Closure properties of Regular Languages :-

If certain languages are regular, and a language L is formed from them by certain operations (Ex. L is the union of two regular languages), then L is also Regular.

So, the class of Regular Languages are closed under the operations mentioned below —

i) The Union of two Regular languages is Regular.

ii) The Intersection of two Regular Languages is Regular.

iii) The complement of a Regular Language is Regular.

iv) The Difference of two Regular languages is Regular.

v) The Reversal of a Regular Language is Regular.

vi) The Closure of a Regular Language is Regular.

vii) The Concatenation of Regular Languages is Regular.

viii) A Homomorphism (substitution of strings for symbols) of a Regular Language is Regular.

ix) The Inverse Homomorphism of a Regular Language is Regular.

* The Family of Regular Language is closed under Union, Concatenation and Star-closure

— If L_1 and L_2 are regular, then there exist regular expression r_1 and r_2 such that,

$$L_1 = L(r_1) \text{ and } L_2 = L(r_2)$$

But we know, when r_1 and r_2 are regular expression then,

$r_1 + r_2$, $r_1 \cdot r_2$ and r_1^* are also regular expression denoting languages $L_1 \cup L_2$, $L_1 L_2$ and L_1^* respectively.

So, they are closed under union, concatenation and star-closure.

Hence proved.

---

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

— To show the closure under complement let, $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L. Then DFA

$$\bar{M} = (Q, \Sigma, \delta, q_0, Q - F) \text{ accepts } \bar{L}.$$

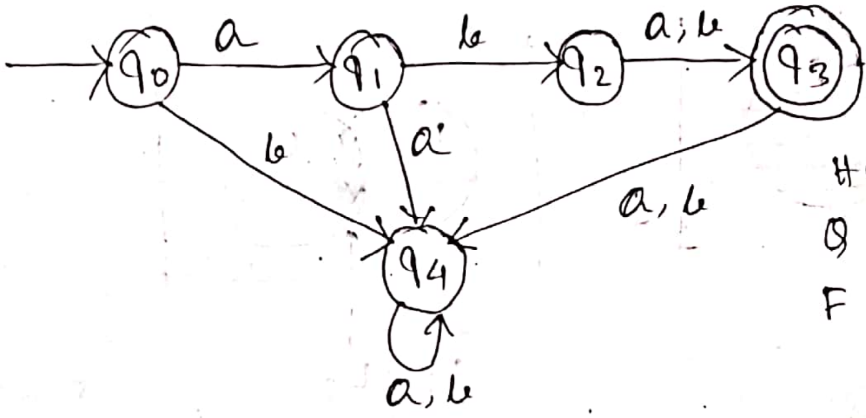
That is $\bar{M}$ is exactly like M, but the accepting/final states of M become non-accepting states of $\bar{M}$ and vice-versa.

Then a string w is in $L(\bar{M})$ iff $\hat{\delta}(q_0, w)$ is in $Q - F$, which occures iff w is not in $L(M)$.

Scanned with CamScanner

Ex An FA that accepts only strings aba and abb is shown in the figure. Find FA for the complement of L, i.e. accept every string other than aba and abb.

— The FA which accepts only aba and abb is – $M = (Q, \Sigma, \delta, q_0, F)$



Here,
$Q = \{q_0, q_1, q_2, q_3, q_4\}$
$F = \{q_3\}$

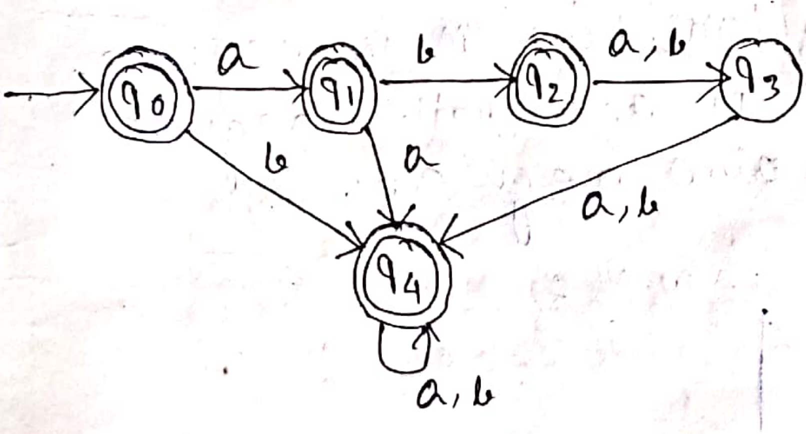
Now, applying the theorem Closure under complement,

$\overline{M} = (Q, \Sigma, \delta, q_2 ; Q - F)$

$\therefore \quad Q = \{q_0, q_1, q_2, q_3, q_4\}$

$Q - F = \{q_0, q_1, q_2, q_3, q_4\} - \{q_3\}$
$\quad = \{q_0, q_1, q_2, q_4\}$

$\therefore$ The FA is,

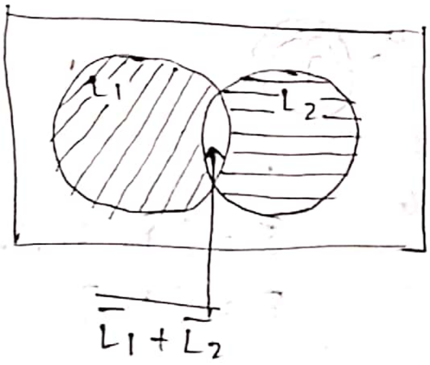


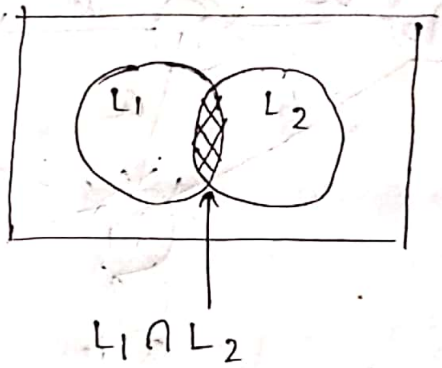
It accepts all the strings except aba and abb.

* The set of Regular Languages is closed under Intersection.

— Let, L_1 and L_2 are regular languag.
Then, according to D'morgans Theorem,

$$L_1 \cap L_2 = \overline{\overline{L_1} + \overline{L_2}}$$





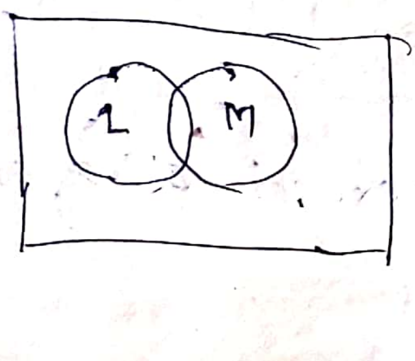
Now, L_1 and L_2 are regular, so $\overline{L_1}$ and $\overline{L_2}$ is also regular, and so $\overline{L_1} + \overline{L_2}$ and so $\overline{\overline{L_1} + \overline{L_2}}$ is also regular.
which implies $L_1 \cap L_2$ is regular.

* If L and M ~~and~~ are regular the L-M is also regular.

— We know,

$$L - M = L \cap \overline{M}.$$

Here, $\overline{M}$ is regular, bcoz M is regular.
so, when L and $\overline{M}$ are both regular the $L \cap \overline{M}$ is also regular.



Reversal

The reversal of a string $a_1 a_2 \cdots a_n$ is the string written backwards, that is $a_n a_{n-1} \cdots a_2 a_1$. We use w^R for the reversal of string w.

Ex – $(0010)^R = (0100)$

$\Lambda^R = \Lambda$

If L is a language then, the reverse is L^R.

Let, $L = \{001, 10, 111\}$

$\therefore L^R = \{100, 01, 111\}$

Automata for L^R

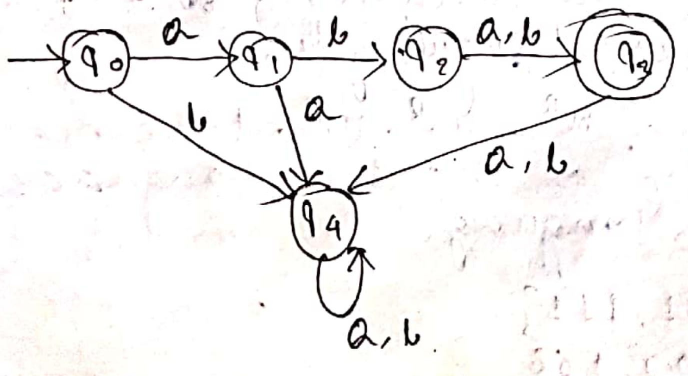
Given a language L, the reversal of L, which is L^R. Now, with Nondeterminism and Λ-transition we can construct an automaton for L^R from $L(M)$ –

i) Reverse all the arcs in the transition diagram for M.

ii) Make the start state of M be the only accepting state for the new automaton.

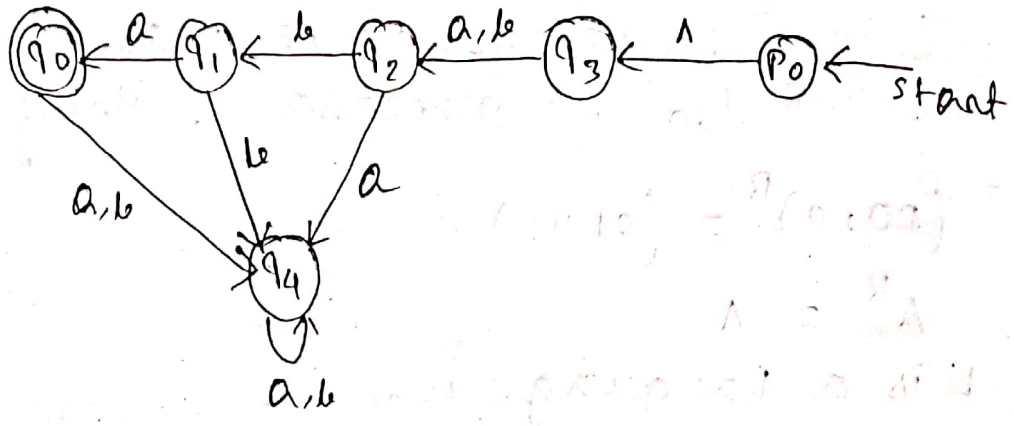
iii) Create a new start state p_0 with transition on Λ to all the accepting states of M.

Ex Let, $L(M)$ = Accepts only aba and abb.



So, L^R will accepts only ~~aaa~~ aba and bba.

∴ Automaton for L^R is –



The above automaton is reverse of $L(m)$. It accepts only {aba, bba}.

---

* If L is regular the L^R is also regular.

Let us assume that L is defined by the regular expression r. Now, we show that there is another regular expression r^R, such that $L(r^R) = (L(r))^R$, that is the language of r^R is the reversal of the language of r.

Now, If r is Λ, ϕ, or a for some symbol a, then r^R is same as r. bcoz,

$\{\Lambda\}^R = \{\Lambda\}$, $\{\phi\}^R = \{\phi\}$ and $\{a\}^R = \{a\}$

Now, there are three cases depending on the forms of the r,

i) $r = r_1 + r_2$. Then $r^R = r_1^R + r_2^R$. The justification is that the reversal of the union of two languages is obtained by computing the reversal of the two languages and taking the union of those languages.

ii) $r = r_1 . r_2$. Then $r^R = r_2^R . r_1^R$. Here we reverse the order of the two languages, as reversing the language themselves.

Ex- Let, $L(r_1) = \{01, 111\}$

$L(r_2) = \{00, 10\}$

then $L(r_1 r_2) = \{0100, 0110, 11100, 11110\}$

Now, $(L(r_1 r_2))^R = \{0010, 0110, 00111, 01111\}$

$= (L(r_2))^R \cdot (L(r_1))^R$

$= \{00, 01\}\{10, 111\}$

$= \{0010, 00111, 0110, 01111\}$

$= (L(r_1 r_2))^R$

iii) $r = r_1^*$. Then $r^R = (r_1^R)^*$. The justification is that any string w in $L(R)$ can be written as $w_1; w_2, \ldots, w_n$, where each w_i is in $L(r)$.

But, $w^R = w_n^R \cdot w_{n-1}^R \cdots w_1^R$

each w_i^R is in $(L(r))^R$.

So, w^R is in $(r_1^R)^*$

Homomorphism

A string homomorphism is a function on strings that works by substituting a particular string for each symbol.

Let, Σ and Σ' are alphabets. Then the function,

$$h : \Sigma \longrightarrow \Sigma' \text{ is called a homomorphism.}$$

If, $w = x_1 x_2 \cdots x_n$ then,

$$h(w) = h(x_1)\, h(x_2) \cdots h(x_n)$$

If L is a language on Σ, then its homomorphic image is defined as,

$$h(L) = \{h(w) : w \in L\}$$

Ex Let $\Sigma = \{0, 1\}$ and $\Sigma' = \{0, 1, 2\}$ and defined h by, $h(0) = 01$, $h(1) = 112$. Then find $h(010)$ and homomorphic image of $L = \{00, 010\}$.

— Given, $\Sigma = \{0, 1\}$ and $\Sigma' = \{0, 1, 2\}$

Now, $h(0) = 01$, $h(1) = 112$.

So, $h(010) = 0111201$

The homomorphic image of $L = \{00, 010\}$ is the language,

$$h(L) = \{0101, 0111201\}$$

---

* Let h be a homomorphism. If L is a regular, then its homomorphic image $h(L)$ is also regular. The family of regular languages is closed under arbitrary homomorphisms.

— Let $L = L(R)$ for some regular expression R. In general if r is a regular expression with symbol Σ, let $h(r)$ be the expression we obtain by replacing each symbol a of Σ in r by $h(a)$. we claim that $h(R)$ define the language $h(L)$.

Now, if r is Λ or ϕ, then $h(r)$ is same as r, since h doesnot affect the string Λ or the language ϕ. Thus $L(h(r)) = L(r)$. However if r is ϕ or Λ, then $L(r)$ contains either no strings or a string with no symbol respectively. Thus $h(L(r)) = L(r)$ in either case. we conclude $L(h(r)) = L(r) = h(L(r))$.

In other case if $r = a$ for some symbol a in Σ, $L(r) = \{a\}$, so $h(L(r)) = \{h(a)\}$. Also, $h(r)$ is the regular expression that is the strings of symbols $h(a)$. Thus $L(h(r))$ is also $\{h(a)\}$, and we conclude, $L(h(r)) = h(L(r))$.

There are three cases –

i) let $r = r_1 + r_2$

$\therefore h(r) = h(r_1 + r_2) = h(r_1) + h(r_2)$.

we know, $L(r) = L(r_1) \cup L(r_2)$ and

$L(h(r)) = L(h(r_1) + h(r_2)) = L(h(r_1)) \cup L(h(r_2))$ ---- (a)

Again, h is applied to a language by application to each of its string individually. So we can write,

$$h(L(R)) = h(L(R_1) \cup L(R_2)) = h(L(R_1)) \cup h(L(R_2)) \quad \text{---} (b)$$

But, we know according to induction hypothesis,

$$L(h(R_1)) = h(L(R_1)) \quad \text{and}$$
$$L(h(R_2)) = h(L(R_2))$$

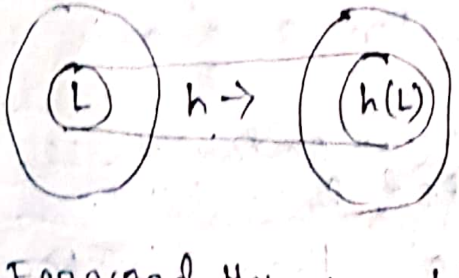
Thus the final expression for equation (a) and (b) are equivalent. So,

$$L(h(R)) = h(L(R))$$

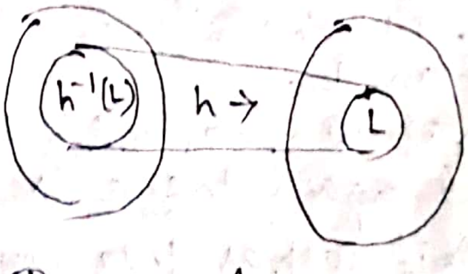
ii) The second case is for concatenation
iii) Third case is for closure.

Inverse Homomorphism

Let h is a homomorphism from some alphabet Σ to strings in another (possibly the same) alphabet T. Let L be a language over alphabet T. Then $h^{-1}(L)$ is the set of strings w in Σ^* such that h(w) is in L.



Forward Homomorphism



Backward or Inverse Homomorphism

$\therefore$ If $h : \Sigma \to T$, $L = L(T)$ then

$$h^{-1}(L) = \{w : w \in \Sigma^* \text{ and } h(w) \in T^*\}$$

Scanned with CamScanner

Ex
Let, L be the language of regular expression $(00+1)^*$.

Let h be the homomorphism defined by,
$h(a) = 01$, $h(b) = 10$

Now, we claim that $h^{-1}(L)$ is the language of all regular expression $(ba)^*$, that is all strings of repeating ba pairs.

Now, P.T h(w) is in L iff w is of the form baba---ba :-

Suppose w is n repetitions of ba for some $n \geq 0$. Note that $h(ba) = 1001$, so h(w) is n repetitions of 1001. Now, $1001 \in L$. Therefore any repetition of 1001 is also formed by 1 and 00 segments and is in L. Thus, h(w) is in L.

Now, we'll prove,
if and only if h(w) is in L, then w is of the form baba----ba :-

To prove the above we prove the contrapositive of the statements written below :-

i) if w begins with a, then h(w) begins with 01, It therefore has an isolated 0, and is not in L.

ii) if w ends with b, then h(w) ends with 10, and again there is an isolated 0 in h(w).

iii) if w has two consecutive a's, then h(w) has a substring 0101. Here too, there is an isolated 0 in w.

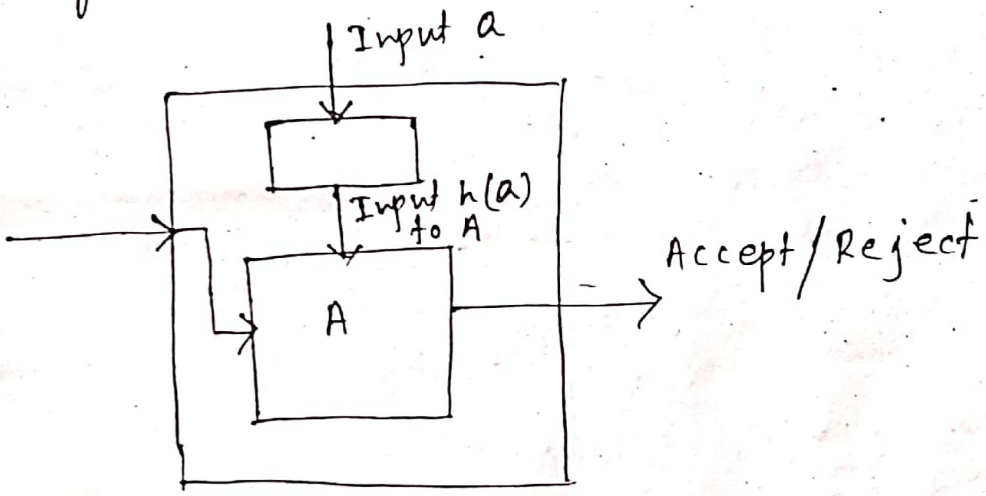
iv) Likewise, if w has two consecutive b's, then h(w) has substring 1010 and has an isolated 0.

Now the logical OR of above four statements is equivalent to the statement " w is not of

the form baba--- ba ...
The statement is contrapositive of the statement we wanted:
"If $h(w)$ is in L, then w is of the form baba---ba".

---

* If h is a homomorphism from alphabet Σ to alphabet T, and L is a regular language over T, then $h^{-1}(L)$ is also a regular language.
— The proof starts with a DFA A for L. We construct from A and h a DFA for $h^{-1}(L)$. This DFA uses the states of A but translates the input symbol according to h before deciding on the next state.



The DFA for $h^{-1}(L)$ applies h to its input, and then simulates the DFA for L

Let, L be L(A), where DFA $A = (Q, T, \delta, q_0, F)$.
Define a DFA B,
$$B = (Q, \Sigma, \gamma, q_0, F) \text{ where}$$
transition function γ is constructed by the rule $\gamma(q, a) = \hat{\delta}(q, h(a))$, That is the transition B makes on input a is the result of the sequence of transitions that A makes on the strings of symbols $h(a)$.
Now, by the method of induction on $|w|$ $\hat{\gamma}(q_0, w) = \hat{\delta}(q_0, h(w))$. Since the accepting state of A and B are same, B accepts w iff A accepts $h(w)$. In other words we can

say B accepts exactly those strings w that are in $h^{-1}(L)$.
