Compiler Construction

Finite State Machines with Output

- Mealy Machine
- Moore Machine

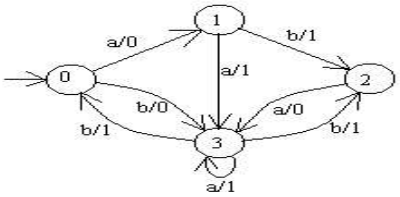
Mealy Machine & Moore Machine

- Finite automata are like computers in that they receive input and process the input by changing states. The only output that we have seen finite automata produce so far is a yes/no at the end of processing.
- We will now look at two models of finite automata that produce more output than a yes/no.

Mealy Machine

Transitions are labelled *i/o* where

- *i* is a character in the input alphabet and
- *o* is a character in the output alphabet.

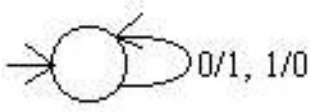



```
Input:  aaabb
Output: 01110
```

- Mealy machines are complete in the sense that there is a transition for each character in the input alphabet leaving every state.
- There are no accept states in a Mealy machine because it is not a language recogniser, it is an output producer. Its output will be the same length as its input.

Mealy Machine

- Mealy Machines are exactly as powerful as Moore machines
  - (we can implement any Mealy machine using a Moore machine, and vice versa).
- However, Mealy machines move the output function from the state to the transition. This turns out to be easier to deal with in practice, making Mealy machines more practical.




```
Input:  010110
Output: 101001
```

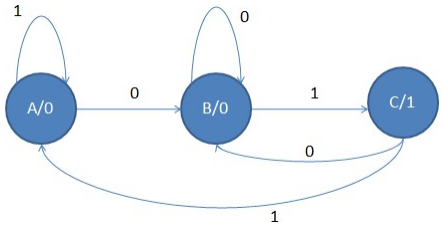
The above Mealy machine takes the one's complement of its binary input. In other words, it flips each digit from a 0 to a 1 or from a 1 to a 0.

Mealy Machine & Moore Machine

- Moore Machine M1=(Q,∑,λ,O,δ,qo)
  - ✓Q: Set of States
  - ✓∑: input alphabet
  - ✓O: Output alphabet
  - ✓λ : Output function, Q→O
  - ✓δ : Transition Function, Qx ∑ → Q
  - ✓qo: Initial State

- Mealy Machine M1=(Q,∑,O, λ',Δ,qo)
  - ✓Q: Set of States
  - ✓∑: input alphabet
  - ✓O: Output alphabet
  - ✓Δ : Transition Function, Qx ∑ → Q
  - ✓λ': Output function, Qx ∑ →O
  - ✓qo: Initial State

Mealy Machine (Count Substring 01)



