Finite Automata with Output

Moore = Melay

- So far, we have define that two machines are equivalent if they accept the same language.
- In this sense, we cannot compare a Mealy machine and a Moore machine because they are not language definers.

Definition:

- Given the Mealy machine *Me* and the Moore machine *Mo* (which prints the automatic start state character x), we say that these two machines are **equivalent** if for every input string, the output string from *Mo* is exactly x concatenated with the output string from *Me*.

Theorem 8

If Mo **is a Moore machine, then there is a Mealy machine** Me **that is equivalent to** Mo**.**

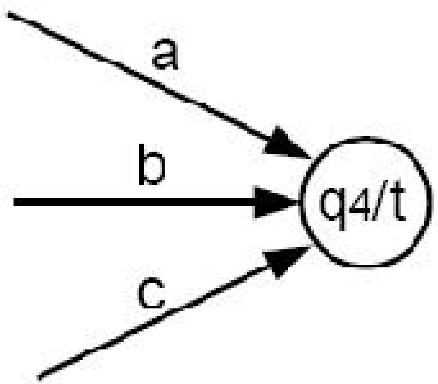
Proof by constructive algorithm:

- Consider a particular state in *Mo*, say state q_4, which prints a certain character, say *t*.

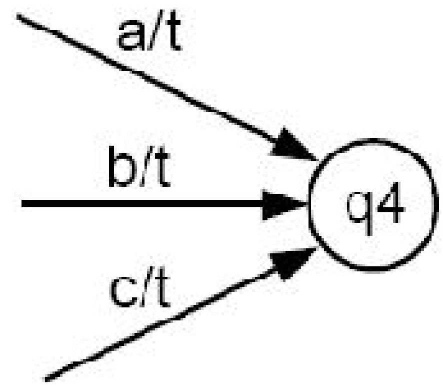
- Consider all the incoming edges to q_4. Suppose these edges are labeled with a, b, c, ...

- Let us re-label these edges as a/t, b/t, c/t, ... and let us erase the t from inside the state q_4. This means that we shall be printing a t on the incoming edges before we enter q_4.

Proof by constructive algorithm contd.



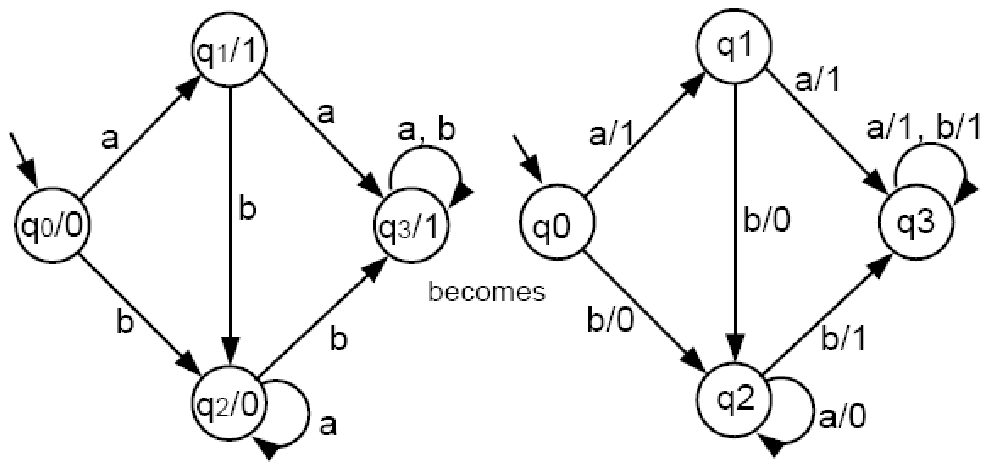
becomes



- We leave the outgoing edges from q_4 alone. They will be relabeled to print the character associated with the state to which they lead.
- If we repeat this procedure for every state q_0, q_1, ..., we turn *Mo* into its equivalent *Me.*

Example

- Following the above algorithm, we convert a Moore machine into a Mealy machine as follows:

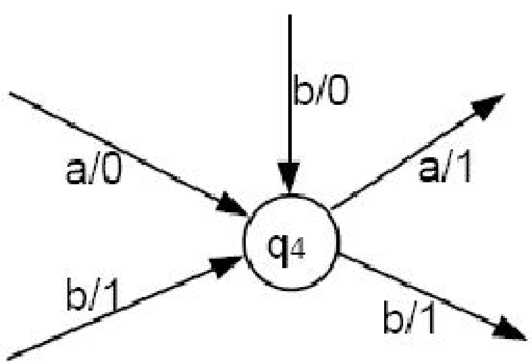


Theorem 9

For every Mealy machine Me**, there is a Moore machine** Mo **that is equivalent to it.**

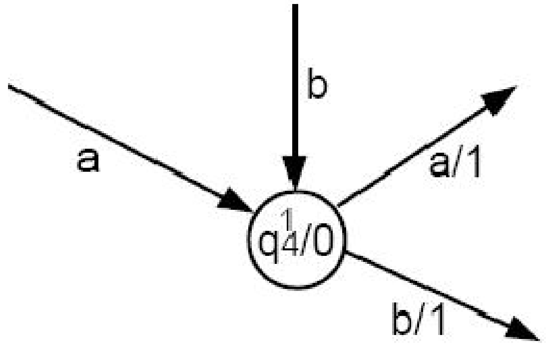
Proof by constructive algorithm:

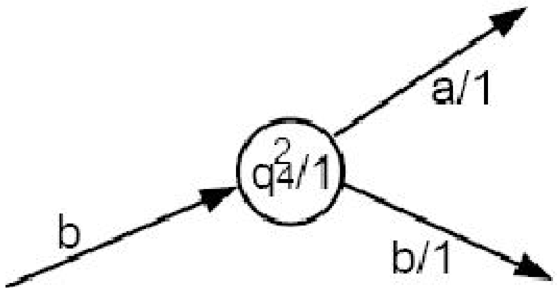
- We cannot just do the reverse of the previous algorithm. If we try to push the printing instruction from the edge (as it is in Me) to the inside of the state (as it should be for Mo), we may end up with a conflict: Two edges may come into the same state but have different printing instructions, as in this example:



Proof by constructive algorithm (cont.):

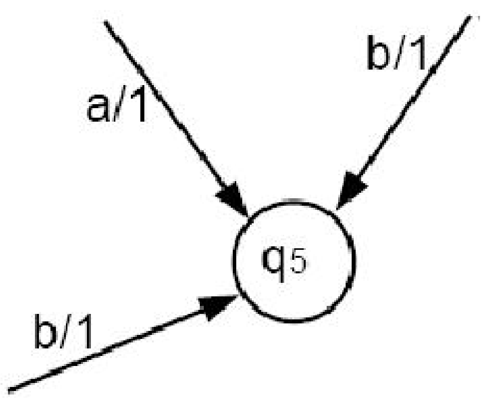
- What we need are two copies of q_4, one that prints a 0 (labeled as $q^1{}_4/0$), and the other that prints a 1 (labeled as $q^2{}_4/1$). Hence,
  - The edges a/0 and b/0 will go into $q^1{}_4/0$.
  - The edge b/1 will go into $q^2{}_4/1$.
- The arrow coming out of each of these two copies must be the same as the edges coming out of q_4 originally.



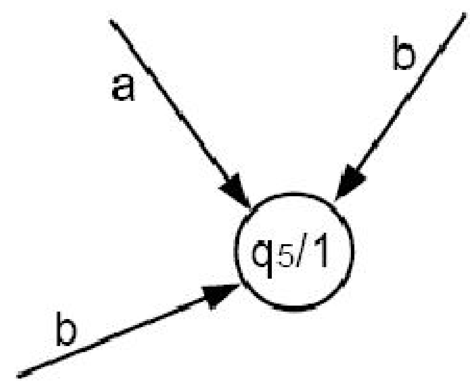


Proof by constructive algorithm (cont.):

- If all the edges coming into a state have the same printing instruction, we simply push that printing instruction into the state.

