FORMAL LANGUAGES & AUTOMATA THEORY
(06CS56)

UNIT – 2

FINITE AUTOMATA & REGULAR EXPRESSIONS

* Finite automata with ε-transitions
* Regular Expressions
* Finite Automata & Regular Expressions.
* Application of Regular Expressions.

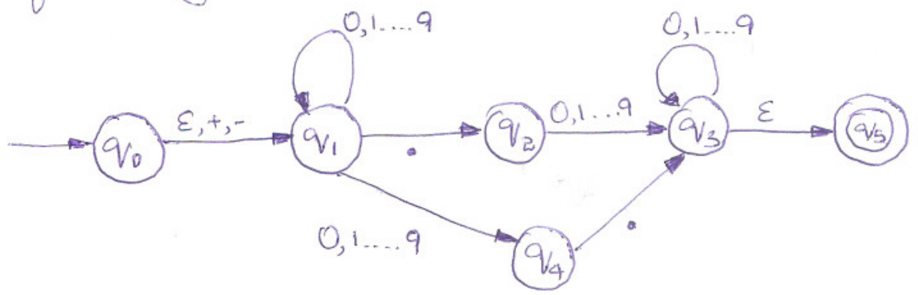
A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

ε-nfa

Introduction

When nfa is allowed to make a transition spontaneously without an input, it is called ε-nfa. The ε is really a transition on the empty string.

In order to understand ε-transitions clearly, consider the following example:



A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

Now in this figure, the ε-nfa accepts decimal numbers consisting of +,- sign or a string of digits or a decimal point or an empty string.
While the transitions from q_0 to q_1, q_1 to q_2, q_2 to q_3, q_1 to q_4, q_4 to q_3 involve some sort of operator like +,-,., 0,1...9 etc, the transition from q_3 to q_5 involves nothing. That is to say that the contribution to the transition is an empty string. In other words, state q_3 goes to q_5 automatically.

Definition of ε-nfa

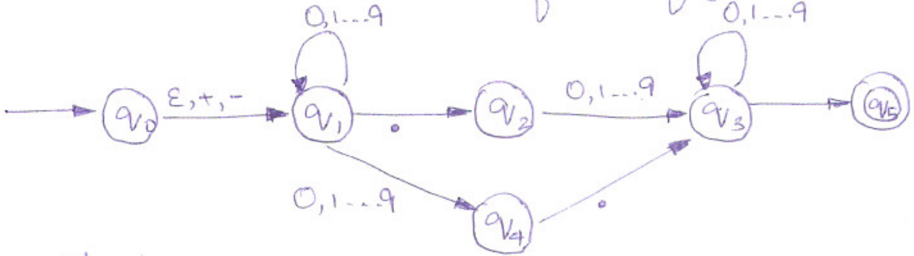
The definition of ε-nfa is almost similar to that of nfa, the only exception being the information regarding ε transitions.

ε-nfa is represented by a 5-tuple. So formally an ε-nfa A is represented by $A = (Q, \Sigma, \delta, q_0, F)$ where,

Q = finite set of states
Σ = finite set of input symbols
δ = a function that takes as arguments, (transition func.)
(i) a state in Q
(ii) a member of $\Sigma \cup \{\varepsilon\}$, meaning, any input symbol or the symbol ε (representing empty string)

q_0 = a member of Q; is the start state
F = subset of Q; is the final (or accepting) state.

For example, The ε-NFA of the figure



would be,

$$E = (\{q_0, q_1, \dots q_5\}, \{\cdot, +, -, 0, 1 \dots 9\}, \delta, q_0, \{q_5\})$$

where δ is defined by the following transition table,

	ε	$+, -$	$\cdot$	$0, 1 \dots 9$
q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1, q_4\}$
q_2	ϕ	ϕ	ϕ	$\{q_3\}$
q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
q_4	ϕ	ϕ	$\{q_3\}$	ϕ
q_5	ϕ	ϕ	ϕ	ϕ

A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

Epsilon closures

We ε-close a state 'q' by following all transitions out of q that are labelled ε. When we get to (or reach) other states by this process, we list the ε-transitions of those states. So, eventually we find every state that can be reached from q, along the paths that are labelled ε.

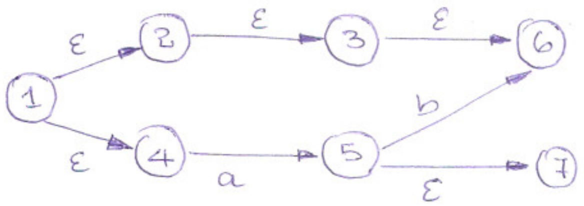
* definition of ε-closure

State q is in ECLOSE(q)

If state p is in ECLOSE(q), then there is a transition from state p to state r labelled ε, then r is in ECLOSE(q)

In other words, ε-closure of q is the set of all states which are reachable from q on ε-transitions only.

For example,



ECLOSE (1) = $\{1, 2, 3, 4, 6\}$ (Note: $q = 1$ here; (the state))

Each of these states can be reached on a path labelled ε. For example 6, $1 \rightarrow 2 \rightarrow 3 \rightarrow 6$. Here $5 \notin$ ECLOSE(1). Hence 5 is not reachable on a path labelled ε from 1, because $4 \rightarrow 5$ is not labelled ε.

Now, based on this idea of ε-closures, we define a transition function,

$$\hat{\delta}(q, w) = \varepsilon\text{-closure}(q)$$

where $w = \varepsilon$ (empty string)

A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

Conversion of ε-nfa to DFA

Given any ε-NFA, we can find a DFA 'D' that accepts the same language. This is very similar to subset construction because 'D' is a subset of the ε-NFA states. The only difference is that, we have to also consider the ε-transitions.

Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ be the ε-NFA. Then, the equivalent DFA, $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$

Step 1: $q_D = \varepsilon\text{-closure}(q_0)$, which means that the start state of DFA is obtained by taking the ε-closure of q_0.

Step 2:
Now Let S = set of all the ε-closed subsets of Q_E or $S = \text{ECLOSE}(S)$ or ε-closed sets of states of S are those such that any ε-transitions of one of the states of S belong to S. $\phi \in S$.

Now, $F_D = \{S \mid S \text{ is in } Q_D \text{ and } S \cap F_E \neq \phi\}$, that is, F_D is the set of states that contain atleast one of the (accepting) or final states of E.

Step 3: Q_D is the set of subsets of Q_E such that, $S = \text{ECLOSE}(S)$.

$\delta(S, a)$ is computed for all a in Σ and sets S in Q_D by,

(i) $S = \{p_1, p_2, \dots p_k\}$

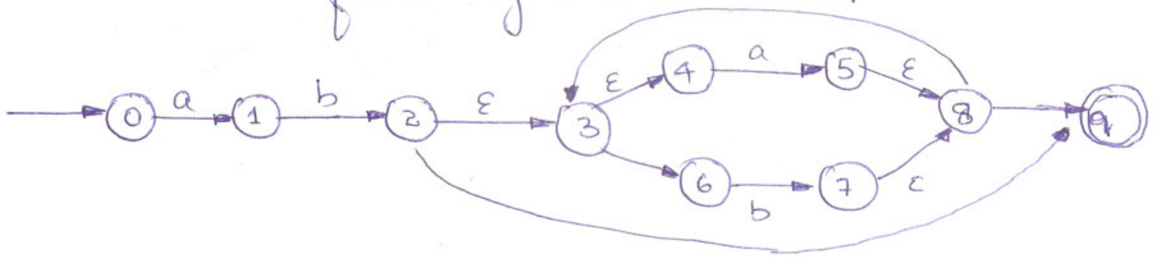
(ii) Calculate, $\delta(p_1, a) \cup \delta(p_2, a) \cup \delta(p_3, a) \dots = \bigcup_{i=1}^{k} \delta(p_i, a)$
Let this set be $\{r_1, r_2, \dots r_m\}$

(iii) Then calculate $\delta_D(S, a) = \bigcup_{j=1}^{m} \text{ECLOSURE}(r_j)$
Or $\varepsilon\text{-CLOSURE}(r_1) \cup \text{ECLOSURE}(r_2) \cup \text{ECLOSURE}(r_3) \cup \dots$

A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

PROBLEMS:

1. Convert the following NFA to its equivalent DFA.



Solution:

The start state of DFA, that is, $q_0 = \varepsilon\text{-closure}(0) = \{0\}$ — (A)

Consider the state (A),

$\delta(A,a) = \varepsilon\text{-closure}(\delta_D(A,a)) = \varepsilon\text{-closure}(\delta_D(0,a))$
$= \{1\}$ —— (B)

