MODULE 2

FINITE AUTOMATA

Finite state automaton is an abstract model of a computer. It is represented in the figure. The components of the automaton are: Input Tape, Finite Control and Tape Head.

Input: String

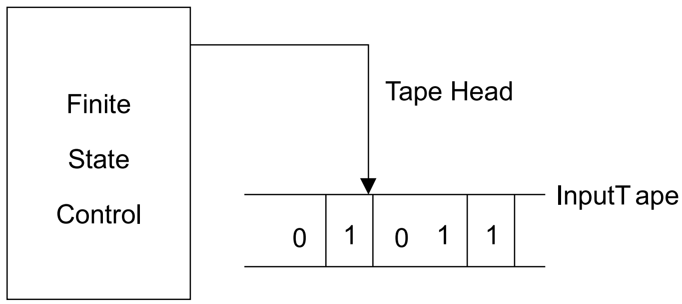


Fig. 1.1 The Working Model of a Finite Automata

Operation

String Processing (scans the string from left to right, one symbol at a time and moves from state to state) using its transition function.

Output: Yes/No (Accepted/Rejected)

1.5.1 Mathematical Representation

A Finite Automaton(FA) is represented by a 5-tuple machine.

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

* Q is a finite non-empty set of states
* Σ is a finite non-empty set of symbols
* (tebahpla na)
* $\delta : Q X \Sigma \rightarrow Q$ is the transition function
* $q_0 \in Q$ is the start state

* F ϵ Q is a set of final states

1. Transition function

It is a function which guides the automata in string processing. It takes two inputs (a state, a symbol) and gives one output (state). Transition function can be represented in three ways. They are,

i. Diagrammatic representation

Nodes and edges are used. Nodes represent the states and edges represent the moves. The labels of the edges represent the processing symbols. There are two types of nodes: a) single circled node indicating non-final (non-accepting) state; b) double circled node indicating final state.

ii. Tabular representation

It consists of Rows and columns. Rows indicate state and columns indicate symbol. The entries of the table indicate the output state. The arrow and star symbols are used to point out the starting and final states respectively.

iii. Functional representation

The name of the function is δ. The input parameters are q,a.

* Where q is a state and a is a symbol. The function returns a state p.

Example:

The automata of the language L={*w* | *w* contains ab}

2. Diagrammatic Representation - Transition diagram

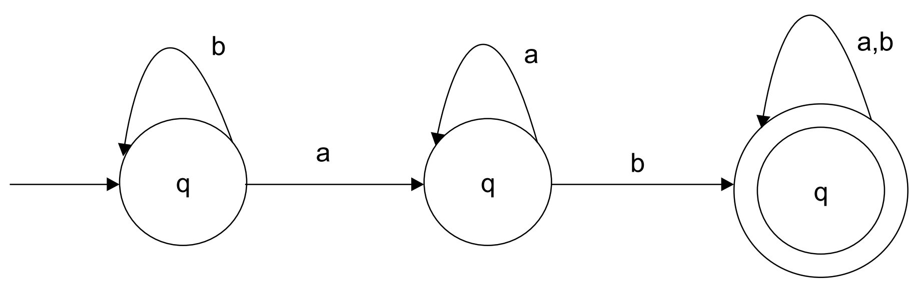


Fig. 1.2 The transition diagram of FA for the language L={w | w contains ab}

δ :

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a,b\}$ q_0

$= q_0$ $F = q_2$

Table 1.2 The transition table of FA for the language L={w | w contains ab}

Δ	**a**	**b**
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
* q	q	q

3. Functional Representation - Transition functions

$\delta(q_0, a) = q_1$	$\delta(q_0, b) = q_0$
$\delta(q_1, a) = q_1$	$\delta(q_1, b) = q_2$
$\delta(q_2, a) = q_2$	$\delta(q_2, b) = q_2$

1.5.2 Types of Finite Automata

1. Deterministic

- à If there is exactly one output state in every transition function of an automata, then the automata is called Deterministic finite Automata (DFA)

- à A Deterministic finite automaton (DFA) is represented by a 5-tuple machine i.e. $M = (Q, \Sigma, \delta, q_0, F)$

  - * Q is a finite non-empty set of states
  - * Σ is a finite non-empty set of symbols
  - * (tebahpla na)
  - * $\delta : Q X \Sigma \rightarrow Q$ is the transition function

* $q_0 \in Q$ is the start state
* $F \in Q$ is a set of final states

2. Non-Deterministic

à If there is zero or more output states in any of the transition functions of an automata then that automata is called Non-Deterministic Finite Automata (NFA).

