Pumping Lemma for regular language

Statement: If language A is regular language the A has a pumping length 'p' such that any string 's' where $|s| \geq p$ any be divided into 3 points $s = xyz$ such that following conditions must be true.

(a) $xy^iz \in A$ for every $i \geq 0$

(b) $|y| > 0$ (length of y)

(c) $|xy| \leq p$ (length of x and y)

Steps to be followed while proving language is not reular by pumping lemma:

(a) Assume that A is regular.

(b) It has to have a pumping length (say P).

(c) All strings longer than P can be pumped.

(d) Now find a string 's' in A such that $|S| \geq P$.

(e) Divide S into x, y, z.

(f) Show that, $xy^iz \in A$ for some i.

(g) Then consider all ways that S can be divided into x, y, z.

(h) Show that none of these can satisfy all the 3 pumping conditions at the same time.

(i) S can't be pumped = contradiction.

(j) We should learn it by some examples.

1. **Using pumping lemma prove that the language $A = \{a^n b^n / n \geq 0\}$ is not regular.**

Solution:

(Theoretically, we can conclude that this language is not regular as the language generates strings which contains certain number of a's followed by equal number of b's. And to do that, finite state machine need to have some memory for storing counted number of a but in actual it doesn't have memory. So, if a language is not recognized by finite state machine, it is not regular.)

Using pumping lemma, let's assume that language A is regular. Now, it must follow following conditions.

(a) $XY^iz \in A$ for every $i \geq 0$ ... ... (i)

(b) $|y| > 0$ ... ... (ii)

(c) $|xy| \leq p$ where pis pumping length... ... ... (iii)

Let, $P = 7$ (say)

Let, $S \in A = a^P b^P$

$= a^7b^7$

= aaaaaaabbbbbbb

Dividing S into x, y and z. Gets these cases.

Case – I

When y is in the 'a' part.

i.e. $S = \underbrace{a}_{x} \quad \underbrace{aaaa}_{y} \quad \underbrace{aabbbbbbb}_{z}$

Here, condition 2 and 3 are satisfied as $|y| = 4 > 0$ and $|xy| = 5 < 7$.

For condition 1

$xy^iz = xy^2z$ [i = 2 say]

Then,

S = a aaaa aaaa aabbbbbbb

So, number of 'a' = 11

number of 'b' = 7

∴ Number of 'a' ≠ number of 'b'.

The first condition is not satisfied.

i.e. $xy^2z \notin A$

Case – II:

When y in the both 'a' and 'b' part

i.e. $S = \underbrace{aaaaaa}_{x} \quad \underbrace{ab}_{y} \quad \underbrace{bbbbbb}_{z}$

Let $S = xy^2z$ $(i = 2)$

$= aaaaaa \; abab \; bbbbbb$

Here, pattern is not followed i.e. $a^n b^n$.

So, condition 1 is not satisfied and condition 3 as well as $|xy| = 8 > 7$

Case - III

When y is in the 'b' part.

i.e. $S = \underbrace{aaaaaaabb}_{x} \quad \underbrace{bbb}_{y} \quad \underbrace{bb}_{z}$

Let, $S = xy^2z$

$= aaaaaaabb \quad bbbbbb \quad bb$

Here, also condition 1 is not satisfied as, number of a's $\neq$ number of b's,

So, none of the cases satisfy the all 3 conditions of regular language as stated by pumping lemma. Therefore language A is not regular.

Proving Langauge not to be Regular

It is shown that the class of language known as regular language has at least four different descriptions. They are the language accepted by DFA's, by NFA's, by Є-NFA, and defined by RE.
Not every language is Regular. To show that a langauge is not regular, the powerfull technique used is known as Pumping Lemma.

Pumping Lemma

Statement: Let L be a regular language. Then, there exists an integer constant n so that for any $x \in L$ with $|x| \geq n$, there are strings u, v, w such that $x = uvw$, $|uv| \leq n$, $|v| > 0$.
Then $uv^kw \in L$ for all $k \geq 0$.

