, there is no 0 preceding it.

So let us go back to the next one. If it is 0, now, the input is 0, the next input is 0. So one possibility is that we remember, we can say that let us keep it in a dot and then we can finalize that. Let us say that this is the next state. 0 and then 0. Are we sure? So make whether we are sure or not, then we have to say, whether S should represent 0 1 0 or should it represent only 1 0?

Now, to understand this, let us look at these input sequences and patterns which we would like to match. Here, what we would like to match is, we would like to say, the pattern which we would like to match here is, let us write that in green. So we would like to match 0, we would like to match… so this would be 1 if 1 0 1 0 is there. So that means S2 already represents that 01 is there and S3, if S3 is coming from S2, that means that 0 1 0 is there. Now, that is what we would like to remember.

Now, we would like to remember 0 0 1 and here, we would like to, for this pattern, we would like to remember 1, we would like to remember 1 0, we would like to remember 1 0 0. And at the time we are here at 1, this state, remembering this state is sufficient. Similarly when we are at this point, remembering this state is sufficient.

(Refer Slide Time 13:25)



So this may not be required. The only thing which we have to remember is this. So that means, this S3, even if we say this 1 0, this is sufficient. So that means, that also means that from S4, if there is a 0 input then we can say that the transition is towards S3. Then we are saying S3 means this only. And yeah, S4 is 1.

So, now, we have, let us come back to the sequence. If input is 1, from initial state if input is 1, we have reached S4 and from there, the next input is 0, from S4, we will go to, the transition would be towards S3.

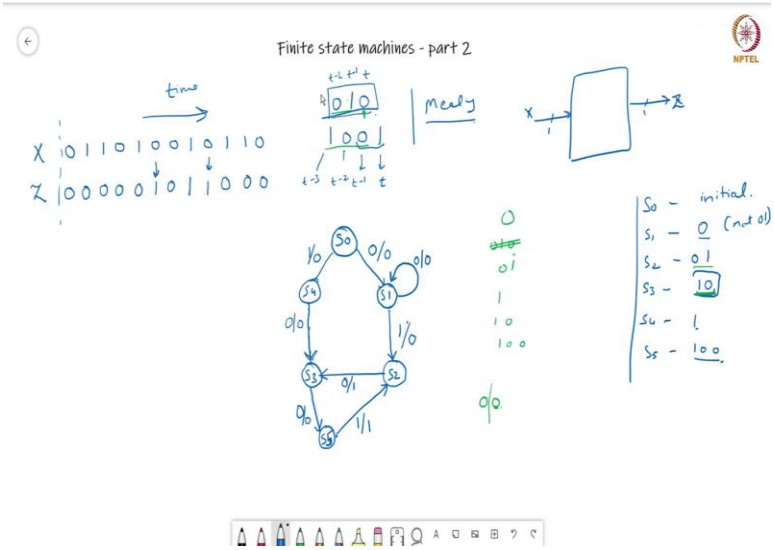
Now, at S3, if there is an input which is 0, now we have to create another state because we would like to remember it is 1 0 0, so we create another state which is S4. S4 is already there so we will create another state which is S5.

Now, the state is S5, and from S5, if it is 1, then we are clear that if it is 1, then we have to go to some state but the output is going to be 1, that we are sure. But to which state we should go?

Let us write it here also, my S5 means that 1 0 0 is, it remembers. So, out of these, if it is 1, then what is the most useful thing? We are there at S5. So from S5, we would like to see that. Now, the input is 1, so that means this, this 1 represents that the, the sequence is so far, 1 0 0 1.

Now, out of these 1 0 0 1, what is the most useful thing here? We can say that actually it is 0 1 which would be helpful in identifying any other pattern, the pattern which we are matching. So that means the next state is supposed to be S2.

(Refer Slide Time 16:03)



So let us make that, so if this is my arrow, this was my arrow. So from S5, if input is 1, then output is 1, the next state is S2. Why S2? Because it represents that the last, the last pattern is 0 1. Actually, this part. So that is why, we will move this state to S2.