$\delta(A,b) = \varepsilon\text{-closure}(\delta_D(A,b)) = \varepsilon\text{-closure}(\delta_D(0,b)) = \{\phi\}$

Consider the state (B),

$\delta(B,a) = \varepsilon\text{-closure}(\delta_D(B,a)) = \varepsilon\text{-closure}(\delta_D(1,a)) = \{\phi\}$

$\delta(B,b) = \varepsilon\text{-closure}(\delta_D(B,b)) = \varepsilon\text{-closure}(\delta_D(1,b)) = \varepsilon\text{-closure}(\{2\})$
$= \{2,3,4,6,9\}$ —— (C)

because $\varepsilon\text{-closure}(2) = \{2,3,4,6,9\}$

Consider the state (C),

$\delta(C,a) = \varepsilon\text{-closure}(\delta_D(C,a)) = \varepsilon\text{-closure}(\delta_D(\{2,3,4,6,9\},a))$
$= \varepsilon\text{-closure}(\{5\}) = \{5,8,9,3,4,6\} = \{3,4,5,6,8,9\}$ — (D)

$\delta(C,b) = \varepsilon\text{-closure}(\delta_D(C,b)) = \varepsilon\text{-closure}(\delta_D(\{2,3,4,6,9\},b))$
$= \varepsilon\text{-closure}(\{7\}) = \{7,8,9,3,4,6\} = \{3,4,6,7,8,9\}$ — (E)

Consider the state (D),

$\delta(D,a) = \varepsilon\text{-closure}(\delta(D,a)) = \varepsilon\text{-closure}(\delta_D(\{3,4,5,6,8,9\},a))$
$= \varepsilon\text{-closure}(\{5\}) = \{5,8,9,3,4,6\} = \{3,4,5,6,8,9\}$ — (D)

$\delta(D,b) = \varepsilon\text{-closure}(\delta(D,b)) = \varepsilon\text{-closure}(\delta_D(\{3,4,5,6,8,9\},b))$
$= \varepsilon\text{-closure}(\{7\}) = \{7,8,9,3,4,6\} = \{3,4,6,7,8,9\}$ — (E)

Consider the state (E),

$\delta(E,a) = \varepsilon\text{-closure}(\delta(E,a)) = \varepsilon\text{-closure}(\delta_D(\{3,4,6,7,8,9\},a))$

$= \varepsilon\text{-closure}(\{5\}) = \{5,8,9,3,4,6\} = \{3,4,5,6,8,9\}$ – D

$\delta(E,b) = \varepsilon\text{-closure}(\delta(E,b)) = \varepsilon\text{-closure}(\delta_D(\{3,4,6,7,8,9\},b))$

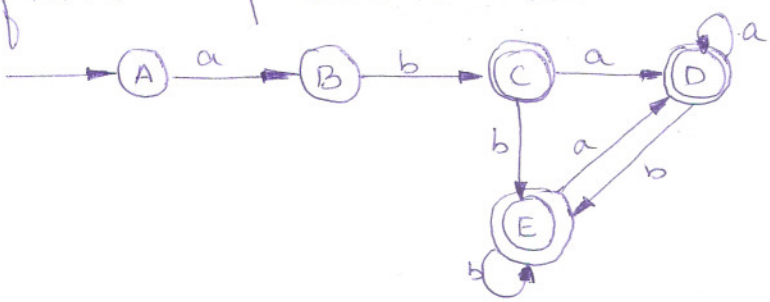
$= \varepsilon\text{-closure}(\{7\}) = \{7,8,9,3,4,6\} = \{3,4,6,7,8,9\}$ – E

At this point, there are no new states. Hence the transition table is as follows:

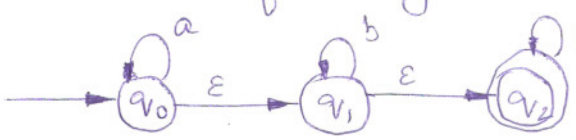
← Σ →

δ	a	b
A	B	–
B	–	C
(C)	D	E
(D)	D	E
(E)	D	E

The states C, D, E are the final states, since 9 (final state of NFA) is present in C, D, E. The transition diagram is,



2. Convert the following NFA to DFA.



Solution:

The state q_0 is the start of the ε-NFA and so the start state of DFA is the ε-closure (q_0), that is,

$$\varepsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$$

Now, we find transitions from the state $\{q_0, q_1, q_2\}$ on $\Sigma = \{a,b,c\}$

Consider the state $\{q_0, q_1, q_2\}$,

$\delta(\{q_0, q_1, q_2\}, a) = \varepsilon\text{-closure}(\{q_0\}) = \{q_0, q_1, q_2\}$ — A

$\delta(\{q_0, q_1, q_2\}, b) = \varepsilon\text{-closure}(\{q_1\}) = \{q_1, q_2\}$ — B

$\delta(\{q_0, q_1, q_2\}, c) = \varepsilon\text{-closure}(\{q_2\}) = q_2$ — C

Consider the state $\{q_1, q_2\}$

$\delta(\{q_1, q_2\}, a) = \phi$

$\delta(\{q_1, q_2\}, b) = \varepsilon\text{-closure}(\{q_1\}) = \{q_1, q_2\}$ — B

$\delta(\{q_1, q_2\}, c) = \varepsilon\text{-closure}(\{q_2\}) = q_2$ — C

Consider the state $\{q_2\}$

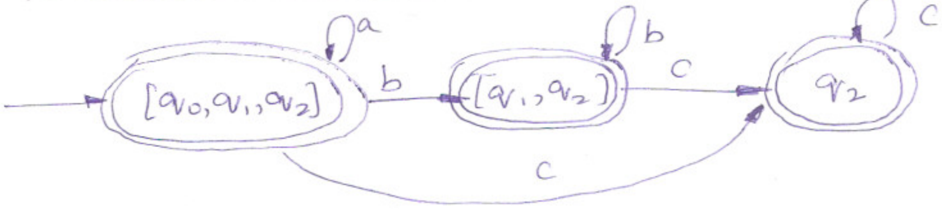
$\delta(q_0, a) = \phi$

$\delta(q_1, b) = \phi$

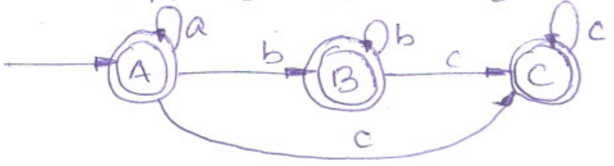
$\delta(q_2, c) = \varepsilon\text{-closure}(q_2) = q_2$ — C

The transition table along with transition diagram is,

δ	a	b	c
$\{q_0, q_1, q_2\}$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$
$[q_1, q_2]$	ϕ	$[q_1, q_2]$	$[q_2]$
$[q_2]$	ϕ	ϕ	$[q_2]$



The DFA is written as

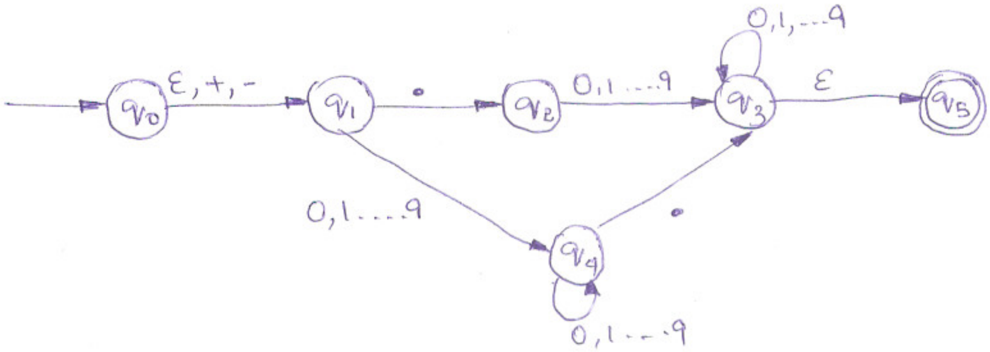


where $A = \{q_0, q_1, q_2\}$

$B = \{q_1, q_2\}$

$C = \{q_2\}$

3. Obtain an NFA with ε-transitions (ε-NFA) to accept decimal numbers and then obtain the equivalent DFA. The ε-nfa is,



Solution:

Here the start state of ε-nfa is q_0. Hence we find ε-closure(q_0) which forms the start state of DFA.

