Pumping lemma for Regular languages.

Theorem definition.

Let $M = \{Q, \Sigma, \delta, q_0, F\}$ be the deterministic finite automata and has 'n' number of states. Let L be the language accepted by above DFA.

Let every string $x \in L$, there exists a constant 'n' such that $|x| >= n$, i.e. the length of the string is ~~very~~ larger than the no. of states.

Now if the string 'x' is decomposed into three substrings u, v and w such that $|uv| <= n$ and $|v| >= 1$. Then

$$x = uv^i w \in L \quad \text{for all } i = 0, 1, 2, \ldots$$

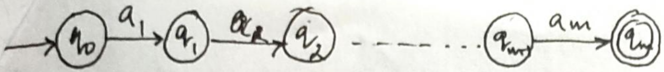
Proof:

Given that the DFA $M = \{Q, \Sigma, \delta, q_0, F\}$. and L is the language accepted by above DFA 'M'. Hence the given 'L' is regular.

Let $x = a_1 a_2 \ldots a_m$ be the string whose length is greater than the number of states 'n' i.e. $|x| >= n$.

~~Since~~ We have 'm' number of symbols.. To accept one symbol by DFA, we ~~have~~ require 2 distinct states, two symbols we require 3 distinct states.. As there are 'm' number of symbols in the string 'x' i.e. $|x| = m$, we must have 'm+1' states in the sequence.

$q_0, q_1 \ldots q_m$.. and which is shown below,



Here q_0 is start state and q_m is final states

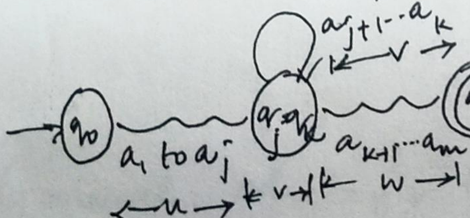
Since $|x| >= n$, according to The pigeon hole principle, we cannot have $n+1$ distinct transitions and states. Hence There must be One state that can have a loop. In other words. there exists two indices j and K. such that $0 <= j <= K <= n$ and associated states q_j and q_K must be equal..

Let x is divided into three sub strings $u, v, \& w$ with, prefix, loop string, and suffix, and is shown below.

ie. u is prefix of $x = a_1 a_2 \cdots a_j$

v is loop string $x = a_{j+1} a_{j+2} \cdots a_K$.

w is suffix of $x = a_{K+1}, a_{K+2} \cdots a_m$



from the above figure it is very clear That the prefix string 'u' takes The DFA from q_0 to q_j, the loop string 'v', takes The DFA from q_j to q_K (Here, $q_j = q_K$). and Suffix string 'w' takes The DFA from q_K to q_m.

if $i = 0$, The string uw is accepted by DFA 'M'

for $i = 1$ The string uvw is " " " "

for $i = 2$ The string $uvvw$ " " " " "

(3)

In general $i > 0$, The Dfa goes from ~~string~~ state to q_0 to q_j on string 'u', loops from q_j to q_K on the string 'v' and goes to accepting state from q_K to q_m on string 'w'

Therefore the string 'x' is divided into 3 - substring uvw. Then for ~~for~~ all $i >= 0$. hence

$x = uv^iw \in L$. and The ~~hence~~ proof

This can be expressed as follows.

$$\delta^*(q_0, a_1 a_2 \cdots a_m) = \delta(q_0, a_0, a_1 \cdots a_j a_{j+1} \cdots a_K, a_{K+1} a_{K+2} \cdots a_m)$$

$$= \delta(q_j, a_{j+1} a_{j+2} \cdots a_K, a_{K+1} a_{K+2} \cdots a_m).$$

$$= \delta(q_K, a_{K+1} a_{K+2} \cdots a_m) \because q_j = q_K.$$

$$= q_m \in F.$$

General strategy to apply pumping lemma to prove certain languages are not Regular

1. Assume that the language 'L' is regular and 'n' be the number of states of DFA.
2. Select the string 'x' such that $|x| >= n$ for some positive number 'n' and divide the string 'x' into three substrings, u, v and w. so then $x = uvw$. Such that $|uv| <= n$ and $|v| >= 1$

3. Find any i such that

$x = uv^iw \notin L$. According to pumping lemma. The result is contradiction to the assumption that the language 'L' is regular, Therefore the given language L is not regular.