à NFA is the preliminary form of a machine, which can be easily constructed using the basic constraints of a language.

à Then it can be converted into DFA using subset construction method and finally minimization methods are used to reduce the size of the machine.

à A Non-Deterministic finite automaton (NFA) is represented by 5-tuples.
i.e. M = (Q, Σ, δ, q0, F)

* Q is a finite non-empty set of states
* Σ is a finite non-empty set of symbols tebahpla na ( )
* δ: QX Σ → 2^Q (subset of Q) is the transition function
* $q_0 \in Q$ is the start state
* $F \in Q$ is a set of final states

3. ε-NFA

à If there is a transition for ε symbol in NFA , then the automata is called ε-NFA. An ε-Non-Deterministic finite automaton (NFA) is represented by 5-tuples. i.e. M = (Q, Σ, δ, q_0, F)

* Q is a finite non-empty set of states
* Σ is a finite non-empty set of symbols tebahpla na ( )
* δ : QX (Σ { ε })→ 2^Q (subset of Q) is the transition function
* $q_0 \in Q$ is the start state
* $F \in Q$ is a set of final states

1.5.3 Language of an Automata

1. **L(M)** à The language of machine M à Set of all strings machine M accepts

2. **L(DFA)**

   $\{w \mid \hat{\delta}(q_0, w) = p \in F\}$

   Where,

   $\hat{\delta}(q_0, w)$ is an extended transition function that takes a state q_0 and a string w and returns a state p which is in F = Regular language.

3. **L(NFA)**

   $\{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$ - Regular language.

1.6 determInIstIc FInIte AutomAtA(dFA)

Deterministic finite Automata is a definite model of computation where there is single output for every symbol from every state. The transition table of a DFA will be complete and unambiguous. There would not be any empty entry and multiple entries.

1.6.1 String Processing

à An automata processes the given string and gives Yes/No as the output.

à During string processing, the symbols in the given string are processed one by one, from left to right according to the moves defined by the transition functions of the automata.

à A set of transition function defines an automata.

à During string processing, automata selects the transition function whose input matches with the current state (state and symbol) and performs a move to output state.

1.6.2 String Processing in DFA

Problem	**1.1**

Let M = (Q, Σ, δ, q_0, F)
where Q={ q_0 , q_1 ,q_2 }, Σ,={a,b}
F={q_2}

δ	**a**	**b**
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
* q_2	q_2	q_2

à Show that the string w= bbabb is accepted by the given FA, M.

$$\hat{\delta}(q_0,\underline{b}babb) = \hat{\delta}(q_0,\underline{b}abb)$$
$$=\hat{\delta}(q_0,\underline{a}bb)$$
$$=\hat{\delta}(q_1,\underline{b}b)$$
$$=\delta(q_2,\underline{b})$$
$$=q_2 \in F$$

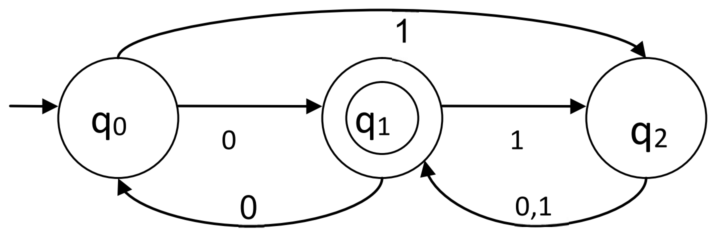
There is a path from
à starting state to final state.

$$q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2 F$$

Therefore the given string is accepted.