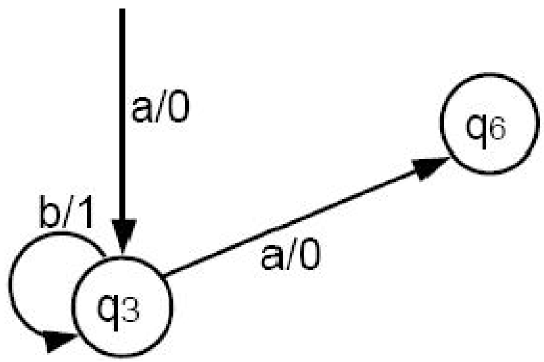


becomes

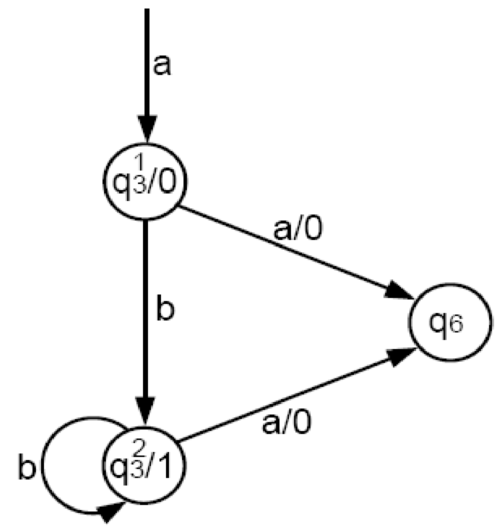


Proof by constructive algorithm (cont.):

- An edge that was a loop in *Me* may becomes two edges in *Mo*, one that is a loop and one that is not.



becomes

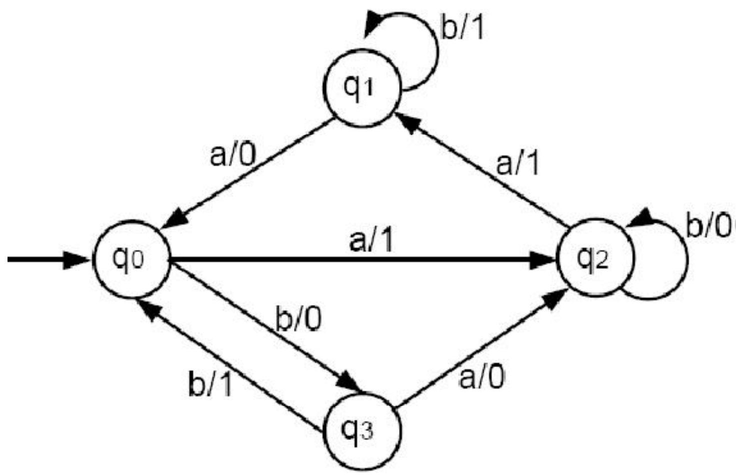


Proof by constructive algorithm (cont.):

- If there is ever a state that has no incoming edges, we can assign it any printing instruction we want, even if this state is the start state.
- If we have to make copies of the start state in *Me*, we can let any of the copies be the start state in *Mo*, because they all give the identical directions for proceeding to other states.
- Having a choice of start states means that the conversion of *Me* into *Mo* is NOT unique.
- Repeating this process for each state of Me will produce an equivalent Mo. The proof is completed.
- **Together, Theorems 8 and 9 allow us to say** Me = Mo**.**