$\varepsilon\text{-closure}(q_0) = \{q_0, q_1\}$ — (A)

δ	ε	+	−	•	0,1...9
q_0	q_1	q_1	q_1	ϕ	ϕ
q_1	ϕ	ϕ	ϕ	q_2	q_4
q_2	ϕ	ϕ	ϕ	ϕ	q_3
q_3	q_5	ϕ	ϕ	ϕ	q_3
q_4	ϕ	ϕ	ϕ	q_3	q_4
q_5	ϕ	ϕ	ϕ	ϕ	ϕ

Consider state (A),

$\delta(A, \pm) = \varepsilon\text{-closure}(\delta_D(A, \pm)) = \varepsilon\text{-closure}(\delta_D(\{q_0, q_1\}, \pm))$

$= \varepsilon\text{-closure}(q_1) = \{q_1\}$ —— (B)

$\delta(A, \bullet) = \varepsilon\text{-closure}(\delta_D(A, \bullet)) = \varepsilon\text{-closure}(\delta_D(\{q_0, q_1\}, \bullet))$

$= \varepsilon\text{-closure}(q_2) = \{q_2\}$ —— (C)

$\delta(A, \{0,1..9\}) = \varepsilon\text{-closure}(\delta_D(A, \{0,1...9\})) = \varepsilon\text{-closure}(\delta_D(\delta_D(\{q_0,q_1\}, \{0,1..9\}))$

$= \varepsilon\text{-closure}(q_4) = \{q_4\}$ —— (D)

$\delta(B, \pm) = \varepsilon\text{-closure}(\delta_D(B, \pm)) = \varepsilon\text{-closure}(\delta_D(q_1, \pm))$

$= \varepsilon\text{-closure}(\phi) = \phi$

$\delta(B, \bullet) = \varepsilon\text{-closure}(\delta_D(B, \bullet)) = \varepsilon\text{-closure}(\delta_D(q_1, \bullet))$

$= \varepsilon\text{-closure}(q_2) = \{q_2\}$ —— (C)

$\delta(B,\{0,1..9\}) = \varepsilon\text{-closure}(\delta_D(B,\{0,1...9\})) = \varepsilon\text{-closure}(\delta_D(q_1,\{0,1..9\}))$
$= \varepsilon\text{-closure}(q_4) = \{q_4\}$ — D

$\delta(C,\pm) = \varepsilon\text{-closure}(\delta_D(C,\pm)) = \varepsilon\text{-closure}(\delta_D(q_2,\pm))$
$= \varepsilon\text{-closure}(\phi) = \phi$

$\delta(C,\cdot) = \varepsilon\text{-closure}(\delta_D(C,\cdot)) = \varepsilon\text{-closure}(\delta_D(q_2,\cdot)) = \varepsilon\text{-closure}(\phi) = \phi$

$\delta(C,\{0,1...9\}) = \varepsilon\text{-closure}(\delta_D(C,\{0,1...9\})) = \varepsilon\text{-closure}(\delta_D(q_2,\{0,1..9\}))$
$= \varepsilon\text{-closure}(q_3) = \{q_3,q_5\}$ — E

$\delta(D,\pm) = \varepsilon\text{-closure}(\delta_D(D,\pm)) = \varepsilon\text{-closure}(\delta_D(q_4,\pm)) = \varepsilon\text{-closure}(\phi) = \phi$

$\delta(D,\cdot) = \varepsilon\text{-closure}(\delta_D(D,\cdot)) = \varepsilon\text{-closure}(\delta_D(q_4,\cdot)) = \varepsilon\text{-closure}(q_3) = \{q_3,q_5\}$ — E

$\delta(D,\{0,1...9\}) = \varepsilon\text{-closure}(\delta_D(D,\{0,1...9\})) = \varepsilon\text{-closure}(q_4) = \{q_4\}$ — D

$\delta(E,\pm) = \varepsilon\text{-closure}(\delta_D(E,\pm)) = \varepsilon\text{-closure}(\phi) = \phi$

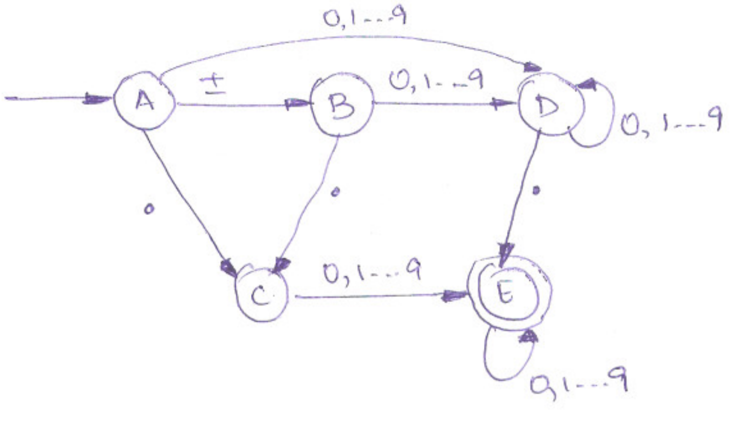
$\delta(E,\cdot) = \varepsilon\text{-closure}(\delta_N(E,\cdot)) = \varepsilon\text{-closure}(\phi) = \phi$

$\delta(E,\{0,1...9\}) = \varepsilon\text{-closure}(\delta_D(E,\{0,1...9\})) = \varepsilon\text{-closure}(q_3) = \{q_3,q_5\}$ — E

Σ

Q

δ	+	−	.	0,1...9
A	B	B	C	D
B	ϕ	ϕ	C	D
C	ϕ	ϕ	ϕ	E
D	ϕ	ϕ	E	D
* E	ϕ	ϕ	ϕ	E

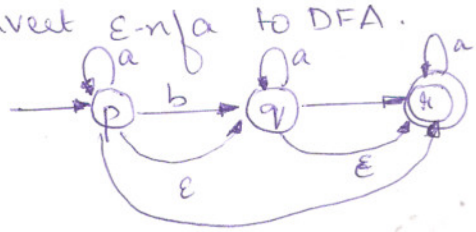


4. Consider the following

	ε	a	b	c
→ p	∅	{p}	{q}	{r}
q	{p}	{q}	{r}	∅
* r	{q}	{r}	∅	{p}

A. M. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078

Convert ε-nfa to DFA.



Solution: ε-closure (p) = {p} — A

ε-closure (δ(A,a)) = ε-closure (δ(p)) = {p} → A

ε-closure (δ(A,b)) = ε-closure (p,q) = {p,q} → B

ε-closure (δ(A,c)) = ε-closure (p,q,r) = {p,q,r} → C

ε-closure (δ(B,a) = ε-closure (p,q) = {p,q} — B

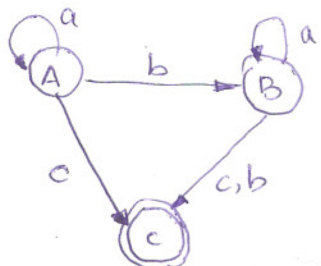
ε-closure (δ(B,b)) = ε-closure (q,r) = {q,p,r} — C

ε-closure (δ(B,c)) = ε-closure (r) = {p,q,r} — C

ε-closure (δ(C,a)) = ε-closure {p,q,r} = {p,q,r} — C

ε-closure (δ(C,b)) = ε-closure {p,q,r} = {p,q,r} — C

ε-closure (δ(C,c)) = ε-closure {r,p} = {p,q,r} — C

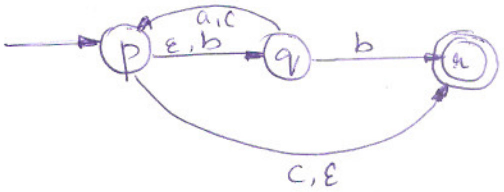


δ	a	b	c
A	A	B	C
B	B	C	C
C	C	C	C

5. Consider the following ε-nfa.

	ε	a	b	c
p	{q, r}	∅	{q}	{r}
q	∅	{p}	{r}	{p, q}
* r	∅	∅	∅	∅

A. N. PRASAD
Assistant Professor
Department of Computer Science & Engineering
Dayananda Sagar College of Engineering
Bangalore - 560 078



Solution: ε-closure(p) = {p, q, r} — A = ε-closure(δ,(A,a))

ε-closure(δ(A,b)) = ε(q, r) = {q, r} — B

ε-closure(δ(A,c)) = ε(r, q) = {p, q, r} — A

ε-closure(δ(B,a)) = ε(p) = {p, q, r} — A

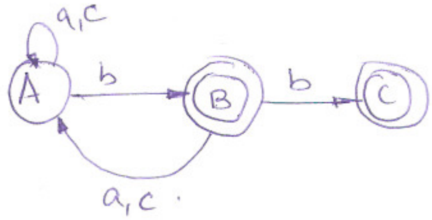
ε-closure(δ(B,b)) = ε(r) = {r} — C

ε-closure(δ(B,c)) = ε(q,p) = {p, q, r} — A

ε-closure(δ(C,a)) = ε(∅) = ∅

ε-closure(δ(C,b)) = ε(∅) = ∅

ε-closure(δ(C,c)) = ε(∅) = ∅



Regular Expressions

The Language accepted by the Automata is called regular language. A regular language can be defined using regular expressions consisting of symbols of alphabet Σ, the operators $\cdot$, $+$, and $*$.