Problem	**1.2**

Consider the following DFA. Compute $\hat{\delta}(q_0,1101)$



$\hat{d}(q_0,1101) = (q_2,101) = (q_1,01) = (q_0,1) = q_2 \in F$

So the string is not accepted.

1.7 non-determInIstIc FInIte AutomAtA(nFA)

NFA is the simple and initial model of computation .Constructing Automata to recognize a Language includes the following steps:

r Design an NFA r

Convert NFA to DFA

r Minimize the DFA

1.7 .1 Designing NFA for a language

It is very easy to design NFA for a language by considering the common (compulsory) part of the strings in a given language. There are two types of NFAs.

r NFA without ε-Transitions

r ε-NFA

à Designing NFA without ε-Transitions for a language

Problem	**1.3**

Design an NFA for the following finite languages over the alphabet {a,b}

a. L={ε}

b. L={a}

c. L={b}

d. L={a,b}

e. L={aa,ab}

f. L={aba,abb,aaa}

Solutions:

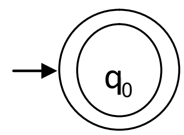
a. L={ε}

NFA M=(Q, Σ, δ, q_0, {q_0})

Where Q={ q_0}

Σ={a,b}　　　**δ :**

Transition diagram



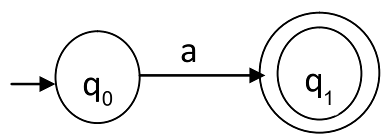
b. L={a}

NFA M=(Q, Σ, δ, q_0, {q_1})

Where Q={ q_0, q_1}

Σ={a,b}　　　**δ :**

Transition diagram



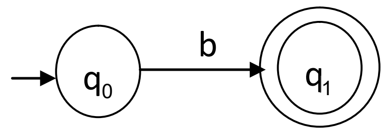
c. L={b}

NFA M=(Q, Σ, δ, q_0, {q_1})

Where Q={ q_0, q_1}

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



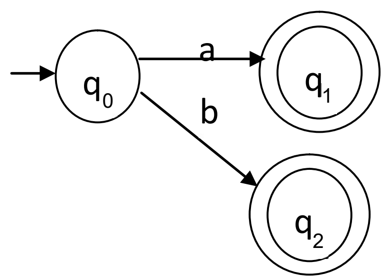
d. L={a,b}

NFA M=(Q, Σ, δ, q_0, $\{q_1, q_2\}$)

Where Q={ q_0, q_1, q_2}

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



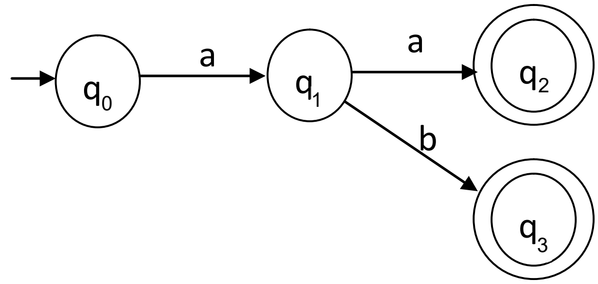
e. L={aa,ab}

NFA M=(Q, Σ, δ, q_0, $\{q_2, q_3\}$)