Note: Here y is the string that can be pumped i.e repeating y any number of times or deleting it, keeps the resulting string in the language.

Statement: Let L be a regular language. Then, there exists an integer constant n so that for any x ε L with $|x| \geq n$, there are strings u, v, w such that x = uvw, $|uv| \leq n$, $|v| > 0$. Then uv^kw ε L for all $k \geq 0$.

Note: Here y is the string that can be pumped i.e repeating y any number of times or deleting it, keeps the resulting string in the language.

Proof:

Suppose L is a regular language, then L is accepted by some DFA M. Let M has n states. Also L is infinite so M accepts some string x of length n or greater. Let length of x, $|x| = m$ where $m \geq n$.

Now suppose;
$X = a_1a_2a_3\ldots\ldots\ldots\ldots a_m$ where each a_i ε Σ be an input symbol to M. Now, consider for j = 1,…………n, qj be states of M
Then,
$\hat{\delta}(q_0,x) = \hat{\delta}(q_0,a_1a_2\ldots\ldots a_m)$ [q0 being start state of M]
$= \hat{\delta}(q_1,a_2\ldots\ldots a_m)$
=…………………
=…………………
=…………………
$= \hat{\delta}(q_m,\epsilon)$ [qm being final state]

Since $m \geq n$, and DFA M has only n states, so by pigeonhole principle, there exists some i and j; $0 \leq i < j \leq m$ such that $q_i = q_j$.

Now we can break x=uvw as
u = a1a2………….ai
v =ai+1……………aj
w =aj+1……………am

i.e. string ai+1 ………………aj takes M from state qi back to itself since qi = qj. So we can say M accepts $a_1a_2\ldots\ldots\ldots a_i(a_{i+1}\ldots\ldots\ldots a_j)^K a_{i+1}\ldots\ldots\ldots a_m$ for all k≥0.

∄ It is a negative test

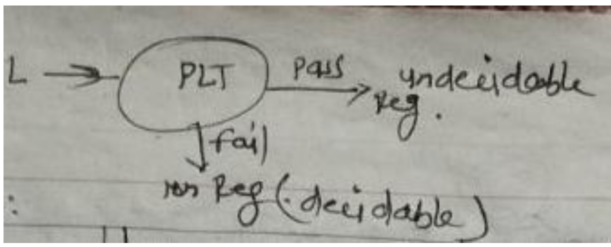
VVI

Pumping Lemma for Regular language

The pumping lemma is used to determine whether the language is regular or not i.e. it is used to determine the class of language of FA.

It is a relationship between the length of string (n) and the no. of states (m) of the given finite automata.

It is a powerful tool for proving the certain language non-regularity.



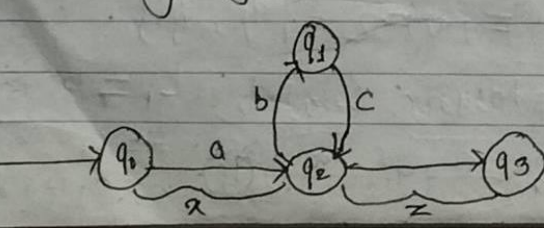
(i) $y \neq \epsilon$ (empty) $[y > 0]$

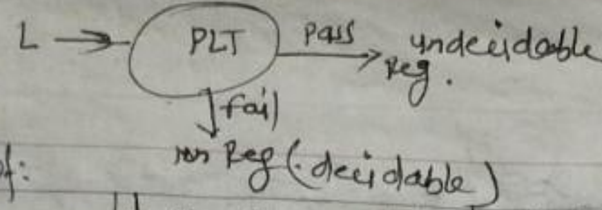
(ii) $|xy| \leq n$

(iii) string $xy^iz \in L$ for all $i \geq 0$

Note: Here y is the string that can be pumped i.e repeating y any number of times or deleting it, keeps the resulting string in the Language.

(eg.)





Proof:

Let $q_1, q_2, \ldots, q_m$ be the m states of finite automata and $w = a_1, a_2, \ldots, a_n$ be the string.