λ'(C,0)= λ(δ(C,0))
= λ(B)
=0

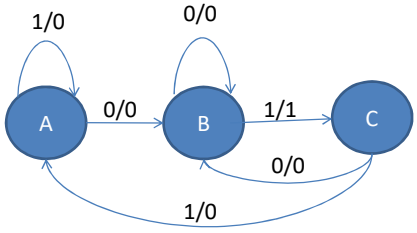
λ'(C,1)= λ(δ(C,1))
= λ(A)
=0

λ'(A,0)=λ(δ(A,0))
= λ(B)
=0

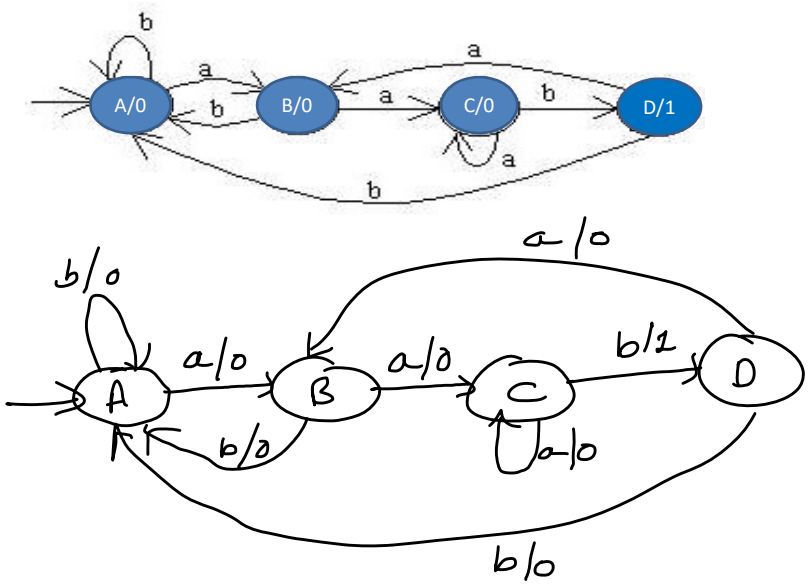
λ'(B,0)= λ(δ(B,0))
= λ(B)
=0

λ'(A,1)= λ(δ(A,1))
= λ(A)
=0

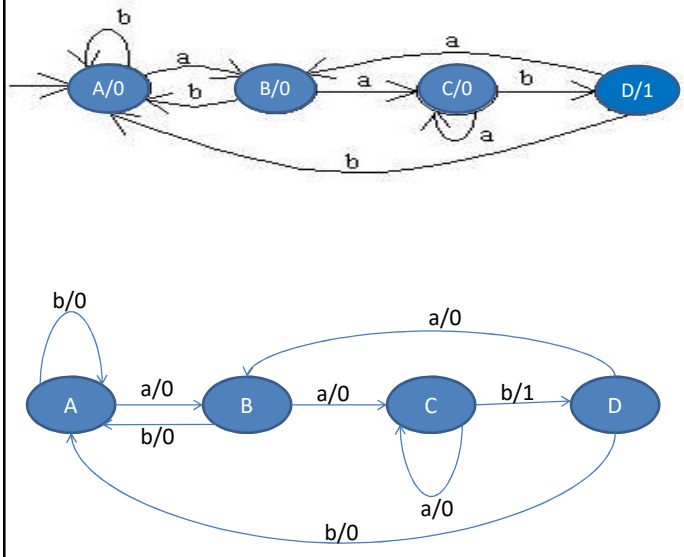
λ'(B,1)= λ(δ(B,1))
= λ(C)
=1



Mealy Machine (Count Substrings aab)

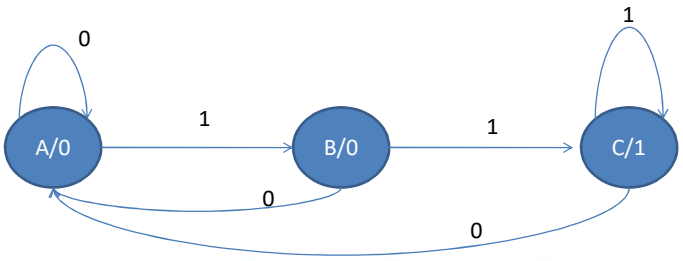


Mealy Machine (Count Substrings aab)



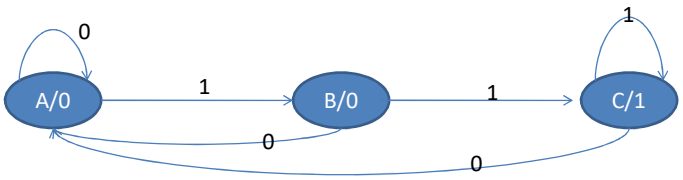
Moore Machine To Mealy Machine

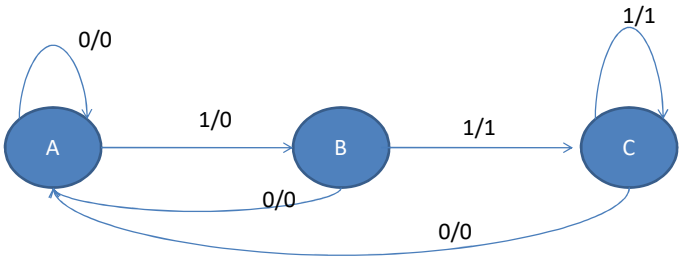
- Replaces the first 1 with 0 from every substring starting with 1



