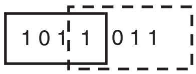
Solved Problems

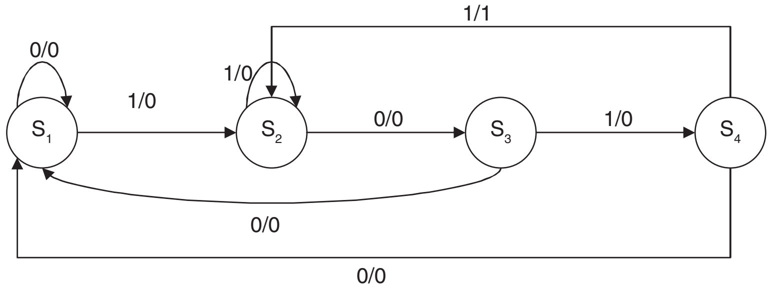
1. Design a two input two output sequence detector which generates an output '1' every time the sequence 1011 is detected. And for all other cases, output '0' is generated. Overlapping sequences are also counted. Draw only the state table and the state diagram.

Solution: The sequence is 1011. We have to start from S_1. If we get input 0, then there is no chance to get 1011, so it is confined in S_1 producing output 0. If we get input 1, then there is a chance to get 1011, and so the control moves to S_2 producing output 0 (as we have not got 1011 as input still). In S_2, if we get 0, then there is a chance to get 1011, and so the control moves to S_3 producing output 0. In S_2, if we get input 1, then there is a chance to get 1011, considering the last 1. So, the control will be confined in S_2, producing output 0.

In S_3, if we get input 0, then there is no chance to get 1011. In this case, we have to start again from the beginning, i.e., from S_1. So, the control moves to S1 producing output 0. In the state S_3, if we get input 0, then there is no chance to get 1011 considering any of the fourth or third and fourth or second, third and fourth or first, second, third, and fourth input combination. In this case, we have to start again from the beginning, i.e., from S_1. So, the control moves to S_1 producing output 0. If we get input 1, then the string 1011 is achieved, and so the output 1 is produced. As the overlapping sequence is also accepted, the control moves to S_2 so that by getting 011, the sequence detector can produce 1.



The state diagram is given.

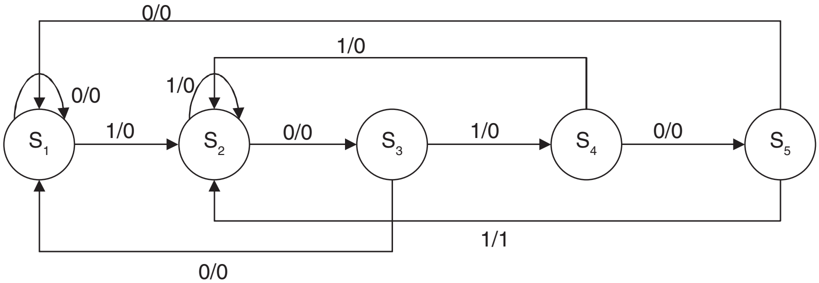


The state table for the sequence detector is

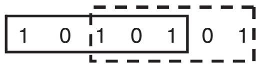
Present State	Next State, O/P	
	X = 0	= 1
S_1	S_1, 0	S_2, 0
S_2	S_3, 0	S_2, 0
S_3	S_1, 0	S_4, 0
S_4	S_1, 0	S_2, 1

2. Design a two input two output sequence detector which generates an output '1' every time the sequence 10101 is detected. And for all other cases, output '0' is generated. Overlapping sequences are also counted. Draw only the state table and the state diagram and make the state assignment.

Solution:



The state diagram is constructed previously.
Here, the overlapping sequence is 101.



The state table for the sequence detector is

Present State	Next State, O/P	
	X = 0	= 1
S_1	S_1, 0	S_2, 0
S_2	S_3, 0	S_2, 0
S_3	S_4, 0	S_1, 0
S_4	S_5, 0	S_2, 0
S_5	S_1, 0	S_2, 1

For doing the state assignment, we have to assign the states to some binary numbers. Here are five states, so we have to take a binary string of length three because $2^3 = 8 > 5$.

Let us assign 000 as S_1, 001 as S_2, 011 as S_3, 010 as S_4, and 100 as S_5.