Let 'L' be the regular language and $w \in L$.

Let us decompose w into three substrings x, y, z.

$w = a_1, a_2, \ldots, a_n$

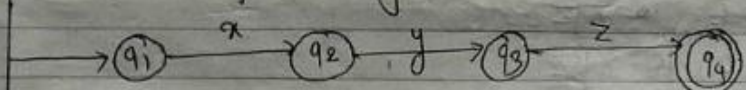
$x = a_1, a_2, \ldots, a_i \leftarrow a$

$y = a_{i+1}, a_{i+2}, \ldots, a_j$

$z = a_{j+1}, a_{j+2}, \ldots, a_n$

Also, we can say that $w = xyz \in L$

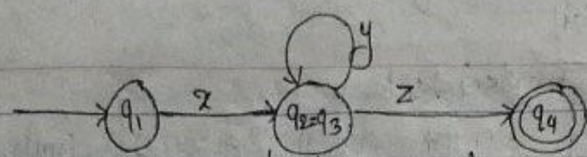
Now, FA for $w = xyz$ is:



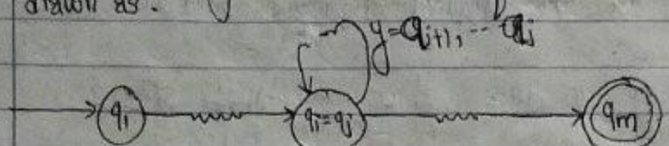
Here, $|w| = 3$ &

total states $(m) = 4$, which violates the rule for puming lemma.

Hence, using <u>Pigeonhole principle</u> we can say that any 2 states of FA must coincide. Let states (q_2 and q_3) coincide then the model is:



Now, the generalized form of above automata can be drawn as:



$x = q_1, \cdots q_i$ $\qquad z = q_{j+1}, q_{j+2}, \cdots q_m$

Here

$$\boxed{\underbrace{1 \le i \le j}_{xy} \le m}$$

$\therefore |xy| \le m$

i.e. $|xy| \le n$

Now,

when $i = 0,\ w = xz$

$i = 1,\ w = xyz$

$i = 2,\ w = xy^2z$

$\vdots$

$i = n,\ w = \cancel{xz}\ xy^nz$

Similarly

xy^iz for all $i \geq 0$

$$\boxed{\therefore xy^iz \in L}$$

~~Proved~~

2. Using pumping lemma prove that A = {yy/y ∈ {0, 1}* is not regular.

Solution:

According to pumping lemma, 'if a language is regular then it must have string $|S| \geq P$ where P is pumping length and can be divided into xyz such that:

(a) $xy^iz \in A$ for some i ... ... (i)

(b) $|y| > 0$ ... ... (ii)

(c) $|xy| \leq p$ ... ... (iii)

To prove: Language A = {yy//y ∈ (0, 1)*} is not regular

Proof: Let us consider language A is regular then it must follow above 3 conditions. Then it must have pumping length P = 7 (say).

Let, the strings be accepted by finite state machine

$S = 0^p10^p1$

$= 0^7 1 0^7 1$

$= 0000000100000001$

Dividing S into x, y and z we get,

$$S = \underbrace{00}_{x} \quad \underbrace{00000}_{y} \quad \underbrace{100000001}_{z}$$

Here, the above condition (ii) and (iii) are satisfied as

$|y| \geq 0$

$|xy| = 7 \leq 7$

Changing for condition 1

$S = xy^iz \qquad \text{let, } i = 2$

$= xy^2z$

$= 00 \; 0000000000 \; 100000001$

Here, , on pumping i.e. (xy^iz) the language doesn't follow pattern (O^p10^p1) (i.e. first half is equal to the second half). So none of the cases follow all 3 conditions of regular language as stated by pumping lemma. Therefore language A is not regular language.

State the pumping lemma for the regular languages. Show that the languages

$L = \{0^{n^2} \mid n >= 1\}$ not regular example,

if x = 1, w = 0, n = 2, w = 0000, n = 3, w = 000000000 [2074, Chaitra]

Pumping lemma states that if a language A is regular then it must have string $|s| \geq P$ where P is pumping length and can be divided into xyz such that

(a) $xy^iz \in A$ for some i

(b) $|y| > 0$

(c) $|xy| \leq P$

Here, given language is $L = \{0^{n^2}\} / n >= 1\}$

To prove: L is not regular

Proof: Using pumping lemma

Let's assume language L is regular.

Then, if must have pumping length P.

Let P = 3 (say)

Then, let s be the string accepted by finite automaton.

$$
\begin{aligned}
\text{i.e. } s &= o^{P^2} \\
&= 0^{3^2} \\
&= 0^9 \\
&= 000000000
\end{aligned}
$$

Dividing s into x, y and z as

$$\underbrace{0}_{x} \quad \underbrace{00}_{y} \quad \underbrace{000000}_{z}$$

Checking for condition 1

$$
\begin{aligned}
s &= xy^iz = xy^2z \qquad (e = 2 \text{ (say)}) \\
&= 0\ 0000\ 000000
\end{aligned}
$$

Now, $|s| = 11$

Here, no value of P(or n) would give (s) = 1 in other words, length of string s on being pumped is not perfect square which means it doesn't belong to language L so, condition 1 fails. Though, condition 2 and 3 are satisfied.

$|y| = 2 > 0$

$|xy|\ 3 \geq P$

Since, condition 1 is failed, our assumption contradicts. That's why language L is not regular.

11. State pumping lemma for regular language. Use pumping lemma and prove that language $L = \{w \mid w \in \{0, 1\}^*$ and w has an equal number of 0's and 1's$\}$ is not regular. **[2073 Chaitra]**

For statement of pumping lemma for regular language, please see theory part.

Proof: Let, assume that language L is regular. Now, since it is regular, it must follow these conditions:

(a) $xy^iz \in L$ for every $i \geq 0$

(b) $|y| > 0$

(c) $|xy| \leq P$ where P is pumping length

Lent, $S \in L = 01010101$ and $P = 3$

Now, dividing S into x, y and z, we get

$$\underbrace{0}_{x} \quad \underbrace{10}_{y} \quad \underbrace{10101}_{z}$$

Here, the above condition (ii) and (iii) are satisfied as

$|y| = 2 > 0$

$|xy| = 2 \leq 3$

Now, checking for condition 1,

xy^iz

Let, $l = 2$

xy^2z

$= 01010\ 10101$

Then, no of 0's is not equal to number of 1's so, it doesn't belong to L and condition 1 fails. If any of the above condition fails, then our assumption is also false which means language L is not regular.

16. State pumping lemma, for regular language and use the theorem to prove that $L = \{a^n b^{2n} : n \geq 1\}$ is not regular. [2073, Shrawan]

If a language L is a regular language, then it must have a pumping length 'P' such that any string $S \in L$ having length $|S| \geq P$ can be divided into 3 parts such that following conditions are true.

(a) xy^iz for every $i \geq 0$

(b) $|y| > 0$

(c) $|xy| \leq p$

Given, $L = \{a^n b^{2n} : n \geq 1\}$

To prove: L is not regular

Proof: We prove it by contradiction using pumping lemma.

Let's assume language L is regular having pumping length 'P'.

Let, $S \in L = a^p b^{2p}$

Let, P = 4 (say)

Then, S $= a^4 b^8$

= aaaabbbbbbbb

Dividing S into x, y and z, we get

Case - I: When y contains m' y 'a'

$$S = \underbrace{aa}_{x}\ \underbrace{aa}_{y}\ \underbrace{bbbbbbbb}_{z}$$

Checking for condition I

Let, i = 2

S $= xy^2z$

= aa aaaa bbbbbbbb

$= a^6b^8 \notin L$

Here, condition 1 fails as 'b' is not as twice as 'a'.

Case – II: Why y contains both kind of symbols 'a' and 'b' i.e.

$$S = \underbrace{aa}_{x}\ \underbrace{aaa}_{y}\ \underbrace{bbbbbbb}_{z}$$

Checking for condition I

Let, $i = 2$

$S = xy^2z$

$= aa\ aab\ aab\ bbbbbbb \notin L$

Again, condition 1 fails as it doesn't follow pattern

Case – III: When y contains only b's i.e.

$$S = \underbrace{aaaa}_{x}\ \underbrace{bbbb}_{y}\ \underbrace{bbbb}_{z}$$

Checking for condition I

Let, $i = 2$

$S = xy^2z$

$= aaaa\ bbbbbbbbbbbb$

$= a^4 b^{12} \notin L$

Again, condition 1 fails

Here, in pumping (i.e. xy^iz) doesn't follow condition 1. So, none of the cases follow all three condition of regular language as stated by pumping lemma. Therefore, language L is not regular language.

18. Define pumping lemma for regular languages. Use pumping lemma for regular language to show L [a^nba^n for n = 0, 1, 2, ...] is not regular. **[2072, Chaitra]**

Pumping lemma states that if a language L is regular then it must have $|s| \geq p$ where p is pumping length and s is the string, $S \in L$ which can be divided into x, y and z parts such that following conditions are true.

(1) $xy^iz \in L$ for every $i \geq 0$

(2) $|y| > 0$

(3) $|xy| \leq p$

To prove: $L = \{a^n b a^n$ for $n = 0, 1, 2, ...\}$ is not regular.

Proof: Let's assume language L is regular. Then it must have pumping length 'p'.

Let, $S \in L = q^p b a^p$

Let, $p = 3$

$S = a^3 b a^3$

aaabaaa

Dividing S into x, y and z, we get

Case – I: When 'y' contains only 'a'

$$S = \underbrace{a}_{x}\ \underbrace{aa}_{y}\ \underbrace{baaa}_{z}$$

Checking for condition I

Let, $i = 2$

$S = xy^2z$

$= a\ aaaa\ bbaaa$

$= a^5ba^3 \notin L$

So, condition 1 fails as the power of a's at the start and end of the string are not equal.

Case – II: When y contains both symbol 'a' and 'b'

$$S = \underbrace{aa}_{x}\ \underbrace{ab}_{y}\ \underbrace{aaa}_{z}$$

Checking for condition I

Let, $i = 2$

$S = xy^2z$

$= aa\ abab\ aaa \notin L$

So, condition 1 fails as it doesn't follow pattern $a^n ba^n$.

Hence, on pumping (i.e. xy^iz), none of the cases follow all three conditions of regular language as stated by pumping lemma. Therefore, language L is not regular.

21. State pumping lemma for regular language. Use this lemma to prove language L : {a^{n^2} : n $\geq$ 0} is not regular. **[2071 Chaitra]**

For statement of pumping lemma, see theory part.

Here, given language L = {a^{n^2}, n $\geq$ 0}

To prove: L is not regular

Proof: Let's assume language L is regular.

Then, it must have pumping length P.

Let, P = 3 (say)

Then, let S be the string accepted by finite automata.

i.e. $S = a^{P^2}$

$= a^{3^2}$

$= a^9$

Dividing S into x, y and z, we get

$$S = \underbrace{a}_{x}\ \underbrace{aa}_{y}\ \underbrace{aaaaaa}_{z}$$

Now, checking for condition I

$S = xy^iz$

Let, i = 2

$\therefore S = xy^2z$

= a aaaa aaaaaa

Now, |S| = 11

Here, no value of P (or n) would give |S| = 11 in other words, length of string S an being pumped is not perfect square which means it doesn't belong to language L.

26. State the pumping lemma for regular language show that the language $L = \{a^n : n \text{ is prime}\}$ is not regular using the pumping lemma. **[2072 Kartik, Back]**

For statement of pumping lemma, see theory part.

To prove: $L = \{a^n : x \text{ is a prime}\}$ is not a regular

Proof: Let's assume that language L is regular. Then, according to pumping lemma, it must follow these conditions.

(1) $xy^iz \in L$ for every $i \geq 0$

(2) $|y| > 0$

(3) $|xy| \leq P$

Since, language L is regular, it must have pumping length P.

Let $P = 7$

Let, $S \in L = a^P$

$= a^7$

$=$ aaaaaaa

Dividing S into x, y and z, we get

$$S = \underbrace{a}_{x}\ \underbrace{aa}_{y}\ \underbrace{aaaa}_{z}$$

Checking for condition I

Let $i = 2$

Then, $S = xy^2z$

$=$ a aaaa aaaa

$= a^9 \notin L$

Since, 9 is not a prime number. Therefore, on pumping i.e. xy^iz, condition 1 fails. Though condition (2) and (3) are satisfied.

$|y| = 2 > 0$

$|xy| = 3 < 7$

Here, assuming language L as regular language doesn't follow all three conditions of pumping lemma to be regular language. So, it contradicts our assumption. Therefore, language L is not regular.

Closure

~~Closs~~ properties of regular sets or regular languages

~~(1) Union~~

If R_1 and R_2 are REs then R_1+R_2, $R_1 \cdot R_2$, R^* are also RE

Let L_1 and L_2 be two RE such that $L_1 \in M_1$ and $L_2 \in M_2$ where M_1 and M_2 are respective finite state machine for L_1 and L_2.

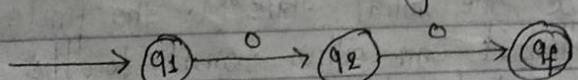
Let $M_1 = \{Q_1, \Sigma_1, \delta_1, q_1, f_1\}$ &
$M_2 = \{Q_2, \Sigma_2, \delta_2, q_2, f_2\}$

(1) for union

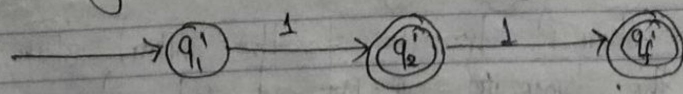
we have to show that the union of L_1 and L_2 ie, L_1+L_2 is also regular. To show that we have to design a finite automata such that it processes L_1+L_2.

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

Let $L_1 = 00$ is a regular then ϵNFA model is:

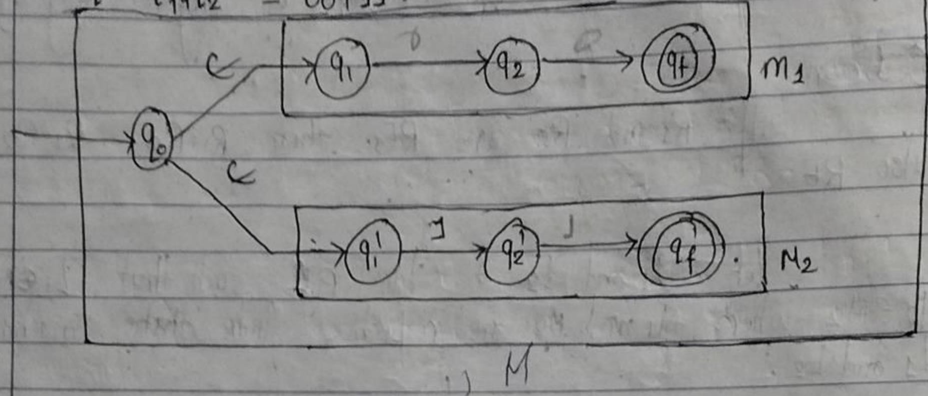


Similarly, $L_2 = 11$ is a regular then model is:



Now, the union is:

$L = L_1 + L_2 = 00 + 11$.



Let $M = \{Q, \Sigma, \delta, q_0, f\}$
where,

Q = finite set of states $= Q_1 \cup Q_2 \cup q_0$
$\Sigma = \Sigma_1 \cup \Sigma_2$
$\delta = \delta_1 \cup \delta_2 \cup \{\delta(q_0, \epsilon) \rightarrow q_1, \delta(q_0, \epsilon) \rightarrow q_1'\}$
q_0 = initial state $= q_0$
f = final state $= F_1 \cup F_2 = \{q_f, q_f'\}$

This finite automata M processes the union of L_1 and L_2. Let ~~of~~ q_0 be the initial state of M. The finite automata M is designed such that it can either transit to the initial state of M_1 or to the initial state of M_2 by non-deterministically consuming empty string (ϵ).

When it transits to the initial state of M_1, it initiates the processing of M_1 and, when it transmits to the initial state of M_2 it ~~mi~~ ~~mit~~ imitates the processing of M_2 resulting in the processing of $L_1 + L_2$. Hence, we can conclude that the class of language of FA is closed

(iv) **Complement**: L is RL then $\bar{L}$ is also RL. $\boxed{\bar{L} = \Sigma^* - L}$

Let L = set of all strings containing 1100 as substring.

Then $\bar{L}$ = " " not " 1100 as substring.

(v) ~~Intersection~~ under union: If L_1 & L_2 are two RL. then $L_1 \cap L_2$ is also a R.L.

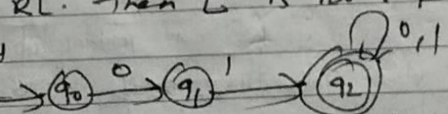
$L_1 = a^*$. $L_2 = b^*$ the $L_1 \cap L_2 = a^* \cap b^*$

~~for concatenation:~~

We have to show that

(vi) **Reversal**: If L is a RL. then L^R is also a RL.

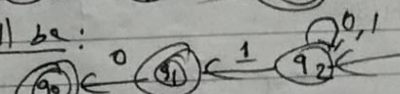
Let assume FA as



$L = (0, 01, 001)$

$L^R = (0, 10, 110)$

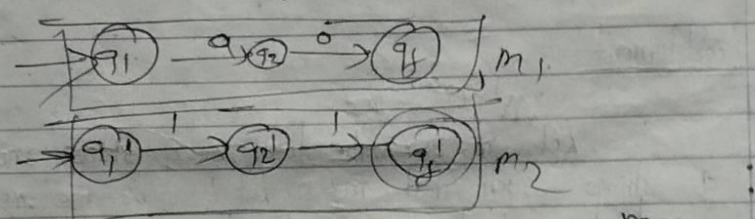
∴ The L^R will be:

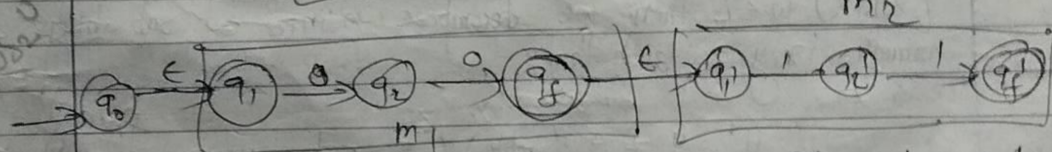


(2) **concatenation**: →

$L_1 = 00$, $L_2 = 11$

∴ $L_1 L_2$ is also Regular Expression. i.e. 00.11





$\delta: \delta_1 \cup \delta_2 \cup \{\delta(q, \epsilon) \to q_1, \delta(q_f, \epsilon) \to q_1'\}$

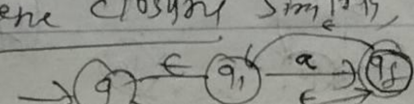
q_0 — initial state

Q — Finite set of states $Q_1 \cup Q_2 \cup q_0$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$F = F_1 \cup F_2 = \{q_f, q_f'\}$

(3) Kleene closure similarly, $L_1 = a$ is regular, $L_1^* = a^*$ is also re



$F = F_1$, q = initial state

$\delta: \delta_1 \cup \{\delta(q_f, \epsilon) \to q_1, \delta(q, \epsilon) \to q_f, \delta(q_f, \epsilon) \to q_1$