Input	Result
0 1 1 1 0 0 1	0 0 1 1 0 0 0
0 0 0 1 0 0 1 1 1 0	0 0 0 0 0 0 0 1 1 0
1 1 1 0 1 0 1 1 0	0 1 1 0 0 0 0 1 0

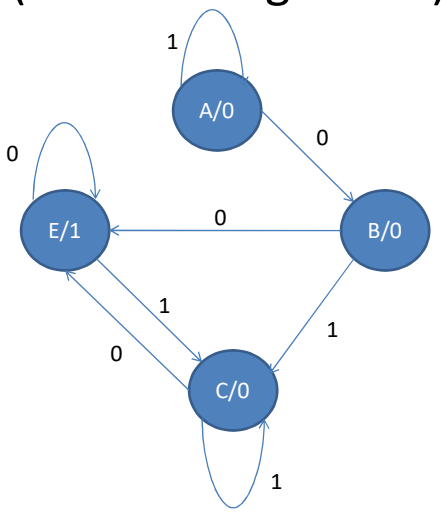
Moore Machine To Mealy Machine





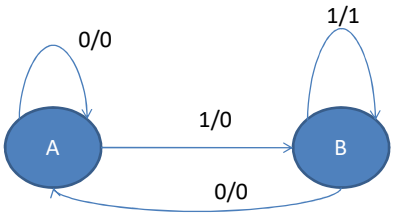
Input	Result
0 1 1 1 0 0 1	0 0 1 1 0 0 0
0 0 0 1 0 0 1 1 1 0	0 0 0 0 0 0 0 1 1 0
1 1 1 0 1 0 1 1 0	0 1 1 0 0 0 0 1 0

Moore Machine (Count strings 01*0)



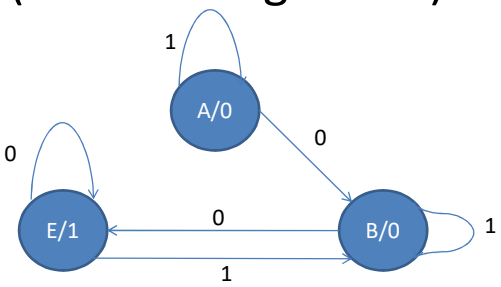
Moore Machine To Mealy Machine

- Replaces the first 1 with 0 from every substring starting with 1

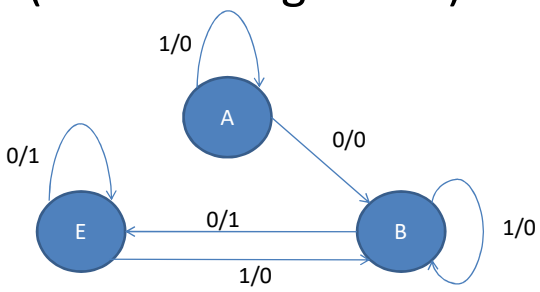


Input	Result
0111001	0011000
00010011110	00000001110
11101011 0	01100001 0

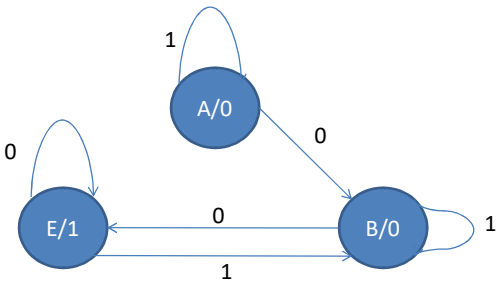
Moore Machine
(Count strings 01*0)