Definition: A regular expression can be defined as

i) ϕ is a Regular Expression denoting Empty language

ii) ε - is a Regular Expression indicating the language containing empty string.

iii) If a and b are i/p symbols, $\Sigma = \{a, b\}$ then

a) 'a' is regular expressions

b) $a+b$ is regular expressions equivalent to $L_a \cup L_b$

c) $a \cdot b$ is regular expression equivalent to $L_a \cdot L_b$

d) a^* is a regular expression corresponding to language L_a

Problems:

1. Write regular expression for strings of a's and b's having length 2

Ans: Strings of a's and b's with length two. ~~are~~

$(aa + db + ba + bb)$ is Regular Expression

or $(a+b).(a+b)$

2. obtain regular expression for strings starting with a and ending with b.

Ans: a ..... b

string of a's & b's

Regular expression is $a\,(a+b)^*\,b$

3. Obtain regular expression for strings starting and ending with same letter over Σ a, b

Ans a .... a or b ..... b

string of a's & b's

$$a\,(a+b)^*\,a + b\,(a+b)^*\,b$$

4. Obtain a regular expression ends with ab or ba over $\Sigma = \{a, b\}$.

Ans $\ldots\ldots (ab+ba)$

↑ string of a's & b's

Regular Expression $(a+b)^* ab+ba$

5. Give regular expression for string containing exactly one a over Σ a, b, c.

Ans: The possible strings are abc, acb, bac, bca, cab, cba

$(b+c)^* a (b+c)^*$ is Regular Expression.

6. obtain regular expression donot end with 01, Σ 0,1

Ans: Not end with 01, $\underbrace{00, 10, 11}$, ↑ can end with

$(a+b)^* (00+10+11)$

7. Obtain regular expression for $L = \{w : n_a(w) \bmod 3 = 0, w \in (a,b)^*\}$

Ans: The minimum string is aaa. The other o/p b can be $b^* a b^* a b^* a b^*$

8. Obtain regular expression for

$L = \{ w : |w| \bmod 3 = 0, \text{ where } w \in (a, b)^* \}$

Ans: Possible strings are

$L = \{ \epsilon, aaa, aba, aab, abb, baa, bab, bbb, \ldots\ldots \}$

The Regular expression is

$$(a+b)(a+b)(a+b)$$

For any length & ϵ

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

9. Obtain a regular expression for language

$L = \{ a^{2n} b^{2m}, n \geq 0, m \geq 0 \}$

Ans: a^{2n}, $n \geq 0$ results in even number of a's

b^{2m} $m \geq 0$ results in even number of b's

Regular Expression = $(aa)^* (bb)^*$

10. Obtain regular expression containing not more than three a's. over Σ a, b

Ans: String can have 0 a, 1 a, 2 a, 3 a. The RE is $(\epsilon + a)(\epsilon + a)(\epsilon + a)$. Now Insert b so $b^*(\epsilon+a)b^*(\epsilon+a)b^*(\epsilon+a)b^*$

11. Obtain regular expression to accept strings of a's and b's of length ≤ 2.

Ans: The string can be $\varepsilon + a + b + aa + ab + ba + bb$

The Regular expression is $(\varepsilon + a + b)(\varepsilon + a + b)$

$$= (\varepsilon + a + b)^2$$

12. obtain regular expression to accept string of a's & b's ≤ 10

Ans: Same as previous problem but length $= 10$

$$= (\varepsilon + a + b)^{10}$$

13. Give regular expression for string of a & b of even length.

Ans: The minimum string of even length 2 or zero is

$(aa + ab + ba + bb)^*$ or

it can be $((a+b)(a+b))^*$

14. Give regular expression for string of a's & b's of odd length.

Ans The minimum length for odd length is a or b

$(a+b)$

For any length $(a+b)((a+b)(a+b))^*$

15. Regular expression for alternative a's & b's

Ans. Minimum String ab

For any length $(ab)^* = \{ab, abab, \ldots\ldots\}$

For bab or aba

Regular Expression is $(\varepsilon+b)(ab)^*(\varepsilon+a)$

16. Regular expression with at least one 03 consecutive 0's.

Ans: The minimum string is 000

000

string of 0's & 1's | string of 0's & 1's

$(0+1)^* 000 (0+1)^*$

17.

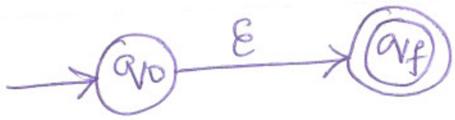
ε-NFA from Regular Expression

Let R be regular expression. Then there exists a finite Automata $M = (Q, \Sigma, \delta, q_0, F)$ which accepts $L(R)$.

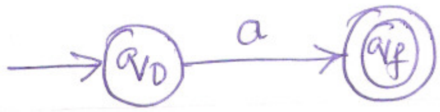
Proof: ϕ, ε, a are regular expressions. The Automata to recognize these expressions are

(i) ϕ

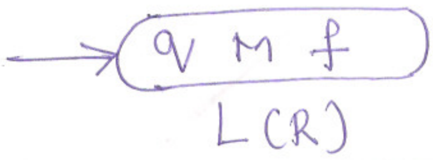
(ii) ε



(iii) a



The Schematic representation of Finite automata to accept language $L(R)$ is (R is regular expression)

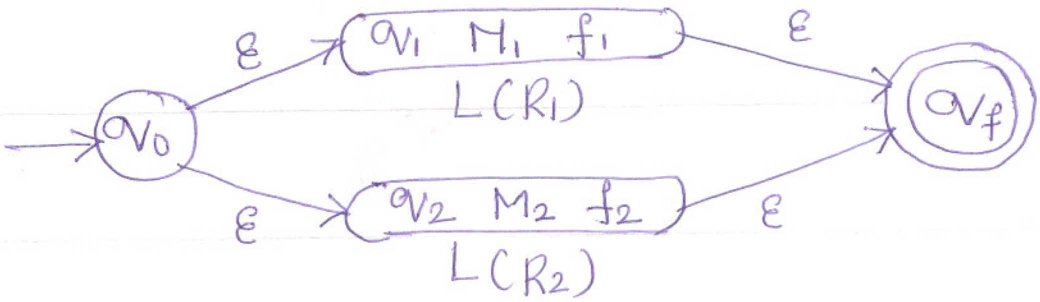


Let $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, f_1)$ accepts $L(R_1)$

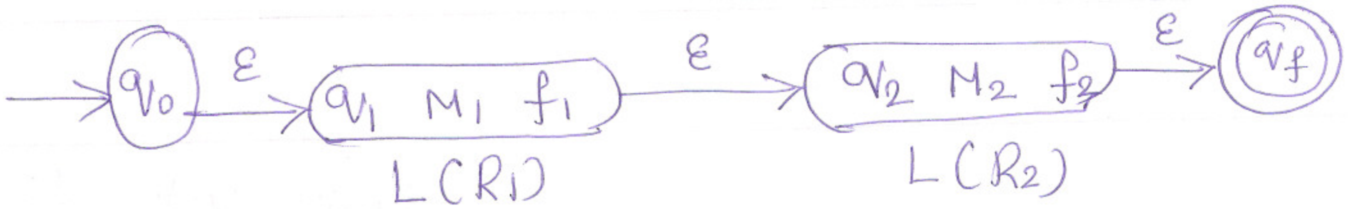
$M_2 = (Q_2, \Sigma_2, \delta_2, q_2, f_2)$ accepts $L(R_2)$.

