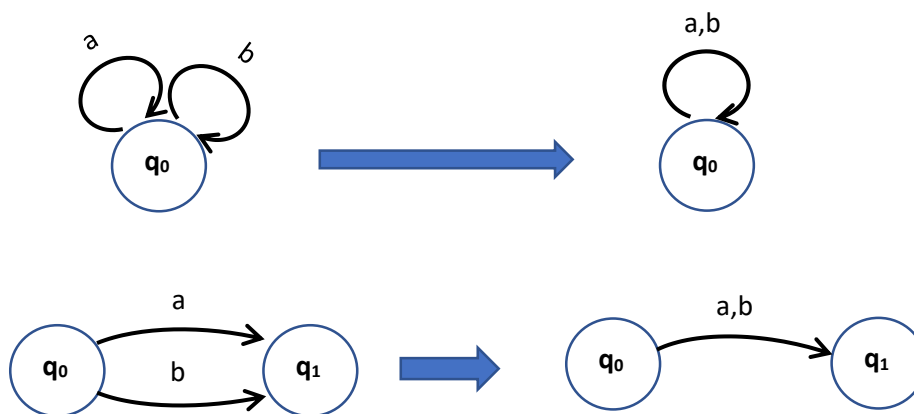


Deterministic Finite State Machine (DFA)

1. DFA refers to deterministic finite automata. Deterministic refers to the uniqueness of the computation. The finite automata are called deterministic finite automata if the machine is read an input string one symbol at a time.
2. In DFA, there is only one path for specific input from the current state to the next state.
3. DFA does not accept the null move, i.e., the DFA cannot change the state without any input character.
4. DFA can contain multiple final states. It is used in Lexical Analysis in Compiler.

It is possible to draw the transitions of this machine as follows:

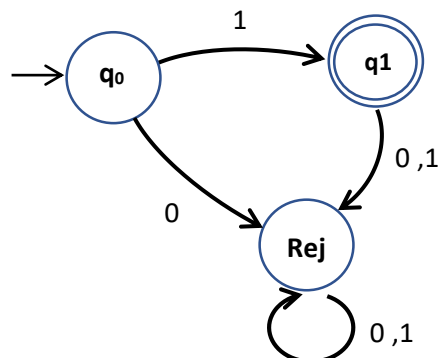


Ex: Draw DFA that accepting exactly 1.

Sol:

$\Sigma = \{0,1\}$

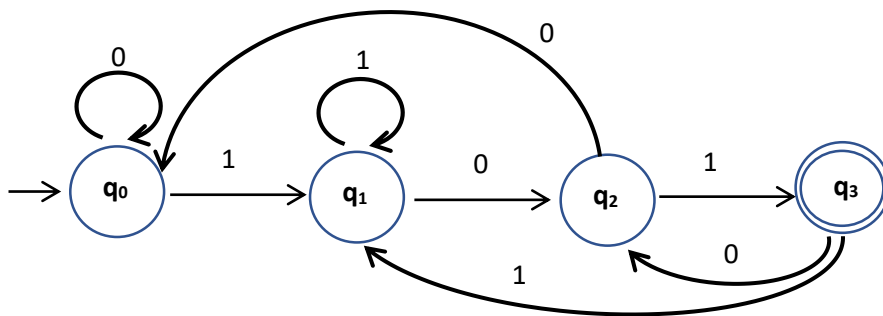
$L = \{1\}$



How to construct DFA for languages consisting of strings ending with a particular substring?

1. Decide the strings for which DFA will be constructed.
2. Determine the minimum number of states required in the DFA
No. of states = $n+1$
 n : length of the string.
3. Construct a DFA for the strings decided in 1.
4. Send all the left possible combinations to starting and do not send the left possible combinations over the dead state.

Ex: Draw a DFA machine for the language accepting strings ending with '101' over input alphabets $\Sigma = \{0,1\}$.



Ex: Draw a DFA machine for the language accepting strings ending with '01' over input alphabets $\Sigma = \{0,1\}$.