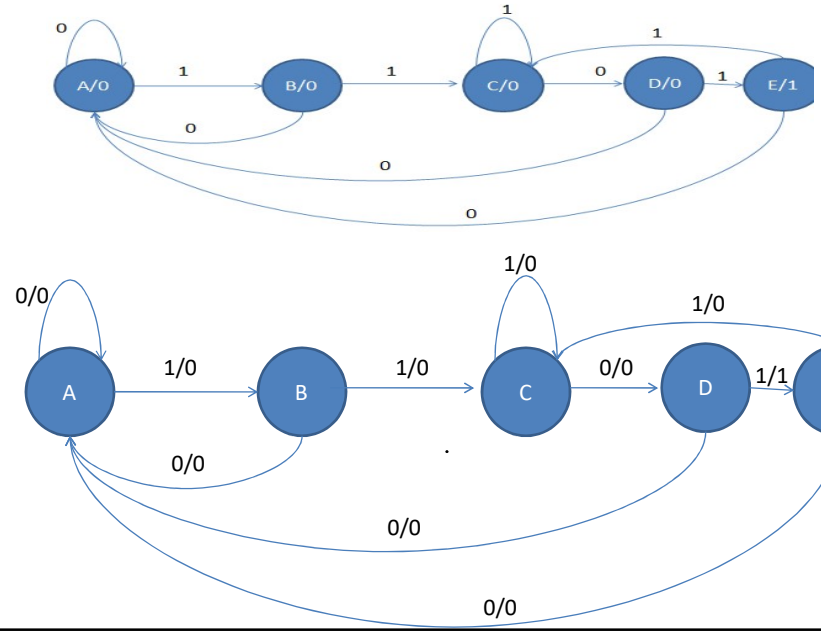
Moore Machine To Mealy Machine
(Count strings 01*0)



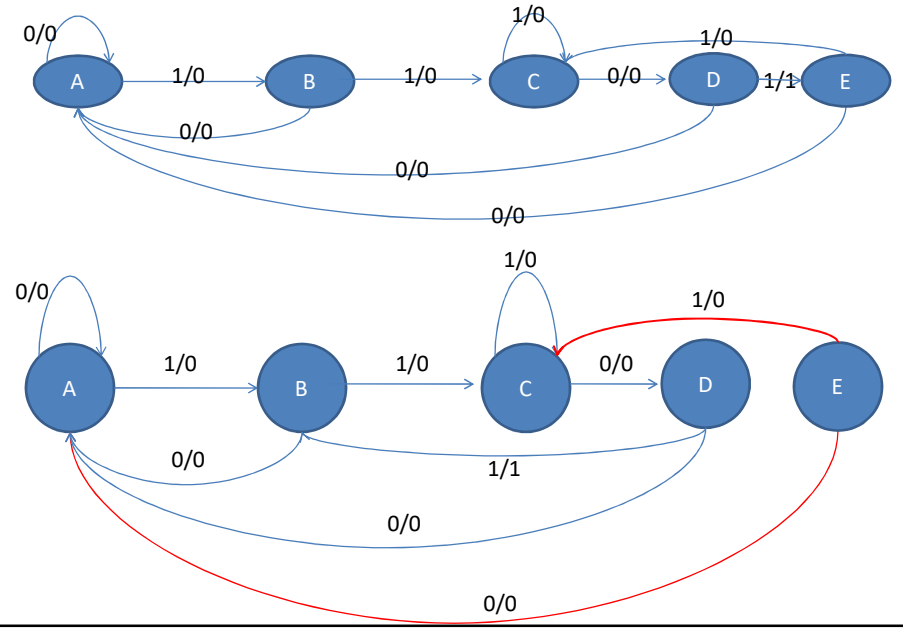
State : ABEBEBB
Input: 0010110
Output:0101001