Then machine corresponding to regular expression R_1+R_2, $R_1 \cdot R_2$, R^* are

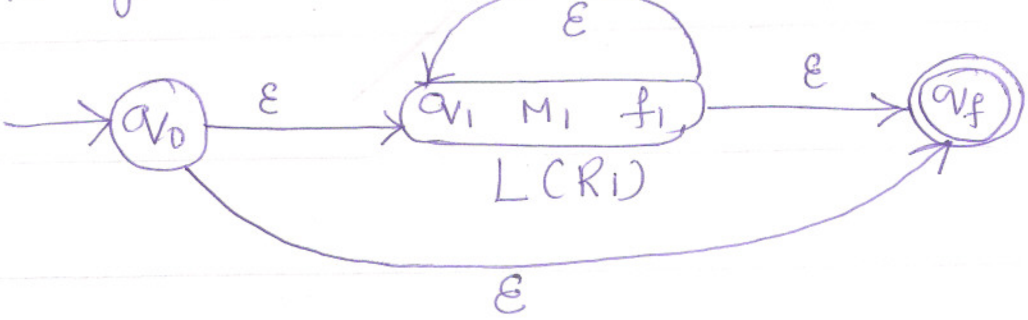
⟨i⟩ R_1+R_2, NFA to accept $L(R_1)$ or $L(R_2)$ is shown below



⟨ii⟩ $R_1 \cdot R_2$, NFA to accept $L(R_1)$ followed by $L(R_2)$ is shown below



⟨iii⟩ $R_1^* = \{\varepsilon, R_1, R_1R_1, \ldots\ldots\}$. NFA to accept $L(R_1)^*$ is given below