After doing the state assignment, the state table becomes

Present State	Next State, (Y_1Y_2)		O/P (z)	
(y_2y_1)	X = 0	= 1	= 0	= 1
000	000	001	0	0
011	011	001	0	0
011	010	000	0	0
010	100	001	0	0
100	000	001	0	1

3. Find the equivalent partition for the following machine.

Present State	Next State, z X = 0	X = 1
A	B, 0	C, 1
B	A, 0	E, 1
C	D, 1	E, 1
D	E, 1	B, 1
E	D, 1	B, 1

Solution: For a string length 0 (i.e., for no input), there is no output. So, all the states are equivalent. It is called 0-equivalent. We can write

$$P_0 = (ABCDE).$$

For string length 1, there are two types of inputs—0 and 1. The states A and B give output 0 for input 0 and states C, D, and E give output 1 for input 0. All of the states give output 1 for input 1.

So, the states in the set P_0 are divided into (AB) and (C, D, E). We can write

$$P_1 = ((AB)\ (CDE)).$$

Here A and E are 1-distinguishable because they produce different outputs for input string length 1.

For input string length 2, check the distinguishability by the next state combination.

The states A and B for input 0 produce next states B and A, respectively, and produce next states C and E for input 1. B and A belong to same set, and C and E also belong to the same set. So, AB cannot be partitioned for the input string length 2.

The states CDE for input 0 produce next states D, E, D, respectively, and produce next states E, B, B for input 1. D, E belong to same set but B and E belong to different sets. So, the set (CDE) is portioned into (C) and (DE). The new partition becomes

$$P_2 = ((AB)\ ((C)\ (DE))).$$

For input string length 3, we have to perform the same as like string length 2. The states A, B for input 0 produce next states B and A, respectively, and produce next states C and E for input 1. B, A belong to same set, but C and E belong to different sets. So, the set (AB) is partitioned into (A) and (B).

C is a single state and cannot be partitioned further.

The states D and E produce next states E and D, respectively, for input 0 and produce next state B for each of the states for input 1. D and E belong to same set. So, D and E cannot be divided. The new partition becomes

$$P_3 = (A)(B)(C)(DE).$$

Next, we have to check for input string length 4. Three subsets contain single state—A, B, and C, and cannot be partitioned further.

The states D and E produce the next states E and D, respectively, for input 0 and the next state B for each of the states for input 1. D and E belong to same set, and so D and E cannot be partitioned further. The partition is the same as P_3. So, the partition P_3 is the equivalent partition of the machine.

Minimization: We know that the equivalent partition is unique. So, P3 = (A)(B)(C)(DE) is the unique combination. Here, every single set represents one state of the minimized machine.

Let us rename these partitions for simplification.
Rename (A) as S_1, (B) as S_2, (C) as S_3 and (DE) as S_4.
The minimized machine becomes

Present State	Next State, z	
	X = 0	X = 1
S_1(A)	S_2, 0	S_3, 1
S_2(B)	S_1, 0	S_4, 1
S_3(C)	S_4, 1	S_4, 1
S_4(DE)	S_4, 1	S_2, 1

4. Find the equivalent partition of the following machine.

Present State	Next State, z	
	X = 0	X = 1
A	B, 0	D, 1
B	D, 1	F, 1
C	F, 1	B, 1
D	F, 0	A, 1
E	C, 0	A, 1
F	C, 1	B, 1

Solution: The partitions are

P_0 = (ABCDEF)
P_1 = (ADE)(BCF) (Depending on o/p for i/p 0)
P_2 = (ADE)(B)(CF) (For i/p 0, the next state of B goes to another set)
P_3 = (A)(DE)(B)(CF) (For i/p 0, the next state of A goes to another set)
P_4 = (A)(DE)(B)(CF)

As P_3 and P_4 are the same, P_3 is the equivalent partition.

Minimization: We know that the equivalent partition is unique. So, P_4 = (A)(DE)(B)(CF) is the unique combination. Here, every single set represents one state of the minimized machine.

Let us rename these partitions for simplification.
Rename (A) as S_1, (B) as S_2, (DE) as S_3, and (CF) as S_4.
The minimized machine becomes

Present State	Next State, z	
	X = 0	X = 1
S_1(A)	S_2, 0	S_3, 1
S_2(B)	S_3, 1	S_4, 1
S_3(DE)	S_4, 0	S_1, 1
S_4(CF)	S_4, 1	S_2, 1