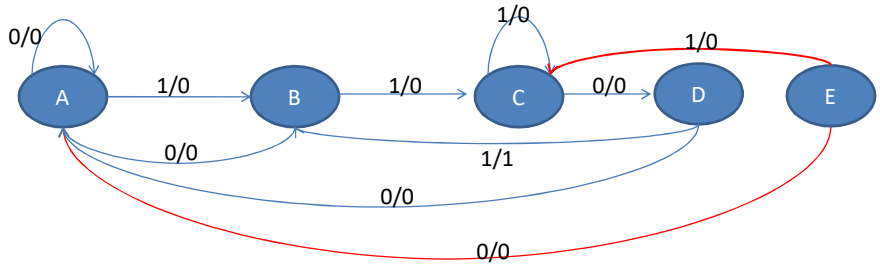
Moore Machine To Mealy Machine
(Count the occurrence of 1101)

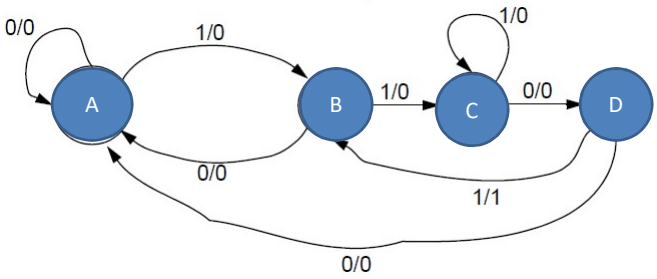


Moore Machine To Mealy Machine
(Count the occurrence of 1101)

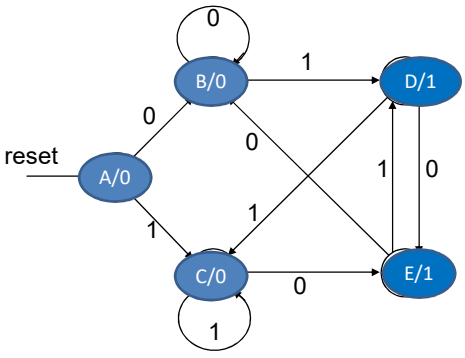


Moore To Mealy Machine
(Count the occurrence of 1101)



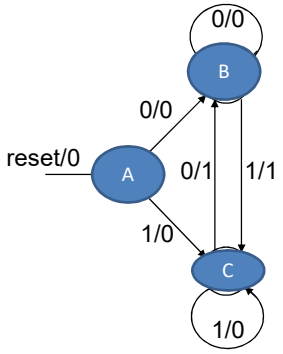


Moore Machine
(Count strings 01 or 10)

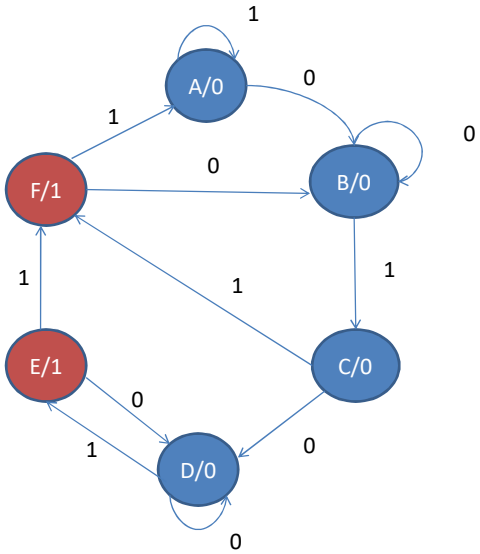


Input: 00**01101**111**10010**0
State ABBB**D**C**ED**CCC**E**B**DE**B
Output: 00001011000010110

Mealy Machine
(Count strings 01 or 10)



Moore Machine
(Count strings 010*1)



Mealy Machine
(Count strings 010*1)