Module 2 continued.

Pumping Lemma for Regular Languages.

Theorem:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a Finite Automata and has n number of states. Let L be the regular language accepted by M. Let for every string x in L, there exists a constant n such that $|x| \geq n$. Now, if the string x can be broken into three substrings u, v and w

such that $x = uvw$

satisfying the following constraints:

(i) $v \neq \varepsilon$ ie $|v| \geq 1$

(ii) $|uv| \leq n$

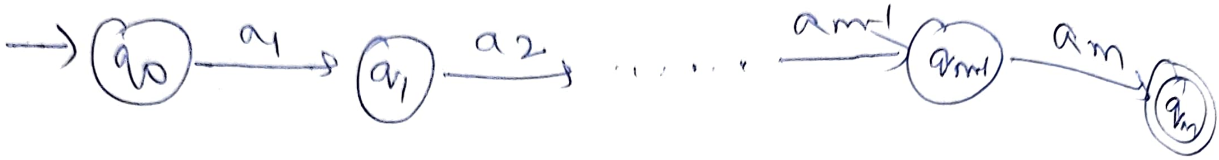
then uv^iw in L for $i \geq 0$

Proof:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an Finite Automata and let L is the language accepted by DFA and is regular. Let $x = a_1a_2 \ldots a_m$ where $m \geq n$ and each a_i is in Σ. Here, n represent the states of DFA.

Since there are m input symbols, we should have $m+1$ states in the sequence $q_0 q_1 \ldots q_m$

where q_0 will be the start state and q_m will be the final state as shown below:

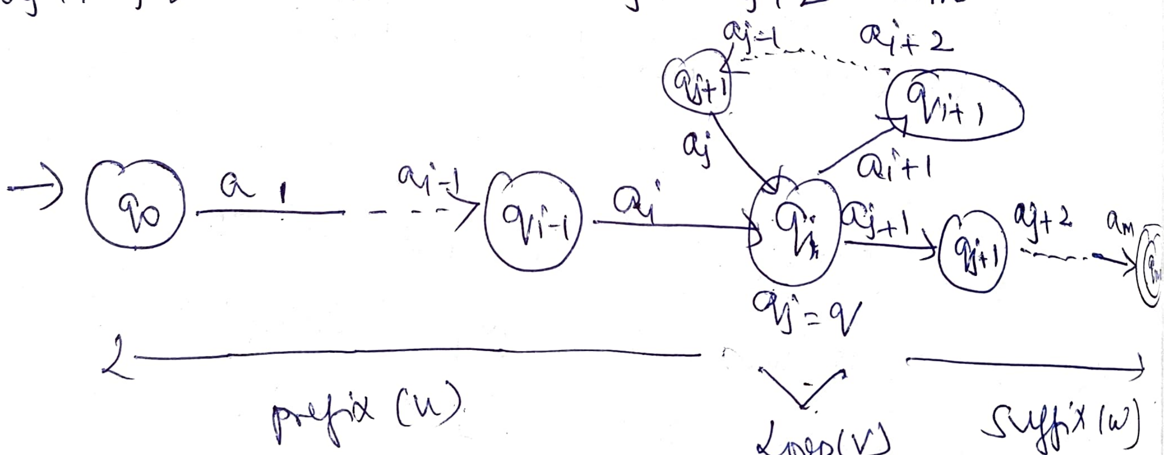


Based on Pigeon Hole principle,
Let the string x is divided into three substrings as shown below:

- The first group is the string prefix from $a_1 a_2 \cdots a_i$ i.e $u = a_1 a_2 \cdots a_i$

The second group is the loop string from $a_{i+1} a_{i+2} \cdots a_{j-1} a_j$ i.e $v = a_{i+1} a_{i+2} \cdots a_{j-1} a_j$

The third group is the string suffix from $a_{j+1} a_{j+2} \cdots a_m$ i.e $w = a_{j+1} a_{j+2} \cdots a_m$



From the above figure, the prefix string u takes the machine from q_0 to q_i, the loop string v takes the machine from q_i to q_i and

suffix string w takes the machine from q_j to q_f.
The minimum string that can be accepted by the above finite automata is uw with $i=0$.

when $i=1$, the string uvw is accepted
when $i=2$, the string $uvvw$ is accepted.

so if $i>0$, the machine goes from q_0 to q_i on input u, circles from q_i to q_i based on values of i and goes to accepting state on input w.

In general, the string x is split into substring uvw, then for all $i \geq 0$,
$$uv^iw \in L$$
This can be shown as:-
$$\delta(q_0, a_1 a_2 \ldots a_{i-1} a_i a_{j+1} a_{j+2} \ldots a_m)$$
$$= \delta(\delta(q_0, a_1 a_2 \ldots a_{i-1} a_i), a_{j+1} a_{j+2} \ldots a_m)$$
$$= \delta(q, a_{j+1} a_{j+2} \ldots a_m)$$
$$= \delta(q_k, a_{k+1} a_{k+2} \ldots a_m)$$
$$= (q_m) = q_m$$
Hence, the machine enters the final state.