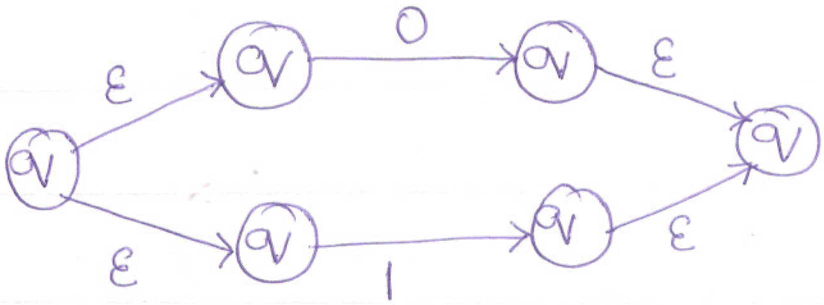


Problems. 1. Write ε-NFA for Regular expression $(0+1)^*+1$

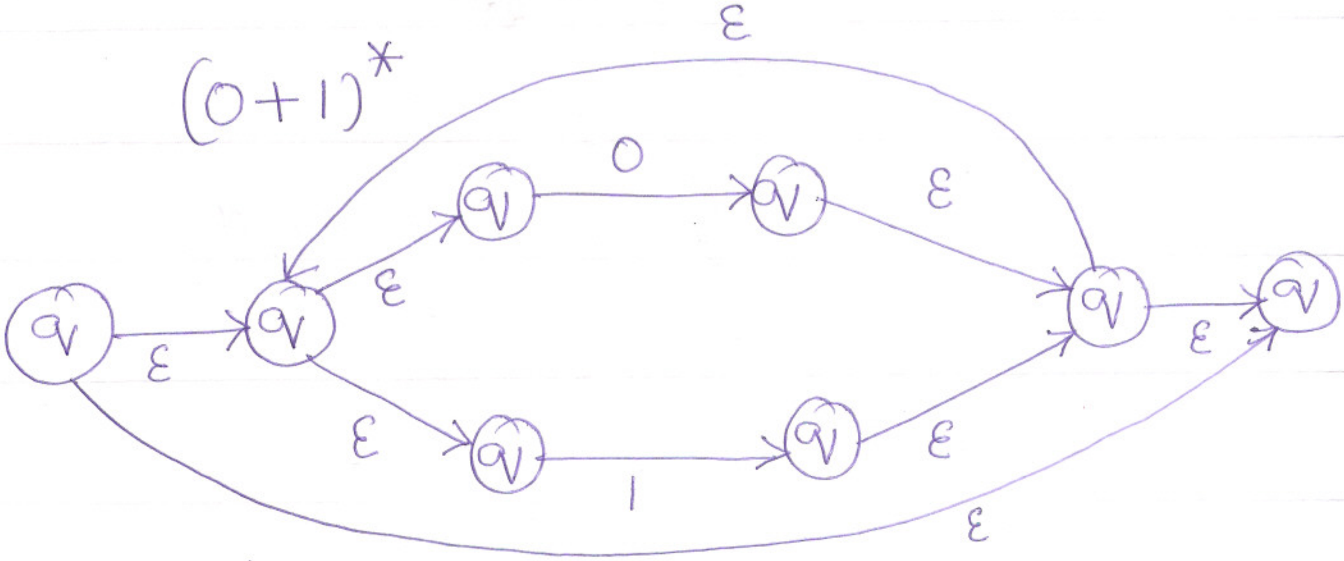
Solution: $(0+1)^* + 1$

step 1: First write ε-NFA for $(0+1)^*$

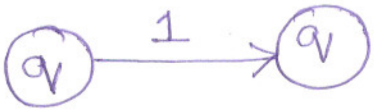
$(0+1)$



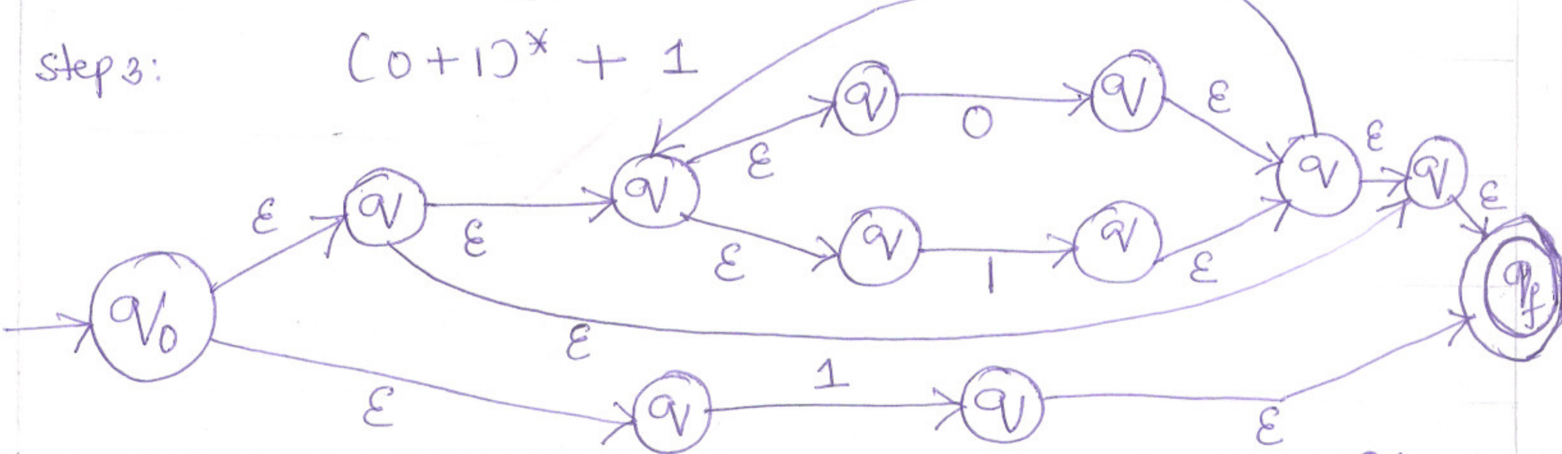
$(0+1)^*$



step 2: Write Automata for 1

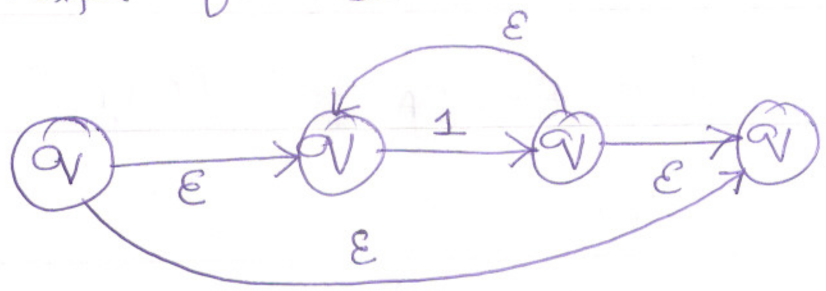


step 3: $(0+1)^* + 1$

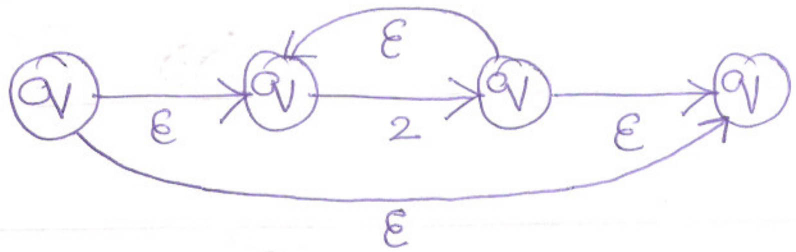


2. Obtain ε-nfa for $1^* + 2^* + 3^*$

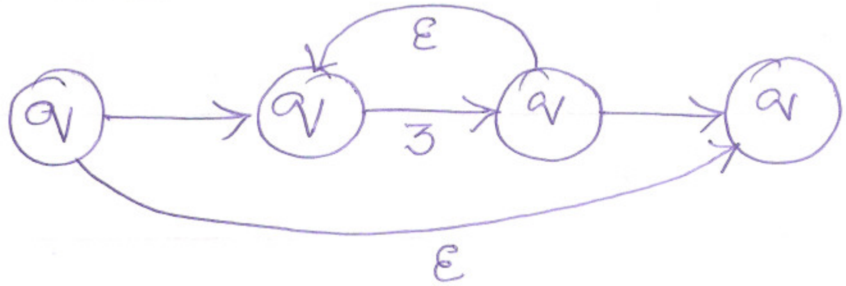
Ans: ε-nfa for 1^*



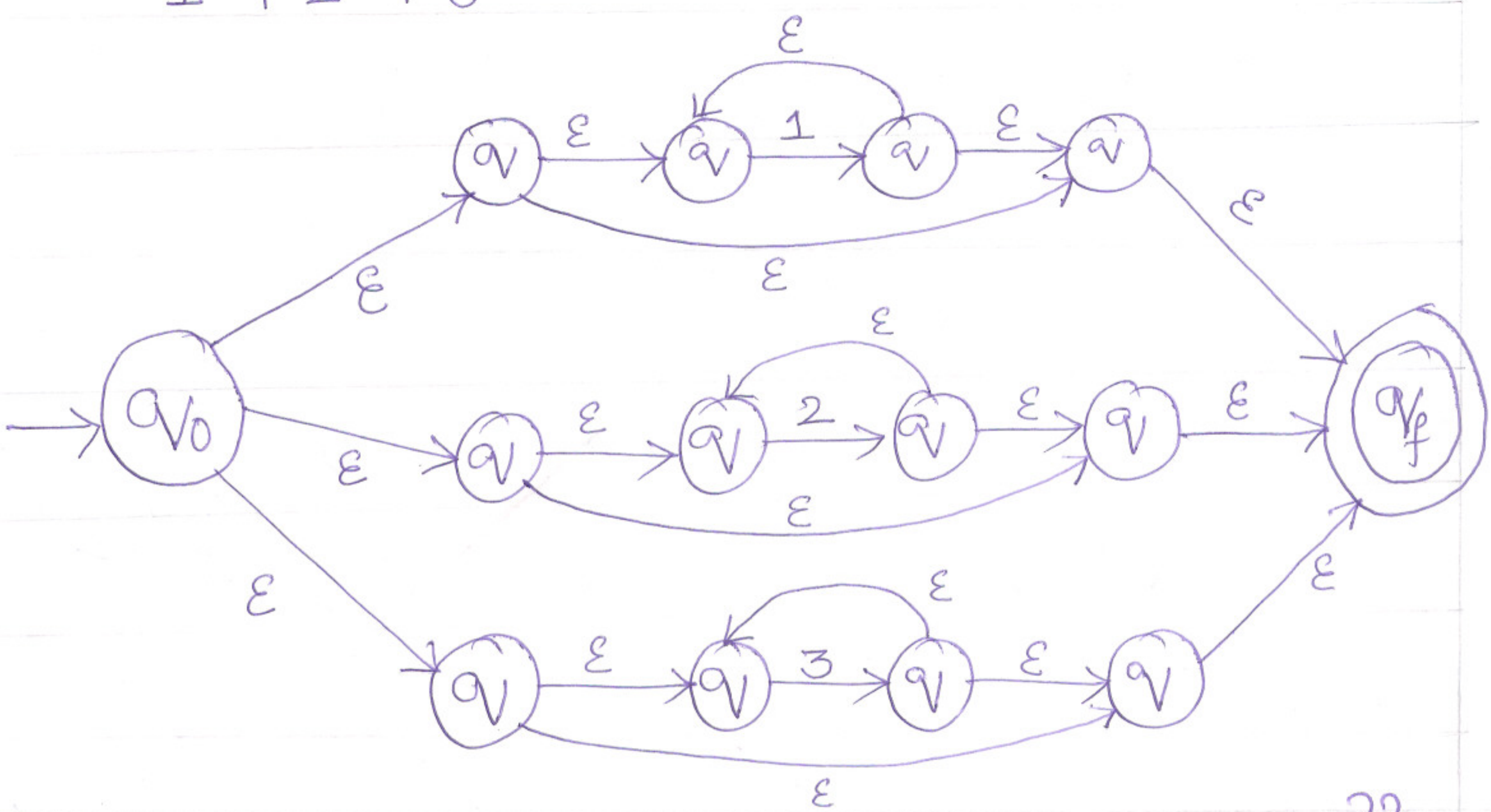
ε-nfa for 2^*



ε-nfa for 3^*

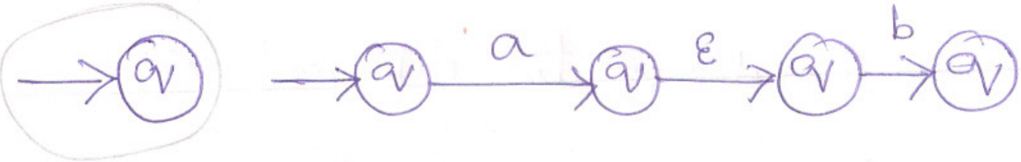


$1^* + 2^* + 3^*$



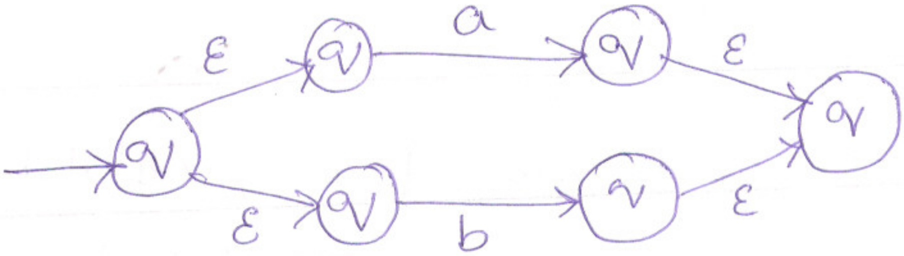
3. Obtain ε-nfa for $ab(a+b)^*$

Ans: ε-nfa for ab is

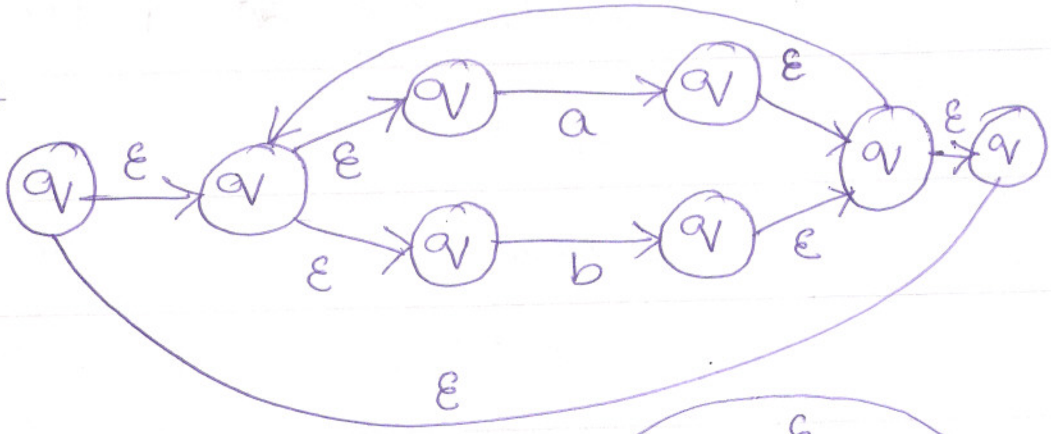


ε-nfa for $(a+b)^*$

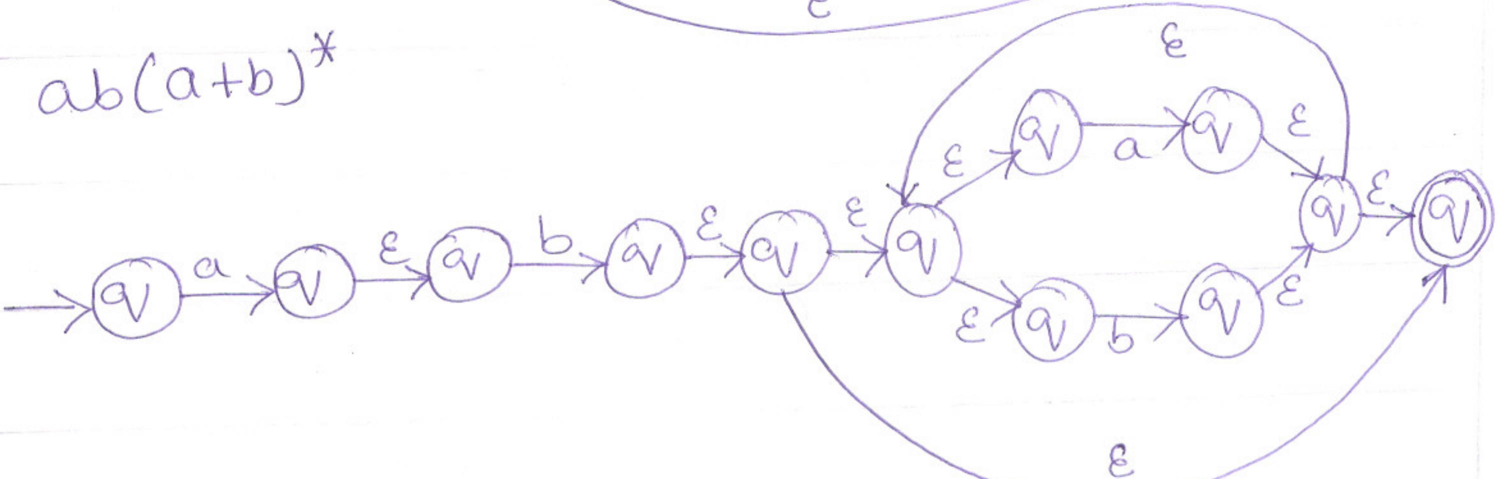
$a+b$



$(a+b)^*$



$ab(a+b)^*$



REGULAR EXPRESSION FROM AUTOMATA

The Regular expression can be obtained from finite automata in two different ways

i) Kleene's Theorem

ii) State Elimination method.

Kleene's Theorem

If the language L is accepted by a DFA then there is a regular expression R for regular language L such that $L = L(R)$

Proof: Let $Q = \{q_0, q_1 \text{-----} q_n\}$ are states of m/c M.
The path from i to j through intermediate state whose number is not greater than k is given by regular expression

$$R_{ij}^{k} = \{ w \in \Sigma^*, \text{ w is string from i to j}$$

1 $K=0$, indicates no intermediate state from i to j

This is possible when $i \neq j$. With all o/p symbols 'a' in DFA there is a transition from i to j with following cases.

a: No i/p symbol, corresponding Regular expression is

$$R_{ij}^{0} = \phi$$

b: Exactly one i/p symbol 'a' on which there is a transition from state i to j, the regular expression is

$$R_{ij}^{0} = a$$

c: There are multiple i/ps $a_1, a_2, \dots a_K$ where there is a transition from each symbol from state i to state j, the regular expression is

$$R_{ij}^{0} = a_1 + a_2 + a_3 + \dots + a_K$$

2. There is only one state such that $i=j$, there is a path from i to itself forming self loop or path of length 0, denoted by ϵ

$$R_{ij} = \phi + \epsilon$$

a. No i/p symbol, Regular expression is

$$R_{ij}^{0} = \phi + \varepsilon$$

b. Exactly one i/p symbol a

$$R_{ij}^{0} = a + \varepsilon$$