So with this, we have matched both the patterns. It is still an incomplete state machine, let us try to complete the state machine. How do we complete the state machine, that every, at every state, we should know that for all the input combination, what should be the next state. So for S0, we know that if it is 0, then the state is S1, if S0, the input is 1, then we will go to S4 state. Similarly, where to find out for the other one?

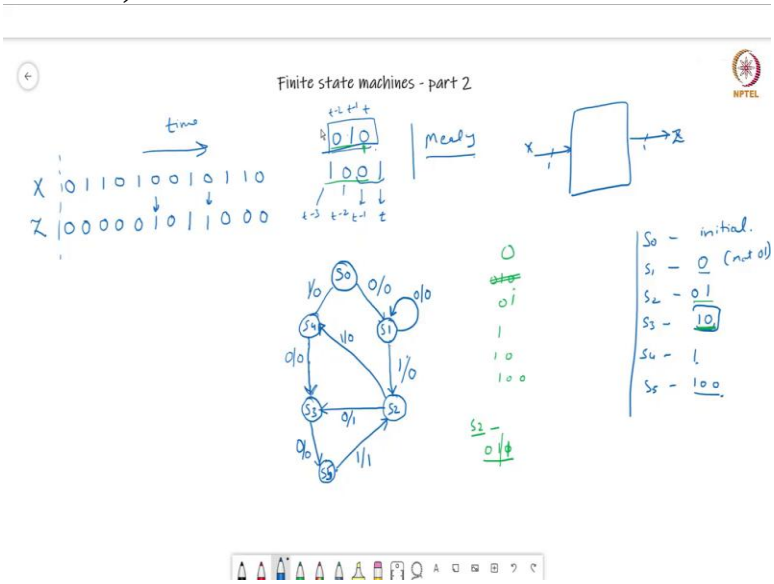
Now, one more thing we have to understand that this, whenever we are saying S0, S1, 0 which means that it is 0 but it is not 0 1. So, this and this states are distinguished.

And similarly this is the, so we can rub this part. Similarly when we are saying 1, 1 means it could be only single 1, anything preceding that we do not know. So, now, let us again look at the S1 state. From the S1 state we know that if it is the, if the input is 1, then we will go to S2 state but if the input is 0, what state we should go to?

So if input is 0, then it will be, we are talking about S1 state. If the input is 0, so that mans the sequence we have received is 0 0. So out of these 0 0, we see that the only thing meaningful we can receive is, we can discard this part, we can say it is still 0 so we can mark it as, it is back to S1 state, if it is 0. The output would be 0.

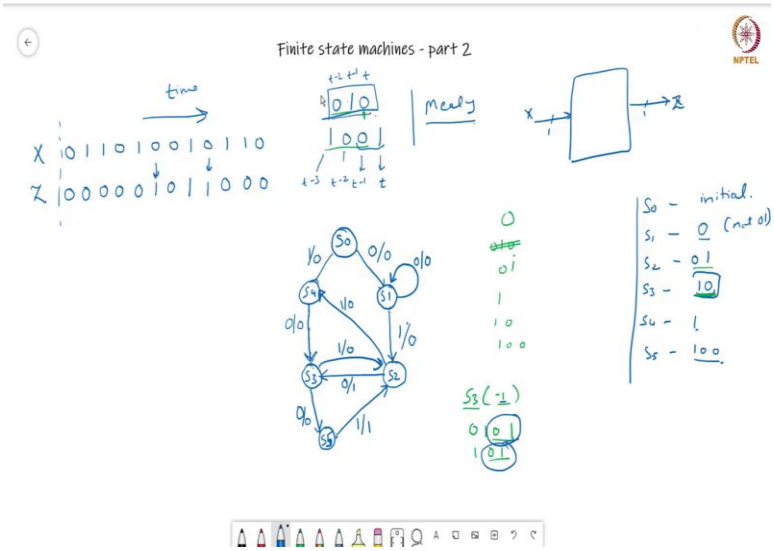
Now, let us look at S2 state. S2 state, we know that if input is 0, then it will go to S3 state but what would happen if the input is 1. If input is 1, what does it mean?