Problems:

To show that the following languages are not regular.

(i) $L = \{WW^R \mid W \in (0+1)^*\}$

Let $W = 110$.

$x = WW^R = 110011$

$u = 110$

$v = 0$

$w = 11$

pump v

$x = 1100011$

On pumping v, it gives a contradiction, to the language, hence the language is not regular.

(ii) $L = \{a^n b^n \mid n \geq 0\}$

when $n=1$, $x = ab$

$n = 2$, $aabb$

$u = a$

$v = a$

$w = bb$

pump v

$x = aaabb$, on pumping v, it gives a contradiction to the language, hence the language is not regular.

(iii) $L = \{a^i b^j \mid i > j\}$

$i = 2, j = 01$

~~$x = a\,bb$~~ $\quad x = aab$

$i = 3, j = 2$

$aaabb$

$u = aaa$

$v = b$

$w = b$.

pump v for $i = 3$.

$x = aaabbbb$, it is a

contradictory. hence the language is not regular.

~~(iv) $L = \{a^n b^1 \mid n \neq 1\}$.~~

~~$n = 2$.~~	~~$n = 4$~~
~~$x = aab$.~~	~~$x = aaaab$~~
~~$x = aab$~~	~~$u = a$~~
~~$u = a$~~	~~$v = aa$~~
~~$v = a$~~	~~$w = ab$.~~
~~$w = b$.~~	

(iv) $L = \{w \mid n_a(w) < n_b(w)\}$

$x = aabbb$.

$u = a$

$v = a$

$w = bbb$.

pump v.

$x = aaabbb$, it is contradiction, hence, the language is not regular.

(v) $L = \{0^n \mid n \text{ is prime}\}$

$n = 2$

$l = 00$

$n = 3$

$l = 000$

$u = 0$

$v = 0$

$w = 0$

pump v is 2

$x = 0000$, it is contradiction to the given language, hence, it is not regular.

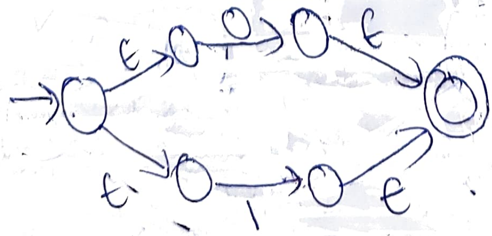
Properties of Regular Language.

Union, Intersection and Complement are known Boolean Operations.

(1) Union: For any languages L and M, $L \cup M$ is regular.

Proof: Since L and M are regular, they have regular expressions; say $L = L(R)$ and $M = L(S)$. Then $L \cup M = L(R \cup S)$ by the definition of the $\cup$ operator for regular expressions.

Ex: $0 \cup 1$



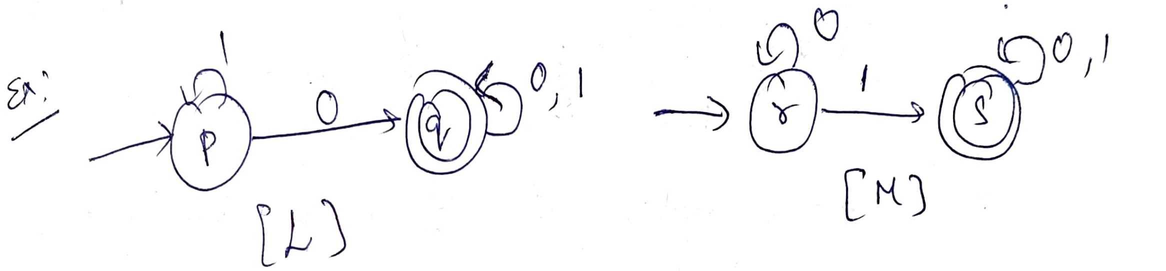
② **Intersection**: If L and M are regular languages, then so is $L \cap M$.

Proof: By DeMorgan's law $L \cap M = \overline{\bar{L} \cup \bar{M}}$

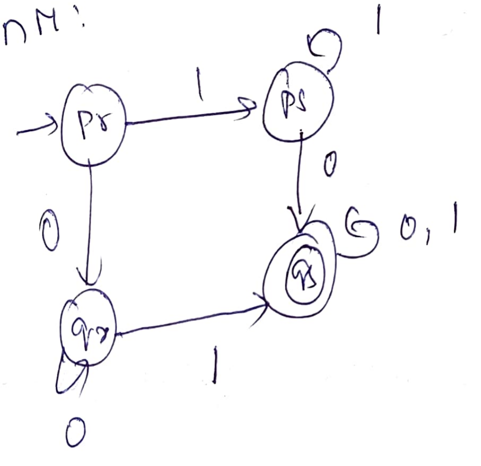
Also, regular languages are closed under complement and union.