Where Q={ q_0, q_1, q_2, q_3}

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



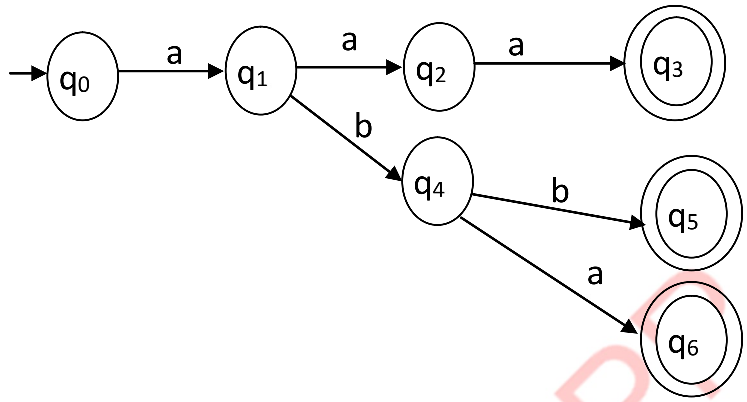
f. L={aba,abb,aaa}

NFA $M=(Q, \Sigma, \delta, q_0, \{q_3, q_5, q_6\})$

Where $Q=\{ q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$\Sigma=\{a,b\}$

δ : Transition diagram



Problem	1.4

Design an NFA without ε-Transitions for the following infinite languages over the alphabet {a,b}.

a. The set of all strings ending in aa (L={w| w ends in aa})

b. The set of all strings with the substring aba(L={w| w has substring aba}).

c. The set of all strings beginning with bb (L={w| w begins with bb)}).

d. The set of all strings with even number of a's (L={w| w has even number of a's}).

e. The set of all strings with even number of b's(L={w| w has even number of b's }).

f. The set of all strings with odd number of a's (L={w| w has odd number of a's}).

g. The set of all strings with odd number of b's (L={w| w has odd number of b's}).

h. The set of all strings whose third symbol from the right end is b (L={w| w's third symbol from the right end is b}).

i. The set of all strings whose third symbol from the left end is b (L={w| w's third symbol from the left end is b}).

Solutions:

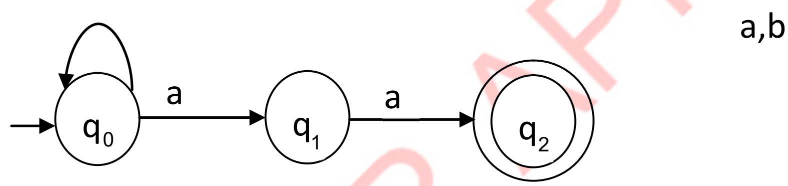
a. The set of all strings ending in aa.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_2\})$

Where $Q=\{ q_0, q_1, q_2\}$

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



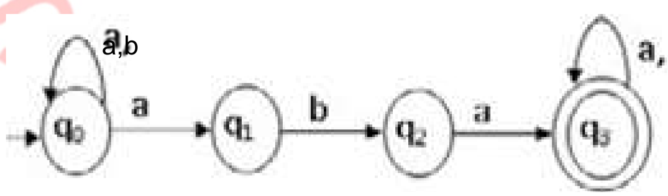
b. The set of all strings with the substring aba.

NFA $M = (Q, \Sigma, \delta, q_0, \{q_3\})$

Where $Q=\{ q_0, q_1, q_2, q_3\}$

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



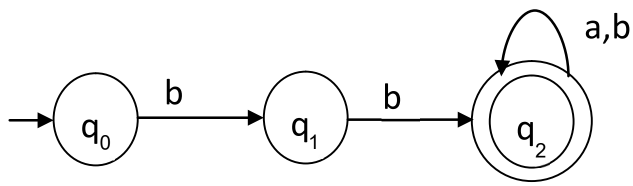
c. The set of all strings beginning with bb.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_2\})$

Where $Q=\{ q_0, q_1, q_2\}$

δ : Transition diagram

Σ={a,b}



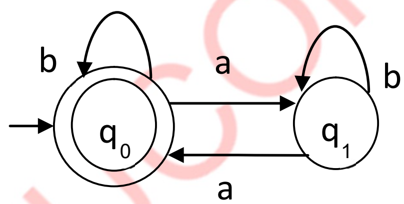
d. The set of all strings with even number of a's.

NFA M=(Q, Σ, δ, q_0, {q_0})

Where Q={ q_0, q_1}

Σ={a,b} **δ :**

Transition diagram



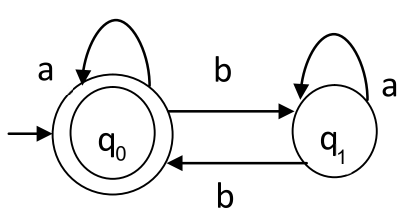
e. The set of all strings with even number of b's.