(Refer Slide Time 18:53)



If input is 1, this means that, we are talking about S2 state. At S2 state if the input is 0, the sequence it will become is 0 1 0 1, yeah, we are talking about 1, so, 0 1 1. So out of these 0 1 1, what would be the meaningful state? We can look at this sequence, or we can look at this graph itself. So if it is 0 1 1, then we are, this particular, any, any of the pattern has not matched for this particular pattern, any bit has not matched for this pattern and for this pattern, only the first bit has matched. So that means it may be a starting of the next sequence so we have to say if S2, at S2, the input is 1, then we are going to S4 state.

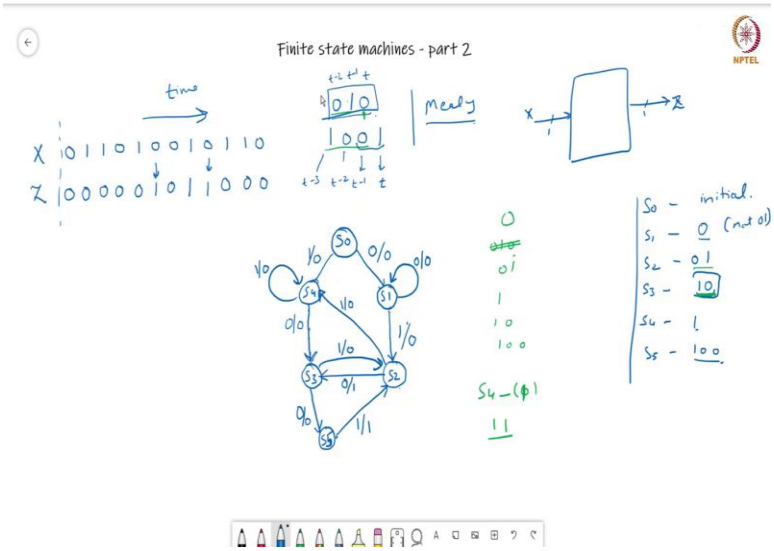
(Refer Slide Time 19:57)



Now, let us look at the S3 state. At S3 state, we know if input is 0, then we are going to S5 state but what if input is 1? So, at S3 state, we are talking about S3 state and S3 state my input is 1. If input is 1, the sequence, the possible sequences are, one sequence is 0 1 and 0 and then another 1. So this could be the sequence, I could have reached S3 because of this path, S0, S1, S2 and S3. The other path could have been S0, S4, S3, that means 1 0 1.

So, by looking at these states, the, if the input is, at S3 if the input is 1, then by looking at it, appears, this would be the next state. So I can say, if he input is 1, then the next state is S2, but the output is 0 because we have not completely matched any of the pattern, we are in the midway. So because of that, we say from S3 if the input is 1, the next state is S2. So this was about S3.

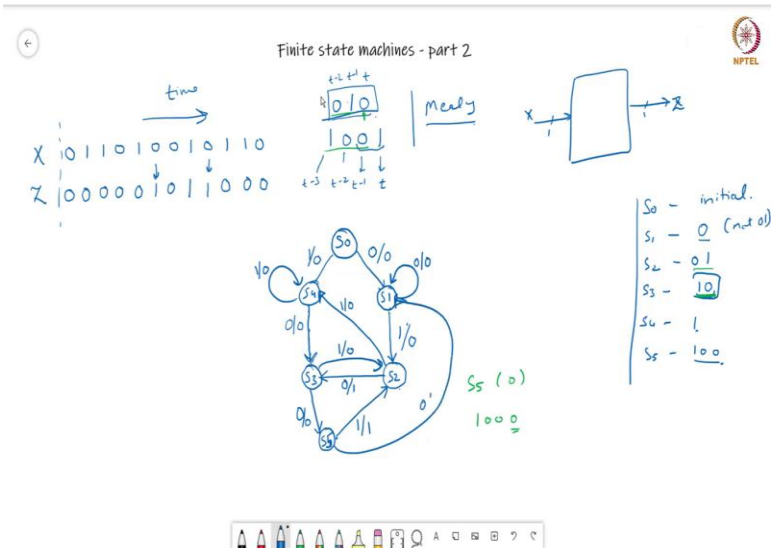
(Refer Slide Time 21:31)



Now, at S3 also our state machine is complete. Now, let us look at S4. At S4, we know what would be the state if the input is 0. Now, what would be the state if the input is 1? So at S4, if the input is 0, sorry, if input is 1, if input is 1, the pattern would be 1 1. Now, 1 1 is still, the starting of this particular pattern. So the state is going to be S4 again. If it is 1, then 0.