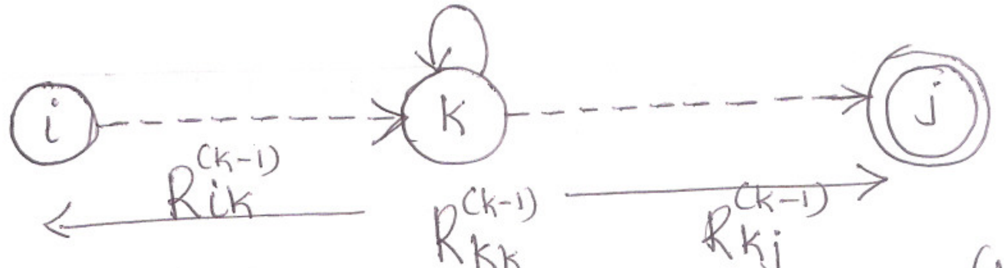
c. Multiple i/ps $a_1, a_2, a_3, \ldots a_k$

$$R_{ij}^{0} = a_1 + a_2 + a_3 + \ldots + \varepsilon$$

3. There exists a path from i to j through a state which is not greater than k. The two cases are

i) a path from i to j does not go through k. $R_{ij}^{(k-1)}$

ii) a path from i to j go through k shown below

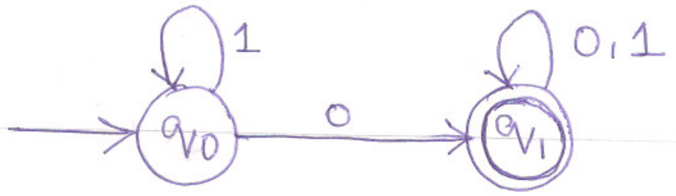


1. path from i to k is $R_{ik}^{(k-1)}$
2. path from k to k is $\left(R_{kk}^{(k-1)}\right)^*$
3. path from k to j is $R_{kj}^{(k-1)}$

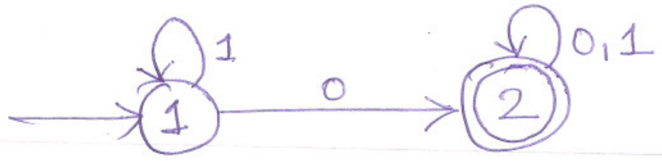
The path from i to j is concatenation of 3 regular expressions

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

Problems. 1. Obtain Regular Expression for automata shown below.



Solution: Let $q_0 = 1$, $q_1 = 2$, so automata is



When $k = 0$

$$R_{11}^{(0)} = \varepsilon + 1$$

$$R_{12}^{(0)} = 0$$

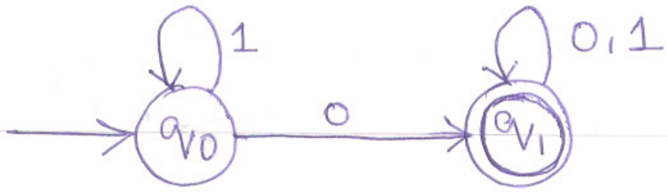
$$R_{21}^{0} = \phi$$

$$R_{22}^{0} = \varepsilon + 0 + 1$$

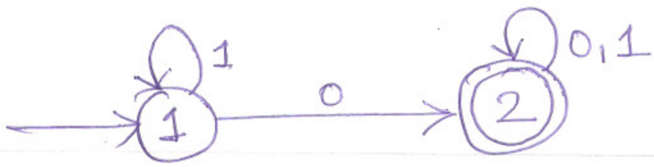
The path from i to j is concatenation of 3 regular expressions

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

Problems. 1. Obtain Regular Expression for automata shown below.



Solution: Let $q_0 = 1$, $q_1 = 2$, so automata is



When $k = 0$

$$R_{11}^{(0)} = \varepsilon + 1$$

$$R_{12}^{(0)} = 0$$

$$R_{21}^{0} = \phi$$

$$R_{22}^{0} = \varepsilon + 0 + 1$$

When $k=1$

Regular expression is

$$R_{ij}^{k} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left[R_{kk}^{k-1} \right]^{*} R_{kj}^{(k-1)}$$

$$R_{11}^{1} = R_{11}^{0} + R_{11}^{0} \left[R_{11}^{0} \right]^{*} R_{11}^{0}$$

$$= (\varepsilon+1) + (\varepsilon+1)(\varepsilon+1)^{*}(\varepsilon+1)$$

$$= (\varepsilon+1) + (\varepsilon+1)\ 1^{*}\ (\varepsilon+1)$$

$$= 1^{*}$$

$$R_{12}^{1} = R_{12}^{0} + R_{11}^{0} \left[R_{11}^{0} \right]^{*} R_{12}^{0}$$

$$= 0 + (\varepsilon+1)(\varepsilon+1)^{*} 0$$

$$= 0 + 1^{*} 0$$

$$= 1^{*} 0$$

$$R_{21}^{1} = R_{21}^{0} + R_{21}^{0} \left[R_{11}^{0} \right]^{*} R_{11}^{0}$$

$$= \phi + \phi\ (\varepsilon+1)^{*}(\varepsilon+1)$$

$$= \phi$$

$$R_{22}^{1} = R_{22}^{0} + R_{21}^{0} \left[R_{11}^{0} \right]^{*} R_{12}^{0}$$

$$R_{22}^{1} = R_{22}^{1}(\varepsilon+0+1)+\phi(\varepsilon+1)^{*}0$$

$$= (\varepsilon+0+1)$$

When $k=2$

$$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}[R_{22}^{1}]^{*}R_{21}^{1}$$