NFA M=(Q, Σ, δ, q_0, {q_0})

Where Q={ q_0, q_1}

Σ={a,b} **δ :**

Transition diagram



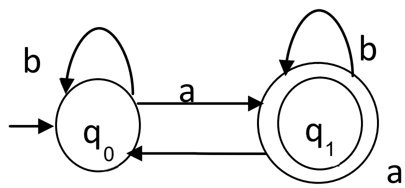
f. The set of all strings with odd number of a's.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_1\})$

Where $Q=\{ q_0, q_1\}$

$\Sigma=\{a,b\}$

δ : Transition diagram



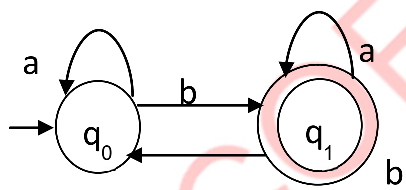
g. The set of all strings with odd number of b's.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_1\})$

Where $Q=\{ q_0, q_1\}$

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



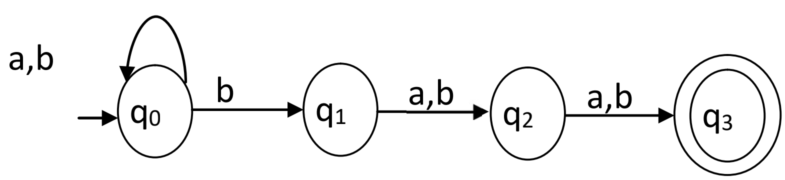
h. The set of all strings whose third symbol from the right end is b.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_3\})$

Where $Q=\{ q_0, q_1 q_2, q_3\}$

$\Sigma=\{a,b\}$ **δ :**

Transition diagram

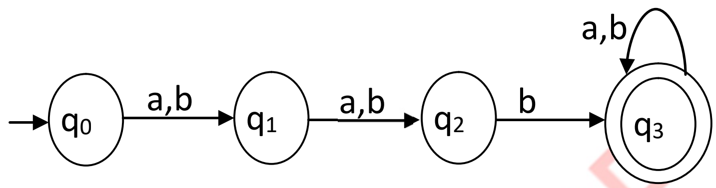


i. The set of all strings whose third symbol from the left end is b.

NFA $M=(Q, \Sigma, \delta, q_0, \{q_3\})$

Where $Q=\{ q_0, q_1, q_2, q_3 \}$

$\Sigma=\{a,b\}$



1. Automata for $\bar{L}$

If the automata is given for a language $\bar{L}$, then the automata for can be easily constructed by changing all the non-final states to final states and final states to non-final states.

Given:

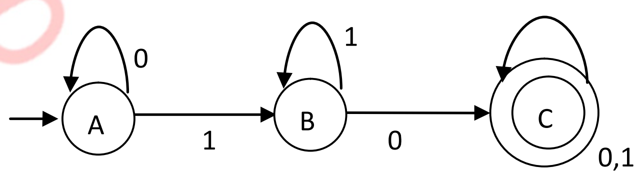
The FA of L= {w | w consists of 10 as substring}

$M(L) = (Q, \Sigma, \delta, A, \{C\})$

Where Q={ A, B ,C }

$\Sigma=\{a,b\}$ **δ :**

Transition diagram



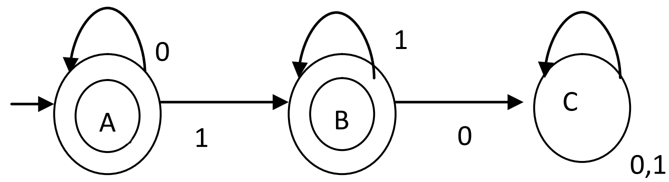
$M(\bar{L}) = (Q, \Sigma, \delta, A, \{A, B\})$

Where Q={ A, B ,C }

$\Sigma=\{a,b\}$ **δ :**

Transition diagram

δ : Transition diagram



2. Automata for L $L_1 \cap {}_2$

The intersection of two regular languages can be constructed by taking Cartesian product of states.