So, $L \cap M$ is regular when L and M are regular.

Ex:



$L \cap M$:



③ **Complement**

If L is a regular language over alphabet Σ, then $\bar{L} = \Sigma^* - L$ is also a regular language.

Proof:

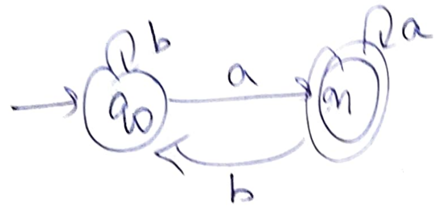
Let $L = L(A)$ for some DFA $A = (Q, \Sigma, \delta, q_0, F)$

Then $\bar{L} = L(B)$, where B is the DFA $(Q, \Sigma, \delta, q_0, Q - F)$

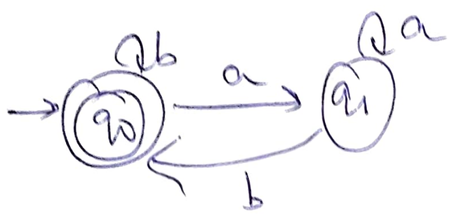
i.e B is exactly like A, but the accepting states of A have become non-accepting states of B, and vice versa.

note:
set difference
$L(M_1) - L(M_2) = L(M_1) \cap \neg L(M_2)$

<u>Ex:</u> L: ends with 'a'



Now complement of L is does not ends with 'a'



④ <u>Concatenation</u>

If L and M are regular language, then so is LM.

Let $L = L(R)$ and $M = L(S)$, then

$LM = L(RS)$

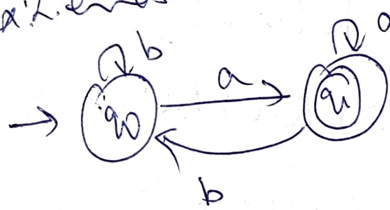
Ex: 011

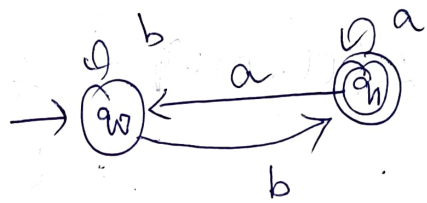
Ex

⑤ Reverse

The reversal of a language L, written L^R, is the language consisting of the reversals of all its strings.

Ex: L: ends with 'a' L^R: Reversal of ends with 'a'



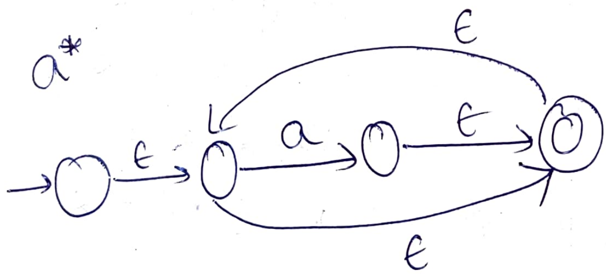


⑥ Kleen star

If L is a regular language, then so is L^*.

Let $L = L(R)$, then $L^* = L(R^*)$

Ex: a^*



⑦ Homorphism

Homorphism is a substitution where a single letter is replaced by a string.

If L is made of alphabets from Σ, then $h(L) = \{h(w) \mid w \in L\}$ is called a homomorphic image.

Ex: Let $\Sigma = \{0,1\}$, $\Gamma = \{0,1,2\}$ and $h(0) = 01$, $h(1) = 112$. What is $h(010)$? If $L = \{00, 010\}$ what is the homomorphic image?

$$h(010) = h(0)h(1)h(0) = 0111201$$

$$L\{00, 010\} = L(h(00), h(010))$$

$$= L(h(0)h(0)\ h(0)h(1)h(0))$$

$$= L(0101, 0111201)$$

Simplification of Regular Expressions

Rules:

$\alpha \cup \beta = \beta \cup \alpha$

$(\alpha \cup \beta) \cup \gamma = \alpha \cup (\beta \cup \gamma)$

$\alpha \cup \phi = \phi \cup \alpha = \alpha$

$\alpha \cup \alpha = \alpha$

If $B \subseteq A$, then $A \cup B = A$

Ex: $a^* \cup aa = a^*$

$\epsilon \cdot \alpha = \alpha \cdot \epsilon = \alpha$

9. $\epsilon^* = \epsilon$

10. $\alpha^* . \alpha^* = \alpha^*$

11. $\phi \cup R = R$

12. $\alpha \cup \alpha = \alpha$

13. $\alpha^* \alpha^* = \alpha^*$

14. $(\alpha^*)^* = \alpha^*$

15. $\alpha^* \alpha \cup \alpha = \alpha^* \alpha$