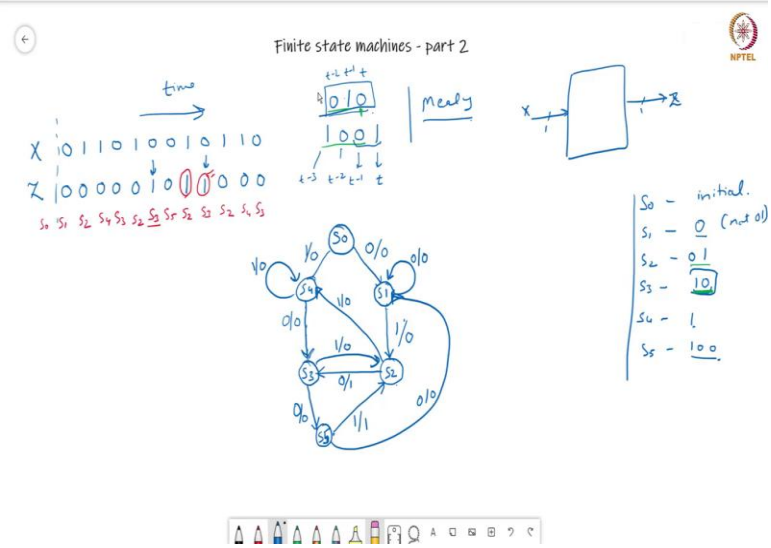
Now, S4 is done, S5 is done. S4 is done, S3 is done, now S5. For S5 if input is 1, we know it is going to be S2 but what would be the next state if the input is 0.

(Refer Slide Time 22:35)



So for S5, let us talk about S5. We can rub this also so at S5, input is 0. Now, if input is 0, that means that what was my S5, S5 was 1 0 0 and there is another zero. So this another 0 means that, this another 0 means that none of the pattern has matched. It could only be the starting match, first bit match of this particular pattern. So we, the next state has to be S1, which means that 0 and the output is 0.

(Refer Slide Time 23:17)



So, this way we have completed this particular state machine and yeah, we are done. Now, the next thing could be that, because it involves some sort of a judgment at every stage so even if our one particular judgment is wrong, there is a possibility that whole state machine construction could be wrong. So the best idea at this point of time would be to relook. Maybe at some particular input sequence we see that whether are good, whether we are correct or not.

So, we will again fall back to this example, the first example. So let us look at this input sequence and we see whether we are able to correctly do the state transitions or not, whether we are able to match out output or not. Or in what state we are there. So let us try to see. When we are, we will try to write that in which state we are there. So when input is 0, this is the initial state machine so if input is 0, then we are there in the S1 state, S0 state was the before this red line.

So if my input is 0, then we will go in S1 state and now, input is 1. The next input is 1, because next input is 1, we will go to S2 state. Good. And after that next state is again 1.

So the next state is 1, so we are there in S4 state. After S4 state, the input is 0. Because input is 0, so we will go to S3 state. And after S3 state the input is 1. So from S3, my input is 1, so we are there in S2 state and then the input is 0, so from S2 state, my input is 0, so we are there in the S3 state which is correct so the output is 1.