Let, $M(L_1) = (Q_1, \Sigma, \delta_1, q_1, F_1)$

$M(L_2) = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Then $M(L_1 \cap L_2) = (Q_2 \text{ X } Q_2, \Sigma, \delta, (q_1, q_2), F_2 \text{ X } F_2)$

Problem	1.5

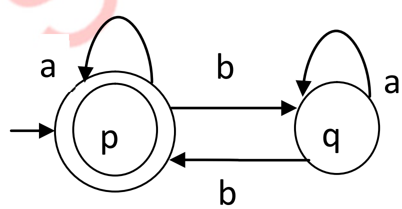
Let L1 = The set of all strings with even number of b's.

NFA $M(L_1) = (Q, \Sigma, \delta_1, p, \{p\})$

Where Q={p, q }

$\Sigma=\{a,b\}$ **δ_1 :**

Transition diagram



Problem	1.6

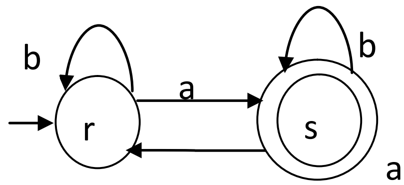
Let L_2 = The set of all strings with odd number of a's.

NFA $M(L_2) = (Q, \Sigma, \delta 2, r, \{s\})$

Where $Q=\{ r, s \}$

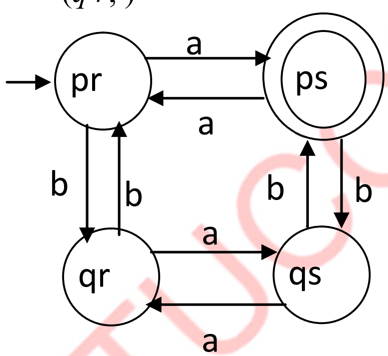
$\Sigma=\{a,b\}$

δ_2 : Transition diagram



Then, $M(L_1 \cap L_2) = (\{pr, ps, qr, qs\}, \{a,b\}, \delta, pr, ps)$

$$d(pr\ a,) = (d_1(p\ a,), d_2(r\ a,))$$
$$= (p\ s,)$$
$$d(pr\ b,) = (d_1(p\ b,), d_2(r\ b,))$$
$$= (q\ r,)$$



1.7.2 String Processing in NFA

Problem	1.7

For the NFA M given in the following table, test whether the strings 01122, 1221 are accepted by M.

δ	0	1	2
→*q0	{q0, q1, q2 }	{q1, q2 }	{q2 }
*q1	Φ	{q1, q2 }	{q2 }

*q2	Φ	Φ	{q2 }

i. $\hat{\delta}(q_0,\underline{0}1122) = \hat{\delta}(\{q_0, q_1, q_2\},\underline{1}122)$

$= \hat{\delta}(\{\delta(q_0,1)\cup\delta(q_1,1)\cup\delta(q_2,1)\},122)$

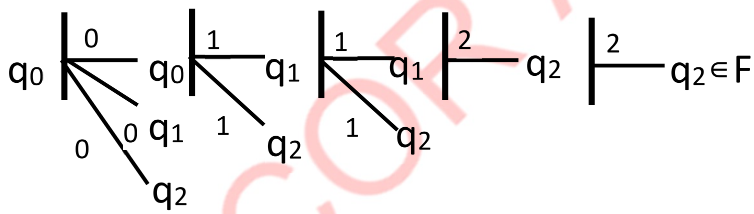
$= \hat{\delta}(\{q_1, q_2\},\underline{1}22)$

$= \hat{\delta}(\{q_1, q_2\},\underline{2}2)$

$= \hat{\delta}(\{q_2\},\underline{2})$

$= q_2 \in F$

There is at least one path from the starting state to final state. Therefore the given string is accepted.