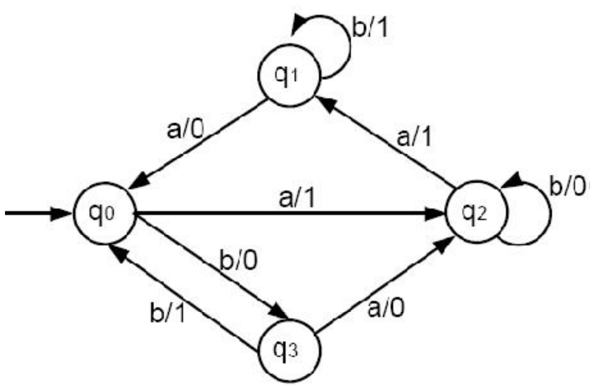
Example

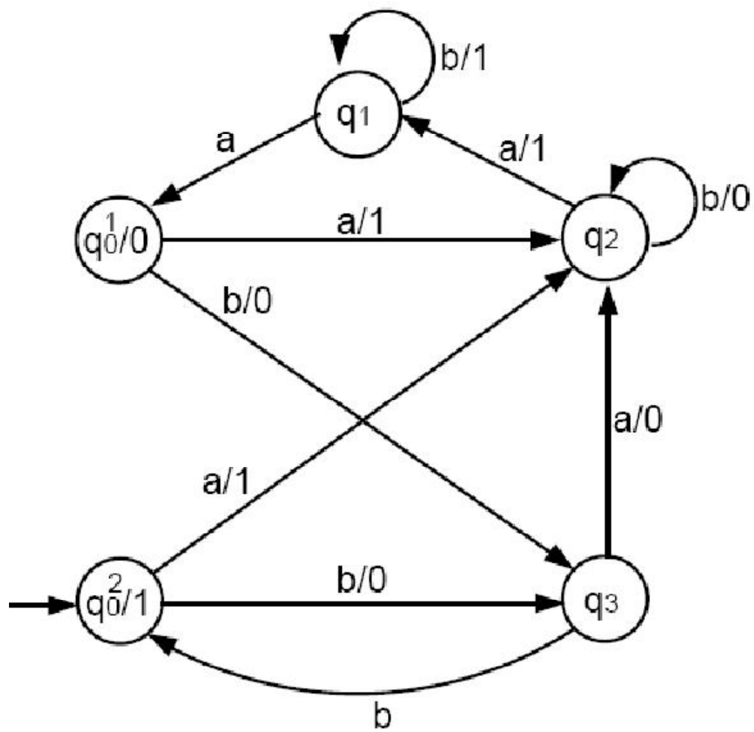
- Convert the following Mealy machine into a Moore machine:



Example contd.

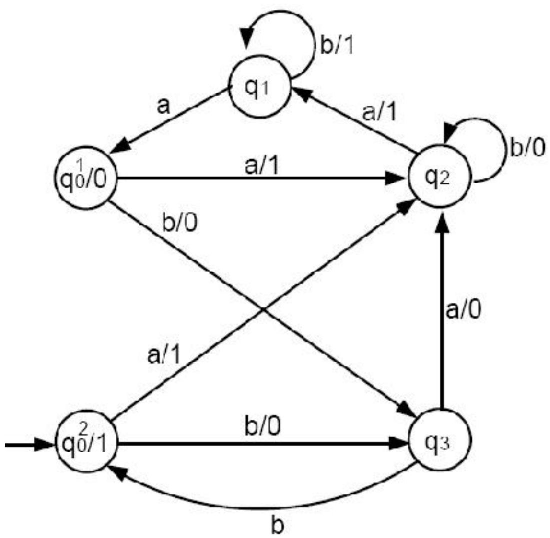
- Following the algorithm, we first need two copies of q_0:

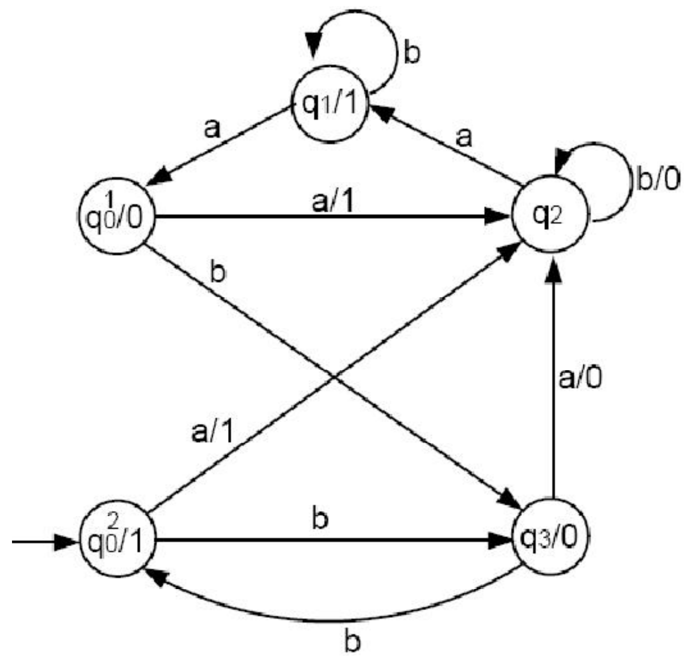




Example contd.

- All the edges coming into state q_1 (and also q_3) have the same printing instruction. So, apply the algorithm to q_1 and q_3:





Example contd.

- The only job left is to convert state q_2. There are 0-printing edges and 1-printing edges coming into q_2. So, we need two copies of q_2. The final Moore machine is: