Lecture Notes
on
Theory of Automata and Formal Languages

Unit-2

Dharmendra Kumar
(Associate Professor)
United College of Engineering and Research, Prayagraj

May 10, 2023

Contents

Chapter 1

Regular Expression and Regular Languages

1.1 Regular expression

1.1.1 Definition

Regular expression is defined recursively as follows:-
(1) ϕ, $\bar{\epsilon}$ and $\bar{a}$ are regular expression, where $a \in \Sigma$. These are called primitives regular expressions.
(2) If $\bar{r_1}$ and $\bar{r_2}$ are two regular expressions then $\bar{r_1} + \bar{r_2}$ and $\bar{r_1}\bar{r_2}$ are also regular expressions.
(3) If $\bar{r}$ are regular expression then $\bar{r}^*$ and $(\bar{r})$ are also regular expressions.
(4) Any expression derived from the step 1 to 3 are also regular expression.

Example: $(\bar{a} + \bar{b} + \bar{c})^*.(\bar{c}+\phi)$ is a regular expression.

1.1.2 Regular language or Regular set

Each regular expression represents a set of elements. This set is said to be regular set.

1.1.3 Some examples

1. Represent the following sets by regular expression:-

   (a) $\{1^{2n} \,!\, n > 0\}$

   (b) $\{w \in \{a, b\}^*!$ w has only one a$\}$

   (c) The set of all strings over {0,1} which has most two 0's.

   (d) $\{a^2, a^5, a^8,$ ....$\}$

   (e) $\{1^n$! n is divisible by 2 or 3 or n=5$\}$

   (f) The set of all strings over {a,b} beginning and ending with a.

(g) The set of all strings over {a,b} in which the number of occurrences of a is divisible by 3.

(h) The set of all strings over {a,b} with three consecutive b's.

2. Find the regular expression for the following:-

(a) L = $\{a^n b^m \mid n \geq 1, m \geq 1, mn \geq 3\}$

(b) L = $\{a^n \mid n \geq 0, n \neq 3\}$

(c) L = $\{a^{2n} b^{2m+1} \mid n \geq 0, m \geq 0\}$

(d) L = $\{a^n b^m \mid$ n+m is even$\}$

3. Write regular expression for the following language over alphabet {0,1}.

(a) All strings ending in 01.

(b) All strings not ending in 01.

(c) All strings containing an even number of 0's.

(d) All strings with at most two occurrences of the substring 00.

(e) All strings not containing the substring 10.

4. Write regular expression for the following language.

(a) L = {w$\in \{0,1\}^* \mid$ w has at least one pair of consecutive 0's }

(b) L = {w$\in \{0,1\}^* \mid$ w has no pair of consecutive 0's }

(c) L = {w $\mid$ w$\in \{a,b\}^*$ and $\mid w \mid \ mod\ 3 = 0\}$

(d) L = {w $\mid$ w$\in \{a,b\}^*$ and $n_a(w)\ mod\ 3 = 0\}$

5. Find the set corresponding to the following regular expressions:-

(a) $(aa)^*(bb)^*b$

(b) $(0+1)^*00(0+1)^*$

(c) $((0+1)(0+1)^*)^*00(0+1)^*$

(d) $(a+b)^*(a+bb)$

(e) $(aa)^*(bb)^*b$

1.1.4 Equivalent regular expressions

Two regular expressions $\bar{p}$ and $\bar{q}$ are said to be equivalent if they represent the same set of strings.

1.1.5 Identities for regular expressions

Let p, q, r are all the regular expressions. Then

(1) ϕ + r = r
(2) ϕ . r = ϕ = r.ϕ
(3) ϵ.r = r.ϵ = r
(4) ϵ^* = ϵ and ϕ^* = ϵ (5) r+r = r
(6) $r^*r^* = r^*$
(7) $rr^* = r^*r$
(8) $(r^*)^* = r^*$
(9) $\epsilon\ +\ rr^* = r^* = \epsilon\ +\ r^*r$
(10) $(p.q)^*p = p.(p.q)^*$
(11) $(p+q)^* = (p^*q^*)^* = (p^*+q^*)^*$
(12) $(p+q).r = p.r + q.r$ and $r.(p+q). = r.p + r.q$

1.2 ARDEN's Theorem

Statement: Let P and Q be two regular expressions over Σ. If P does not contain ϵ , then the following equation in R,
R = Q + RP ....(1)
has a unique solution R = QP^*.
Proof:
Q+RP = Q + QP*P
= Q($\epsilon\ +\ P^*P$)
= QP*
= R
Therefore equation (1) is satisfied when R = QP*.
Therefore R = QP* is a solution of equation (1).
To prove uniqueness of (1), replacing R by Q+RP.
R = Q +RP = Q + (Q+RP)P
= Q + QP + RP^2
= Q + QP + $(Q + RP)P^2$
= Q + QP + QP^2 + RP^3
....
....
= Q + QP +QP^2 +........+ QP^i + RP^{i+1}
= Q(ϵ + P +P^2 +........+ P^i) + RP^{i+1}
Therefore, R = Q(ϵ + P +P^2 +........+ P^i) + RP^{i+1} ....(2)
Let w∈ R and $\mid w \mid$ = i.
From equation (2), w will belong into right hand side of equation (2).
Therefore, w∈ Q(ϵ + P +P^2 +........+ P^i) + RP^{i+1}
Clearly, w∉ RP^{i+1} since $\mid w \mid$ = i.
Therefore, w ∈ Q(ϵ + P +P^2 +........+ P^i)
Therefore, w ∈ QP^*
Therefore, R⊆ QP^* ....(3)
Let w ∈ QP^*.
Then w∈QP^k for some k≥ 0.

Therefore, w $\in$ Q(ϵ + P +P^2 +........+ P^k)
Therefore, w belong into the right hand side of equation (2).
Therefore, w$\in$ R.
Therefore, $QP^* \subseteq$ R ....(4)
From (3) and (4),
R = QP^*

Example:
(a) Give a regular expression for representing the set L of strings in which every 0 is immediately followed by at least two 1's.
(b) Prove that the regular expression
r = ϵ + $1^*(011)^*(1^*(011)^*)^*$
also describes the same set of strings.

Example: Prove that
$(1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1) = 0^*1(0+10^*1)^*$

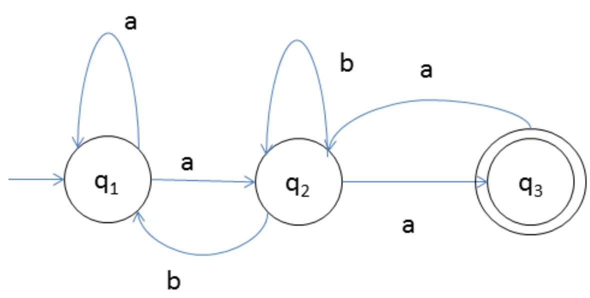
1.3 Determination of regular expression from finite automata

The following assumptions are made regarding the transition system:-
(i) The transition graph does not contain $\epsilon - move$.
(ii) It has only one initial state, say q_1.
(iii) Let all the states are q_1, q_2, q_3,..................,q_n,
(iv) Let α_{ij} denotes the regular expression representing the set of labels of edges from q_i to q_j. When there is no such edge, $\alpha_{ij} = \phi$.

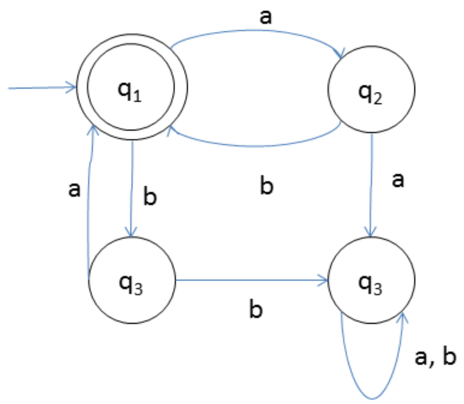
In this process to find regular expression, initially we make n equations as the following:-
$q_1 = q_1\alpha_{11} + q_2\alpha_{21} + q_3\alpha_{31} +$ ....+ $q_n\alpha_{n1} + \epsilon$
$q_2 = q_1\alpha_{12} + q_2\alpha_{22} + q_3\alpha_{32} +$ ....+ $q_n\alpha_{n2}$
....
$q_n = q_1\alpha_{1n} + q_2\alpha_{2n} + q_3\alpha_{3n} +$ ....+ $q_n\alpha_{nn}$
We solve these equations by using ARDEN's theorem . The regular expression will be the union of regular expressions corresponding to each final states.

1.3.1 Some Examples

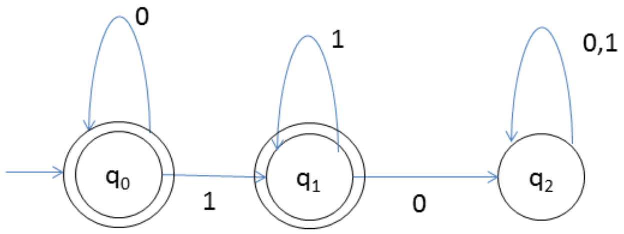
Example: Find the regular expression corresponding to the following finite automata:-



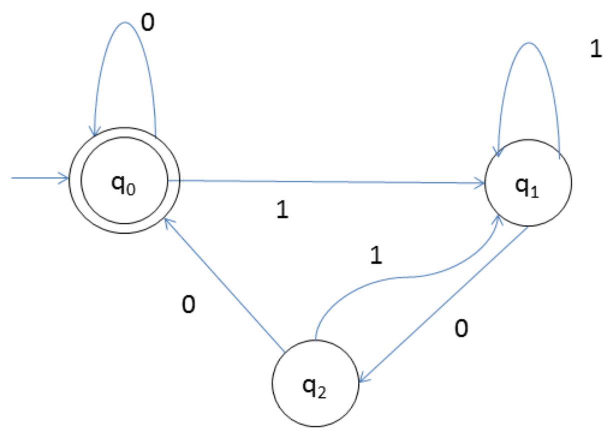
Example: Find the regular expression corresponding to the following finite automata:-



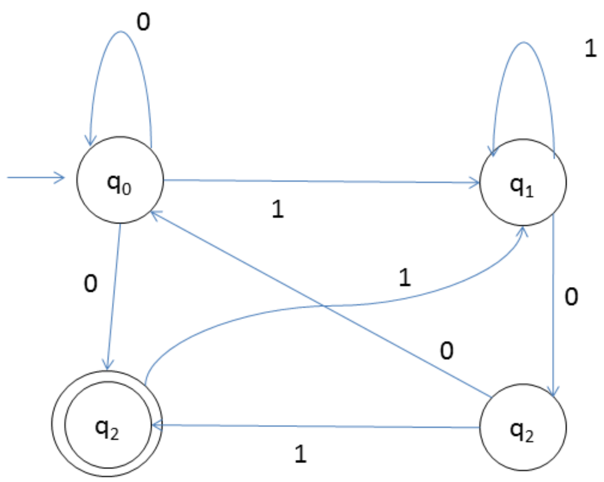
Example: Find the regular expression corresponding to the following finite automata:-



Example: Find the regular expression corresponding to the following finite automata:-

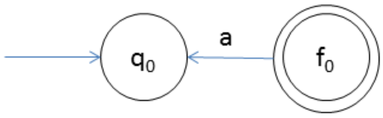


Example: Find the regular expression corresponding to the following finite automata:-

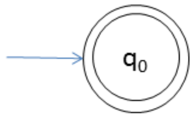


1.4 Kleene's Theorem

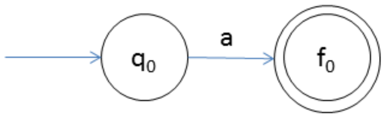
Statement: Let r be a regular expression. Then there exists some non-deterministic finite automata that accepts L(r). Consequently, L(r) is a regular language.
Proof:
We will prove this theorem by using induction method.
Apply the induction method on the no. of operators used in the regular expression.i **Basis step:** In this step, we will prove for the regular expressions which does not contain any operators.
In this case, regular expressions are
ϕ, ϵ and a, where a$\in \Sigma$.
The finite automata corresponding to the above regular expressions are:-



L(M) = ϕ



L(M) = { ε }



L(M) = { a }

Clearly, there exists a finite automata for every regular expressions which not contain any operator. Therefore, theorem is true for this case.

Induction step:

Suppose the theorem is true for the regular expressions which contain n operators. We will prove the theorem for the regular expressions that contain n+1 operators. The $(n+1)^{th}$ operator in the regular expressions may be one of the following:-

(1) + (Union operator) (2) . (Concatenation operator) (3) * (Kleene closure operator)

Case 1 Union operator$(r_1 + r_2)$**:**

Suppose the $(n+1)^{th}$ operator is $+$.

Consider the regular expressions r_1 and r_2 of n operators.

Since the theorem is true for r_1 and r_2, therefore there exists finite automata M_1 and M_2 corresponding to r_1 and r_2. Let these finite automatas are

$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, \{f_1\})$

$M_2 = (Q_2, \Sigma_2, \delta_2, q_2, \{f_2\})$

Now, we construct the finite automata corresponding to $r_1 + r_2$. This is constructed as following:-

$M = (Q, \Sigma, \delta, q_0, \{f_0\})$

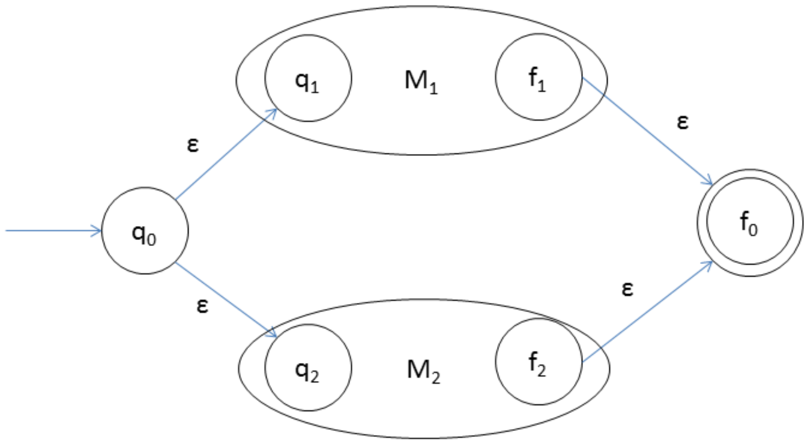
Where, Q $= Q_1 \cup Q_2 \cup \{q_0, f_0\}$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & if q \in Q_1 \ and \ a \in \Sigma_1 \\ \delta_2(q,a), & if q \in Q_2 \ and \ a \in \Sigma_2 \end{cases} \tag{1.1}$$

$\delta(q_0, \epsilon) = \{q_1, q_2\}$

$\delta(f_1, \epsilon) = \{f_0\} = \delta(f_2, \epsilon)$



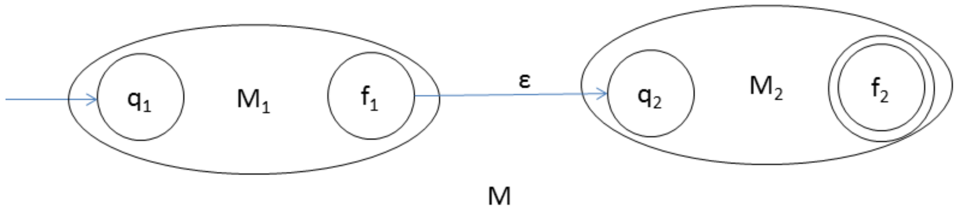
M

Now,we have to show that
L(M) = $L(M_1) \cup L(M_2)$ ....(1)
Let x$\in$ L(M) $\Leftrightarrow$ $\epsilon x \epsilon \in L(M)$
$\Leftrightarrow$ x$\in$ L(M_1) or x$\in$ L(M_2)
$\Leftrightarrow$ x$\in$ $L(M_1) \cup \in L(M_2)$
Therefore, L(M) = $L(M_1) \cup L(M_2)$
Therefore M is a finite automata which accepts the set $L(r_1) \cup L(r_2)$.
Therefore M is a finite automata for the regular expression $r_1 + r_2$.
Therefore theorem is true for this case.

Case 2 Concatenation operator($r_1.r_2$):
Suppose the $(n+1)^{th}$ operator is . .
Consider the regular expressions r_1 and r_2 of n operators.
Since the theorem is true for r_1 and r_2, therefore there exists finite automata M_1 and M_2 corresponding to r_1 and r_2. Let these finite automatas are
$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, \{f_1\})$
$M_2 = (Q_2, \Sigma_2, \delta_2, q_2, \{f_2\})$
Now, we construct the finite automata corresponding to $r_1.r_2$. This is constructed as following:-
$M = (Q, \Sigma, \delta, q_1, \{f_2\})$
Where, Q = $Q_1 \cup Q_2$
$\Sigma = \Sigma_1 \cup \Sigma_2$

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & if q \in Q_1 \ and \ a \in \Sigma_1 \\ \delta_2(q,a), & if q \in Q_2 \ and \ a \in \Sigma_2 \end{cases} \tag{1.2}$$

$\delta(f_1, \epsilon) = \{q_2\}$

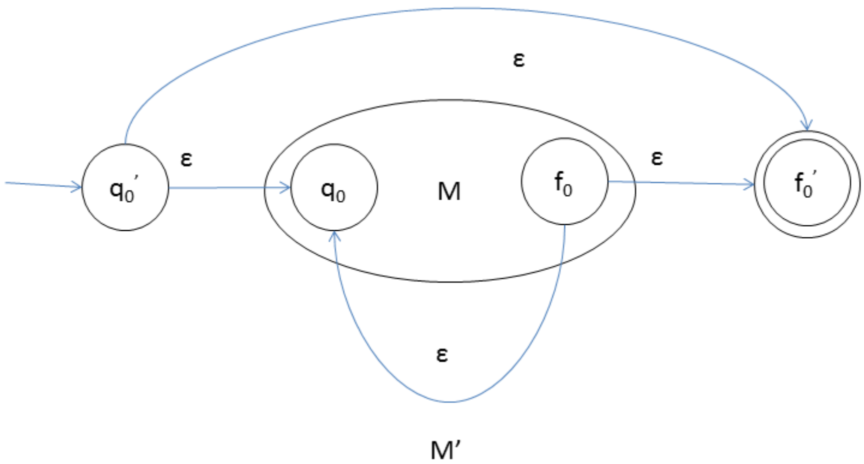


Now,we have to show that
L(M) = $L(M_1).L(M_2)$ ....(2)
Let x∈ L(M) ⇔ $x_1 \epsilon x_2 \in L(M)$ (Let x=$x_1.x_2$)
⇔ $x_1 \in$ L(M_1) and $x_2 \in$ L(M_2)
⇔ $x_1.x_2 \in L(M_1).L(M_2)$
⇔ x ∈ $L(M_1).L(M_2)$
Therefore, L(M) = $L(M_1).L(M_2)$
Therefore M is a finite automata which accepts the set $L(r_1).L(r_2)$.
Therefore M is a finite automata for the regular expression $r_1.r_2$.
Therefore theorem is true for this case.

Case 3 Kleene closure operator(r^*):
Suppose M is a finite automata corresponding to regular epression r.
$M = (Q, \Sigma, \delta, q_0, \{f_0\})$
Now, we construct finite automata corresponding to r^*. This is constructed as following:-
$M' = (Q', \Sigma, \delta', q_0', \{f_0'\})$
Where, Q = $Q \cup \{q_0', f_0'\}$

$$\delta'(q, a) = \begin{cases} \delta(q, a), & if q \in Q \ and \ a \in \Sigma \end{cases} \tag{1.3}$$

$\delta'(q_0', \epsilon) = \{q_0, f_0'\} = \delta'(f_0, \epsilon)$



Now,we have to show that
L(M') = $(L(M))^*$ ....(3)

Let x$\in$ L(M') $\Leftrightarrow$ $\epsilon x\epsilon \in L(M')$ (Let x=$x_1.x_2$)
$\Leftrightarrow$ $\epsilon x_1\epsilon x_2\epsilon.........\epsilon x_n\epsilon \in L(M')$ (Let x=$x_1.x_2........x_n$)
$\Leftrightarrow$ $x_i \in L(M), \quad \forall i = 1, 2, ..., n$
$\Leftrightarrow$ $x_1.x_2........x_n \in (L(M))^n$
$\Leftrightarrow$ $x \in (L(M))^n$
$\Leftrightarrow$ $x \in (L(M))^*$ (since $(L(M))^n) \subseteq (L(M))^*$
Therefore, L(M) = $(L(M))^*$
Therefore M is a finite automata which accepts the set $L(r_1).L(r_2)$.
Therefore M is a finite automata for the regular expression $r_1.r_2$.
Therefore theorem is true for this case.

Since in all the three cases, theorem is true, therefore theorem is true for induction step also.
Therefore, for any regular expression, there exists a finite automata.
Now the theorem is proved.

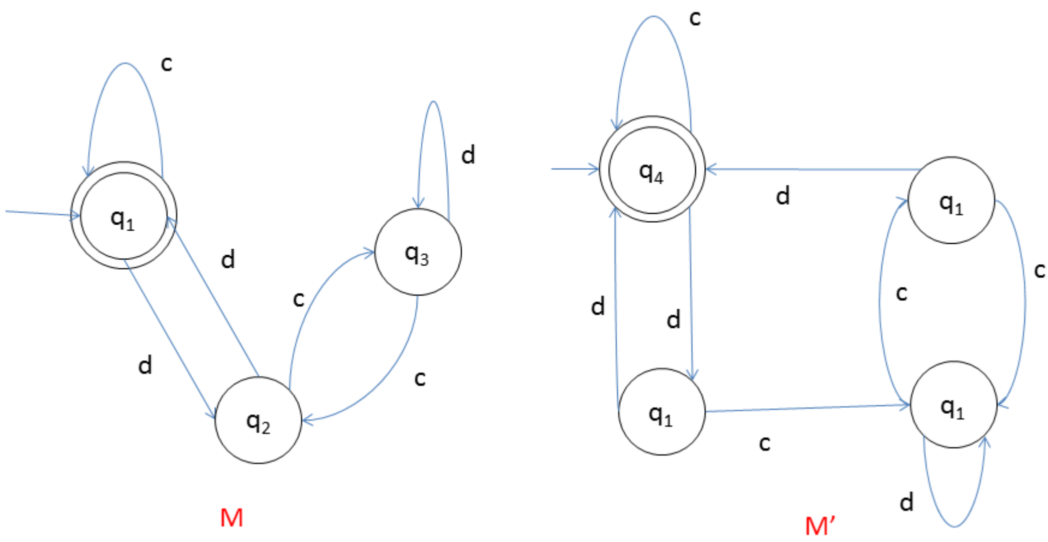
1.5 Construction of finite automata from regular expression

Example: Construct finite automata for the regular expressions:-

1. r = $(a+b)^*(aa+bb)(a+b)^*$
2. r = $10 + (0+11)0^*1$
3. r = $(a+b)^*b(a+bb)^*$
4. r = $aa^* + aba^*b^*$

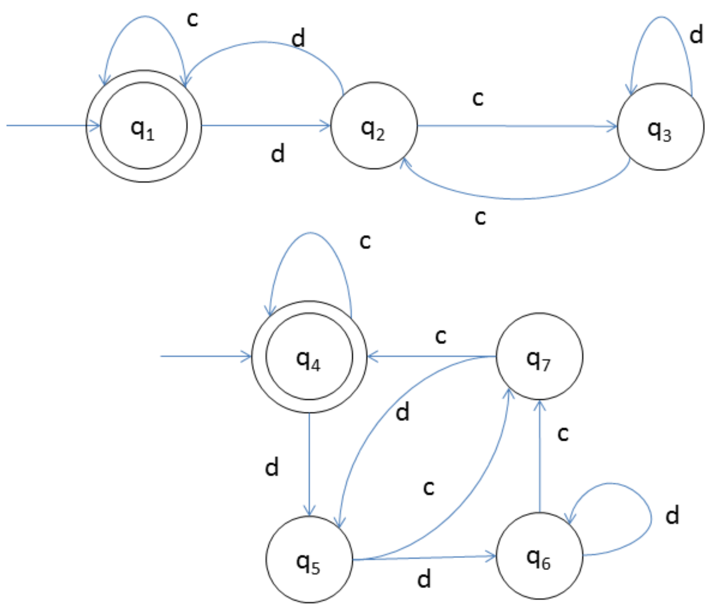
1.6 Equivalence of two finite automata

Example: Consider two DFA M and M':



Determine whether M and M' are equivalent.

Example: Show that following automata M_1 and M_2 are not equivalent.



1.7 Right and Left linear grammars

A grammar is said to be right linear grammar if all production rules are of the following form:-

A $\rightarrow$ xB or A $\rightarrow$ x, where A,B $\in$ V and x$\in \Sigma^*$

A grammar is said to be left linear grammar if all production rules are of the following form:-

A $\rightarrow$ Bx or A $\rightarrow$ x, where A,B $\in$ V and x$\in \Sigma^*$

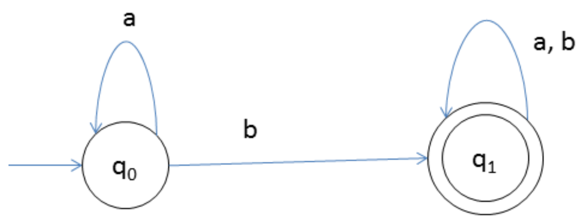
A regular grammar is one that is either right linear or left linear.

1.7.1 Construction of regular grammar from the given DFA

Suppose the given DFA is
$M = (\{q_0, q_1, ..., q_n\}, \Sigma, \delta, q_0, F)$
Now we construct the grammar G for M as
G $= (\{Q_0, Q_1, ..., Q_n\}, \Sigma, Q_0, P)$

Where P is defined as
(i) $Q_i \rightarrow aQ_j \in P$ if $\delta(q_i, a) = q_j \notin F$
(ii) $Q_i \rightarrow aQ_j$ *and* $Q_i \rightarrow a \in P$ if $\delta(q_i, a) = q_j \in F$

Example: Find the regular grammar for the following DFA



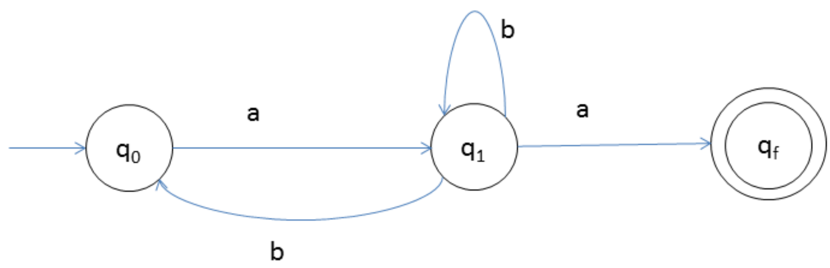
Solution: Since the number of states are 2, therefore number of variables in the grammar will be 2. Let these variables are Q_0 and Q_1 corresponding to states q_0 and q_1. The starting symbol will be Q_0.
The production rules of the grammar are the following:-
$Q_0 \rightarrow aQ_0$
$Q_0 \rightarrow b/bQ_1$
$Q_1 \rightarrow a/b/aQ_1/bQ_1$

1.7.2 Construction of a FA from given regular grammar

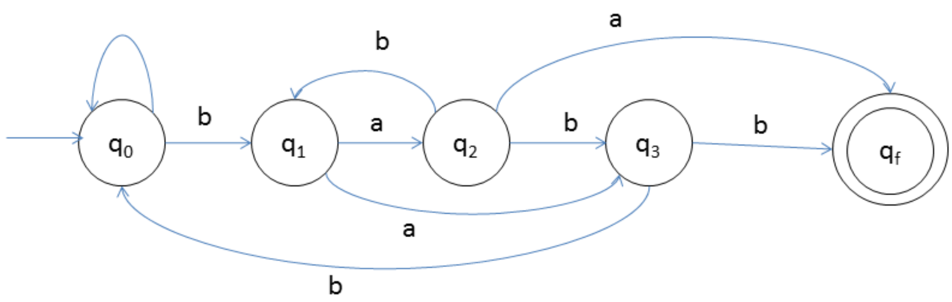
G $= (\{A_0, A_1, ..., A_n\}, \Sigma, A_0, P)$
We construct finite automata M as
$M = (\{q_0, q_1, ..., q_n, q_f\}, \Sigma, \delta, q_0, \{q_f\})$

and δ is defined as
(i) If $A_i \rightarrow aA_j$ then $\delta(q_i, a) = q_j$
(i) If $A_i \rightarrow a$ then $\delta(q_i, a) = q_f$

Example: Let G $= (\{A_0, A_1\}, \{a, b\}, A_0, P)$
Where P is
$A_0 \rightarrow aA_1,$ $A_1 \rightarrow bA_1,$ $A_1 \rightarrow a,$ $A_1 \rightarrow bA_0,$
Construct finite automata accepting L(G).
Solution:



Example: Let G $= (\{A_0, A_1, A_2, A_3\}, \{a, b\}, A_0, P)$
Where P is
$A_0 \rightarrow aA_0/bA_1,$ $A_1 \rightarrow aA_2/aA_3,$ $A_2 \rightarrow a/bA_1/bA_3,$ $A_3 \rightarrow b/bA_0,$
Construct finite automata accepting L(G).
Solution:



1.8 Pumping Lemma

Let L be an infinite regular language. Then there exists some positive integer n such that any x∈ L with x≥n, can be decomposed as

x=uvw

with $\mid uv \mid \leq n$ and $\mid v \mid \geq 1$,
such that
$uv^iw \in$ L, ∀ i = 0,1,2,3,......................

1.8.1 Application of Pumping Lemma

It is used to show that a language is not regular.
Note: Pumping lemma is based on the principle of pigeonhole principle.
Pigeonhole Principle
If we put n objects into m boxes and n>m, then at least one box must have more than one item in it.

1.8.2 Some Examples

1. Show that L = $\{a^{i^2} \; ! \; i \geq 1\}$ is not regular.
2. Show that L = $\{a^nb^n \; ! \; i \geq 1\}$ is not regular.
3. Show that L = $\{ww^R \; ! \; w \in \{a,b\}^*\}$ is not regular.
4. Show that L = $\{a^p \; !$ p is a prime number $\}$ is not regular.
5. Show that L = $\{a^{n!} \; ! \; n \geq 0\}$ is not regular.
6. Show that L = $\{(ab)^na^k \; ! \; n > k, \; k \geq 0\}$ is not regular.

1.9 Closure properties of regular languages

Theorem: Show that class of regular languages is closed under union operation.

or

If L_1 and L_2 are two regular set then $L_1 \cup Ł_2$ is also regular set.

Theorem: Show that class of regular languages is closed under concatenation operation.

or

If L_1 and L_2 are two regular set then $L_1.Ł_2$ is also regular set.

Theorem: Show that class of regular languages is closed under Kleene closure operation.

or

If L is regular set then L^* is also regular set.

Theorem: Show that class of regular languages is closed under complement operation.

or

If L is regular set then $\bar{L}$ is also regular set.

Proof:

Suppose L is a regular set. Since L is regular set then there exists a finite automata. Let this finite automata is

M = $(Q, \Sigma, \delta, q_0, F)$.

Now we construct the finite automata M' as following:-

$M' = (Q, \Sigma, \delta, q_0, F')$.

Where F' = Q-F

Now we have to show that L$(M') = \bar{L}(M) = \Sigma^* - L(M)$

Let x$\in L(M') \;\Leftrightarrow\; \delta(q_0, x) \in F'$

$\Leftrightarrow\; \delta(q_0, x) \in Q - F$

$\Leftrightarrow\; \delta(q_0, x) \notin F$

$\Leftrightarrow\; x \notin L(M)$

$\Leftrightarrow\; x \in \bar{L}(M)$

Therefore L$(M') = \bar{L}(M)$

Clearly M' is a finite automata accepting $\bar{L}$. Therefore $\bar{L}$ is a regular set.

Theorem: Show that class of regular languages is closed under intersection operation.

or

If L_1 and L_2 are two regular set then $L_1 \cap L_2$ is also regular set.

Proof:

$L_1 \cap Ł_2 = (\overline{\bar{L_1} \cup \bar{L_1}}) = \Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$ ....(1)

Since L_1 and L_2 are two regular set therefore $(\Sigma^* - L_1)$ and $(\Sigma^* - L_1)$ are also regular sets according to the previous theorem.

Since $(\Sigma^* - L_1)$ and $(\Sigma^* - L_1)$ is regular therefore $(\Sigma^* - L_1) \cup (\Sigma^* - L_1)$ is also regular

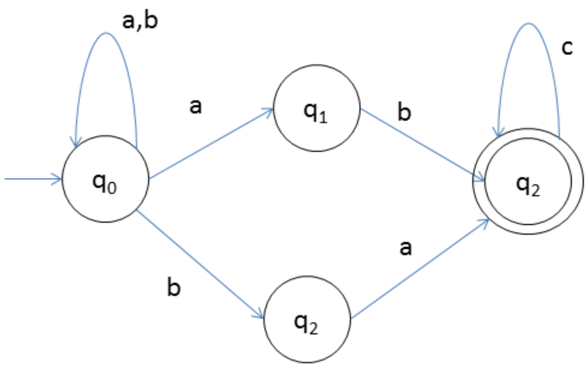
set.
According to the previous theorem, $\Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$ is also regular.
Now from equation (1), $L_1 \cap L_2$ is also regular set. Now it is proved.

1.10 AKTU Examination Questions

1. For the given language $L_1 = \epsilon$, $L_2 = \{a\}$, $L_3 = \phi$. Compute $L_1 L_2^* \cup L_3^*$.

2. Write regular expression for set of all strings such that number of a's divisible by 3 over Σ = {a,b}.

3. Find the regular expression corresponding to the finite automata given bellow:

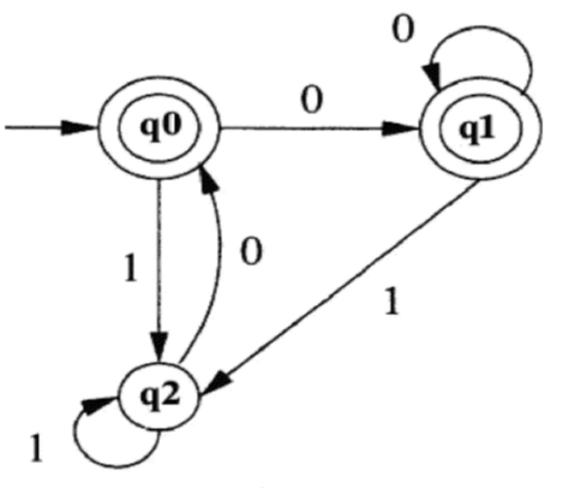


4. Prove that the following Language L = $\{a^n b^n\}$ is not regular.

5. Explain the Closure properties of regular expression.

6. Design a regular expression that accepts all the strings for input alphabet {a,b} containing exactly 2 a's.

7. State recursive definition of regular expression and construct regular expression corresponding to the following state transition diagram:-



8. State pumping lemma for regular sets. Show that L = $\{a^p$! p is a prime number$\}$ is not regular set.

9. Discuss closure properties of regular languages under the operations: concatenation, union, intersection and complement.

10. Give the regular expression for set of all strings over 0,1 containing exactly three 0's.

11. Write a regular expression to denote a language L which accepts all the strings that begin or end with either 00 or 11.

12. State closure properties of regular languages. Also Prove that regular languages are closed under intersection and difference.

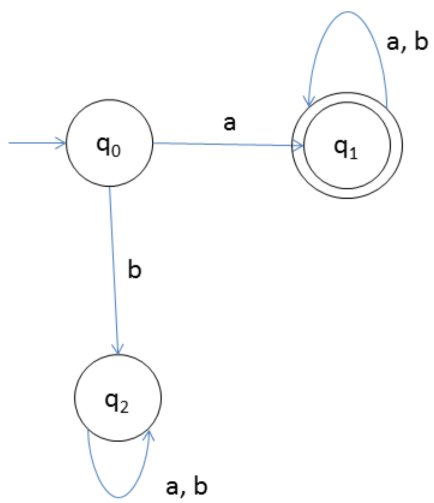
13. State Arden's theorem and construct regular expression for the following FA using Arden's theorem:

State	Input	
	0	1
A	{A, B}	ϕ
B	C	{A, B}
C	B	ϕ
A is the initial state and C is Final State		

14. Using pumping lemma, prove that the language L = $\{a^{i^2}$! i$\geq$ 1$\}$ is not regular.

15. State the pumping lemma theorem for regular languages.

16. Convert the FA given below to left linear grammar.



17. Prove that the compliment, homomorphism and inverse homomorphism, closure of a regular language is regular.

18. State and prove kleene's theorem with an example.

19. Write regular expression for a language containing strings of 0's and 1's which does not end in '01'.

20. State and prove Arden's Theorem.

21. Prove or disprove the following regarding regular expressions:
    i. $(R + S)^* = R^* + S^*$
    ii. $(RS + R)^* RS = (RR^*S)^*$