$$= 1^{*} + 1^{*}0(\varepsilon+0+1)^{*}\phi$$

$$= 1^{*}$$

$$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}[R_{22}^{1}]^{*}R_{22}^{1}$$

$$= 1^{*}0 + 1^{*}0(\varepsilon+0+1)^{*}(\varepsilon+0+1)$$

$$= 1^{*}0 + 1^{*}0(0+1)^{*}(\varepsilon+0+1)$$

$$= 1^{*}0 + 1^{*}0(0+1)^{*}$$

$$= 1^{*}0(0+1)^{*}$$

$$R_{22}^{2} = R_{22}^{1} + R_{22}^{1}[R_{22}^{1}]^{*}R_{22}^{1}$$

$$= (\varepsilon+0+1)+(\varepsilon+0+1)(\varepsilon+0+1)^{*}(\varepsilon+0+1)$$

$$= (\varepsilon+0+1)+(0+1)^{*}$$

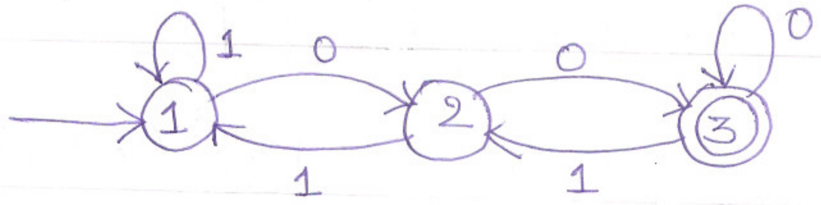
$$= (0+1)^{*}$$

Since total number of states are 02, the maximum value of k is 02. Since start state is 1 and end state is 2, the regular expression is given by

$$R_{12}^{2} = 1^{*}0\,(0+1)^{*}$$

is accepted by given DFA

2. Consider the DFA



Ans: When $K=0$

$R_{11}^{0} = \varepsilon + 1$ $\quad R_{21}^{0} = 1$ $\quad R_{31}^{0} = \emptyset$

$R_{12}^{0} = 0$ $\quad R_{22}^{0} = \emptyset + \varepsilon = \varepsilon$ $\quad R_{32}^{0} = 1$

$R_{13}^{0} = \emptyset$ $\quad R_{23}^{0} = 0$ $\quad R_{33}^{0} = \varepsilon + 0$

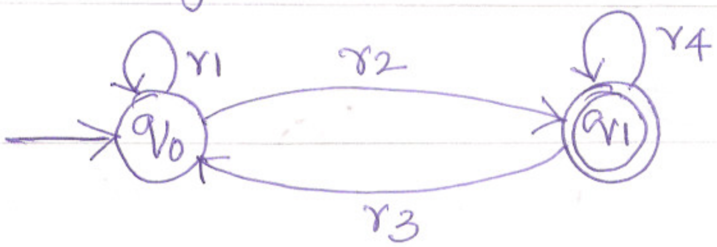
Similarly $K=1$, $K=2$, $K=3$ (to solve)

R_{13}^{3} is the regular expression accepted

Regular Expression from Finite Automata

State elimination method

The general procedure to obtain a regular expression is given below



The regular expression for this is

$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$$

if r_3 is not there

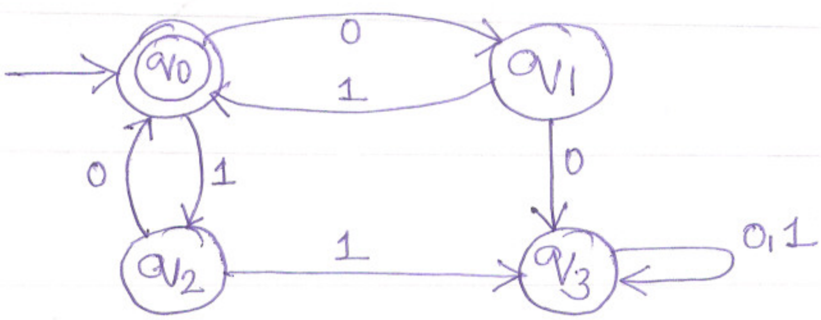
$$r = r_1^* r_2 r_4^*$$

q_0 and q_1 are final states then

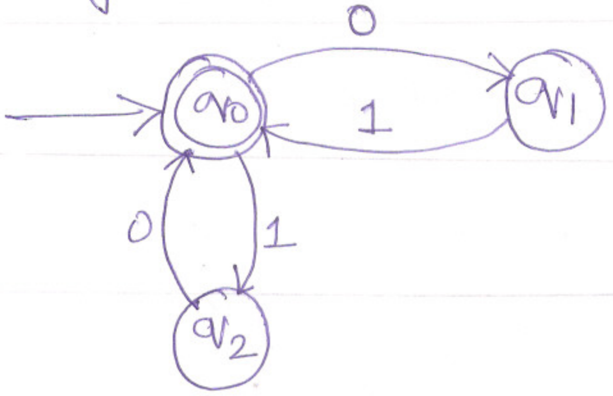
$$r = r_1^* + r_1^* r_2 r_4^*$$

Problem.

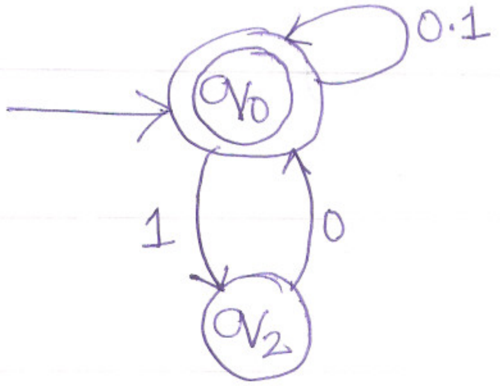
1. obtain regular expression for FA given



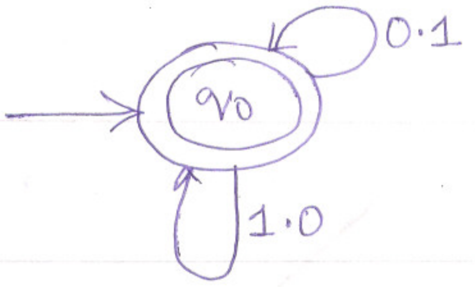
Solution: In the diagram q_3 is dead state, can be directly removed



The transition from q_0 to q_1 and q_1 to q_0 gives the string 01 can be reduced to



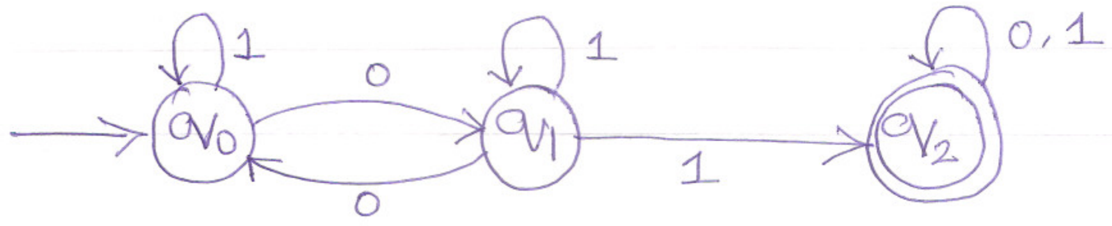
The transition from q_0 to q_2 and q_2 to q_0 gives the string 10. can be reduced to



The Regular expression, the machine accepts is

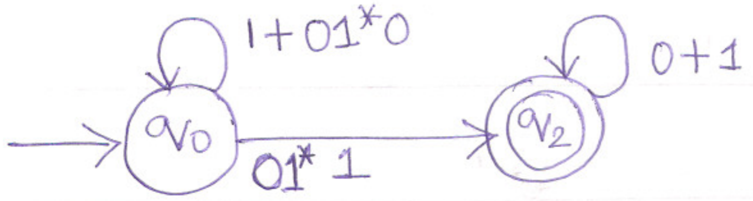
$$(01+10)^*$$

2. Obtain Regular expression for Automata



Solution: The graph should be converted to generalized form by eliminating state.

Transition from q_0 to q_1 01^*0. DFA is



$r_1 = 1 + 01^*0$

$r_2 = 01^*1$ $\qquad r = r_1^* r_2 r_4^*$

$r_3 = \phi$

$r_4 = 0+1$

$$r = (1+01^*0)^* 01^*1 (0+1)^*$$

is accepted by Finite Automata