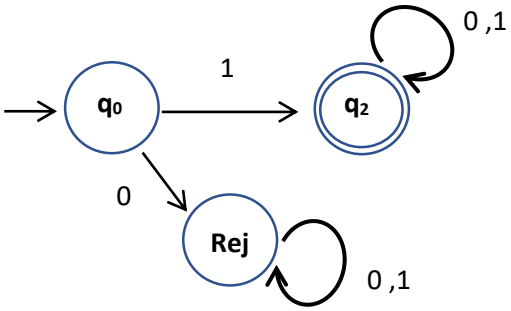
How to construct DFA for languages consisting of strings starting with a particular substring?

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+2
   n: length of the string.
3. Construct a DFA for the strings decided in 2.
4. Send all the left possible combinations to the dead state. Do not send the left possible combinations over starting state.

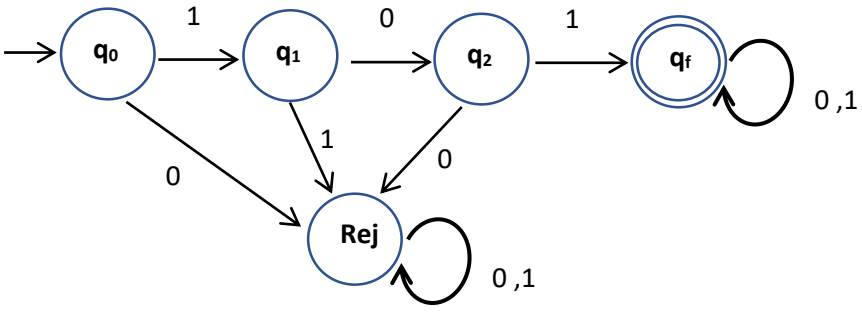
Ex: Draw a DFA machine for the language accepting strings starting with '1' over input alphabets ∑= {0,1}.

Sol:



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

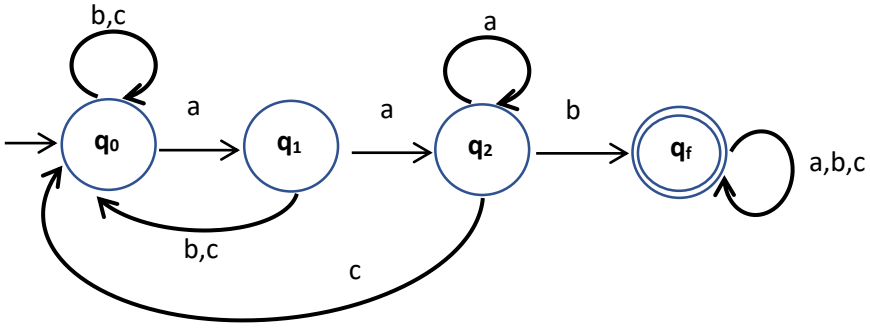
Sol:



Example of constructing DFA for languages consisting of specified substrings anywhere in the word:

Draw a DFA machine for the language accepting substrings ‘aab’ over input alphabets ∑= {a,b,c}.

Sol:

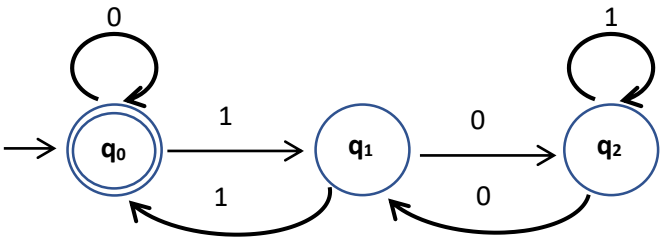


Ex: Design DFA which checks whether a given binary number is divisible by 3.

Sol:

.: binary number ➔ ∑= {0,1}.

Decimal no.	Binary no.	Remainder	States Remaider
0	0000	0	q0
1	0001	1	q1
2	0010	2	q2
3	0011	0	q0
4	0100	1	q1
5	0101	2	q2
6	0110	0	q0
7	0111	1	q1
8	1000	2	q2
9	1001	0	q0



Ex1: Construct a DFA, that accepts set of all strings over ∑= {a,b} of length 2.

i.e |w|=2 → L={aa,ab,ba,bb}

Ex2: Construct a DFA, that accepts set of all strings over ∑= {a,b} of length at least 2.

i.e |w|>=2

Ex3: Draw a DFA for the language accepting string starting with 'ab' over input alphabets ∑= {a,b}.

Ex4: Draw a DFA for the language accepting string ending with 'abba' over input alphabets ∑= {a,b}.

Ex5: Design DFA which checks whether a given binary number is divisible by 4.