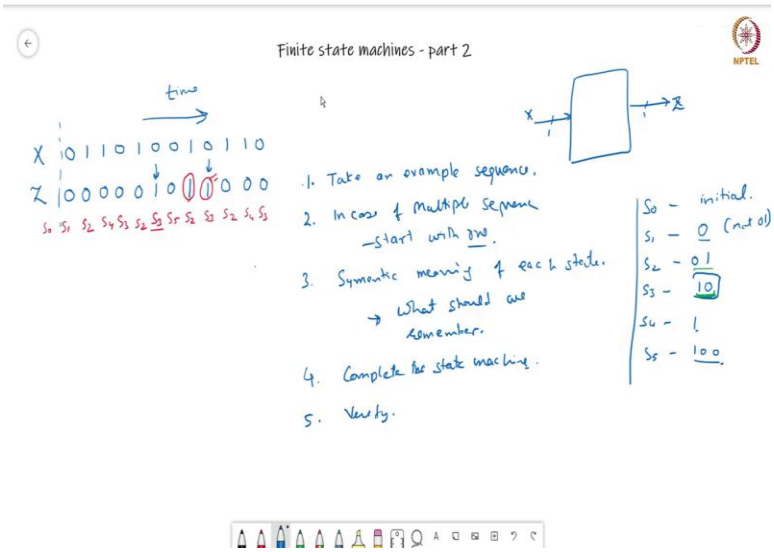
Now, from the S3 state, we have received another 0. So from there we have received S3, we have received another 0 so we are there in the S5 state. From S5 state, we have received 1, from S5 state we have received 1, so we are back in the S2 state. And we are done because from S5 to S2, we are receiving so there is 1 as output which looks good and from S2 state, we have received further 0. So from S2, we have received further 0 so that means we are in the S3 state. And there is a 1 output from S2 so S3 if the input is 0, then output is 1, so this 1 is also correct.

And then from 0, we have 1 input, from S3, the input is 1. That means we are back in the S2 state. And after that there is another 1, so from S2, we have received another 1, we are there in the S4 state, the output is 0. From S4 state, the next input is 0, that means we are in the S3 state.

So this way, we are able to fool proof and we are able to check whether our state machine is correct or not. This is kind of a re correction or re checking method that we should perform.

Now, by looking at this example which was at least most complex than the example which we have taken in last lecture, now, we can also summarize that what are the steps. If we, if we come across any particular sequence, any particular example, what should be the steps which we should follow, which we should follow whenever we are constructing the state machines.

(Refer Slide Time 27:38)



So, let us summarize that what would be the steps. So the first step I would say is, so the first step is, take an example. Take an example sequence like this. So if we have taken an example sequence, then it would always help us. It is helping us in two ways. One, it is, we are clearer in our mind that how should my output look like.

The second thing is that whenever we need to look at any particular sequence, we can look at this particular example and we can see that what should be our third state. The third thing is that when your state machine construction is finished then also you can use the same example to verify whether everything was correct or not.

So this is one. The second thing is that, the second thing is that we can start with one sequence. So if there are multiple sequence, in case of multiple sequence, start with one, any one. It does not matter which one. So if we start with any of the sequence then we can build on.

The third thing is, it is good to create such kind of a table. So this table. So this table is very helpful. So if you use this particular table, try to have some semantic meaning o each state. So if we try to have semantic meaning. So this semantic meaning also, there is also one corollary here that first of all, you have to try to find out the semantic meaning also by looking at the sequence we should also see that what should be remember.

So what should we remember? If we need to remember all the bits, what bits are important to be remembered? Here for example, we have seen that 0 1 0 was not

important but 1 0 was the only information which was important. So what information is important to be remembered, that is what is maybe the semantic meaning of the each, semantic meaning of each step.

So once it is done the fourth step along with the direction is, whenever we are taking, starting with one sequence, and we will take it to the conclusion, we will say that till that output is 1, then we can start with the second sequence. So, one of the next steps would be then, complete the state machine.

Complete the state machine means that for every state, for every input we should know that what would be the next state transition. So, and, the last step is once the state machine is complete, we should verify. So if we are following all these four, five steps, then there should not be any problem, we should be able to create finite state machines.

Now, there is one question which arises in our mind that what would happen if, the states we are creating, could be, could be redundant or could be more than what is expected? Yes, so this is an important problem which we will try to address to in one of the next lectures. So there is a good possibility that we may be redundant in adding the states. We think that okay, this information is important, this is required but that information may not be required as such.

So there must be a method so that we are able to remove those redundant states, we are able to, yeah, so we are able to reduce the state machines, if the number of states are more the desirable states. So with this, this particular topic is finished and now we understand that how to create finite state machines given a more complex problem.