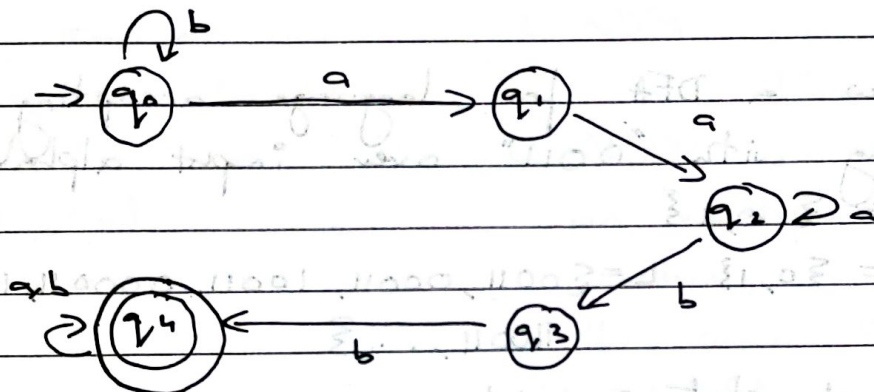


Assignment - 1

- 1) Construct a DFA that accepts any string over $\Sigma = \{a, b\}$ that contain the string "aabb" in it.

$$\rightarrow \Sigma = \{a, b\} \quad L = \{aabb, aaabb, aabbb, baabb, abba, \dots\}$$

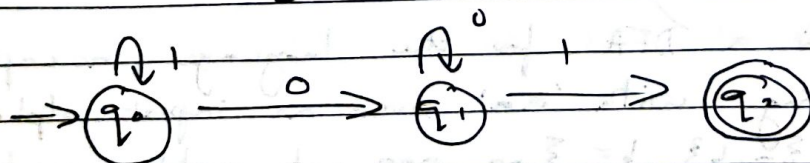
$$\begin{aligned} \text{No. of states} &= |\text{min length}| + 1 \\ &= 4 + 1 \\ &= 5 \end{aligned}$$



- 2) Draw a DFA for the language accepting strings ending with "01" over input alphabets $\Sigma = \{0, 1\}$.

$$\rightarrow \Sigma = \{0, 1\} \quad L = \{01, 001, 101, 0001, 1101, 1001, \dots\}$$

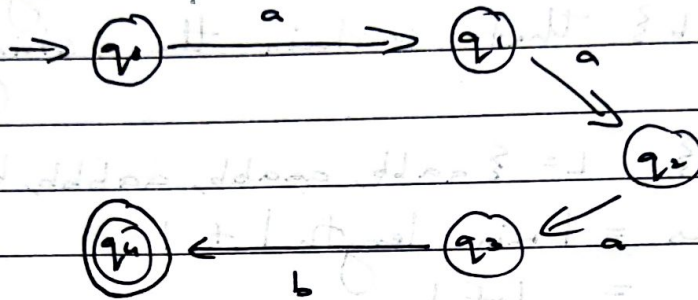
$$\begin{aligned} \text{No. of states} &= |\text{min length}| + 1 \\ &= 2 + 1 \\ &= 3 \end{aligned}$$



- 3) Construct a DFA with $\Sigma = \{a, b\}$ that accepts the only input "aabb".

$$\rightarrow \Sigma = \{a, b\} \quad L = \{aabb\}$$

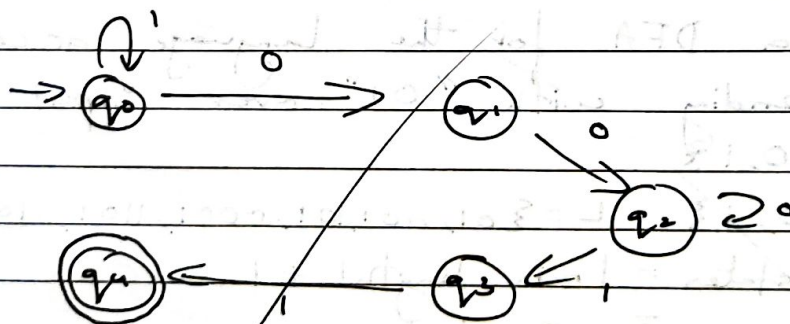
No. of states = $4 + 1$
= 5



4) Draw a DFA for language accepting strings ending with "011" over input alphabets $\Sigma = \{0, 1\}$

$\Sigma = \{0, 1\}$ $L = \{0011, 00011, 10011, 000011, 100011, 110011, \dots\}$

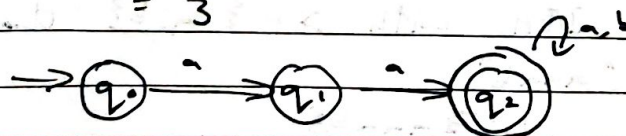
No. of states = $4 + 1$
= 5



5) Draw a DFA for the language accepting strings starting with 'aa' over an input alphabets $\Sigma = \{a, b\}$

$\Sigma = \{a, b\}$ $L = \{aa, aab, aabb, aaab, aaba, \dots\}$

No. of states = $2 + 1$
= 3



Q) Difference between DFA and NDEFA

- | DFA | NDEFA |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1) Stands for Deterministic Finite Automata | 1) Stands for non-deterministic Finite automata |
| 2) For each symbolic representation of alphabet there is only one state transition in DFA | 2) No need to specify how does the NDEFA react according to same symbol |
| 3) DFA cannot use Empty string transition | 3) NDEFA can use Empty string transition |
| 4) DFA can be understood as one machine | 4) NDEFA can be understood as multiple little machines computing at the same time |
| 5) In DFA, the next possible state is distinctly set | 5) In NDEFA, each pair of state and input symbol can have many possible next states. |
| 6) DFA is more difficult to construct | 6) NDEFA is easier to construct. |
| 7) Time needed for executing an input string is less | 7) Time needed for executing an input string is more. |
| 8) All DFA are NDEFA | 8) Not all NDEFA are DFA. |