ii. $\hat{\delta}(q_0,\underline{1}221) = \hat{\delta}(\{q_1, q_2\},\underline{2}21)$

$= \hat{\delta}(\{\delta(q_1,2)\cup\delta(q_2,2)\},21)$

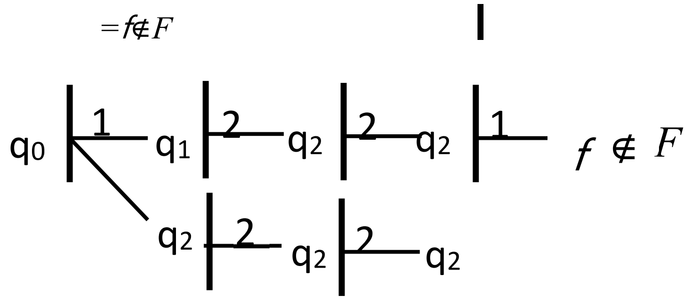
$= \hat{\delta}(\{q_2\},\underline{2}1)$

$= \hat{\delta}(\{q_2\},\underline{1})$

$\vdash^{1} \phi \notin F$

$=f \notin F$

a
state
the



There is no even single path from starting to final state. Therefore given string is not accepted.

1.7.3 Equivalence of NFA and DFA (Converting NFA to DFA)

Theorem	**12**

A Language L is accepted by some DFA if and only if L is accepted by some NFA.

Proof by induction

The "if" part : If L is accepted by some NFA then L is accepted by some DFA. If $D=\{Q_D , \Sigma, \delta_D, \{q_0\},F_D\}$ is the DFA constructed from NFA, $N=\{Q_N , \Sigma, \delta_N, q_0,F_N\}$ by the subset construction ,then L(D) = L(N).

Proof

To prove by induction on $|w|$ $\hat{\delta}_D(\{q_0\},w) = \hat{\delta}_N(q_0, w)$ ... (1.6)

Observe that each of the $\hat{\delta}$ functions returns a set of states from Q_N, but $\hat{\delta}_D$ interprets this set as one of the states of Q_D (which is the power set of Q_N), while interprets this set as a subset of Q_N.

Basis

Let $|w| = 0$; that is, $w=e$. By the basis definitions of $\hat{\delta}$ for DFA's and NFA's, both

$\hat{\delta}_D(\{q_0\}, e)$ and $\hat{\delta}_N(q_0, e)$ are $\{q_0\}$

Induction

Let $|w| = n+1$, and assume the statement for length n. Break w as $w=xa$, where a is the final symbol of w.

By the inductive hypothesis,

$$\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x)$$

Let both these sets of N's states be $\{P_1, P_2, \ldots\ldots P_k\}$ .i.e.

$$\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x) = \{p_1, p_2, \ldots. p_k\} \qquad \ldots (1.7)$$

The inductive part of the definition of for the NFA's say that

$$\hat{\delta}_N(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \qquad \ldots (1.8)$$

The subset construction, on the other hand , says that

$$\delta_D(\{p_1, p_2, \ldots. p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \qquad \ldots (1.9)$$

From (1.7) and (1.9), the inductive part of the definition of $\hat{\delta}$ for DFA is written as: $\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$

$$\ldots (1.10)$$

$$= \delta_D(\{p_1, p_2, \ldots. p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$$

Thus, equations (1.8) and (1.10) demonstrate that

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

When we observe that D and N both accept w if and only if $\hat{\delta}_D(\{q_0\},w)$ or $\hat{\delta}_N(q_0, w)$ respectively, contain a state in F_N.

Hence, L(D) = L(N) is proved.

The "only if" part

If L is accepted by some DFA then L is accepted by some NFA.

We have only to convert a DFA into identical NFA.Put intuitively, if we have the transition diagram for a DFA, we can also Interpret it as the transition diagram of an NFA, which happens to have exactly one choice of transition in any situation.

More formally, let $D=\{Q, \Sigma, \delta_D, q_0,F\}$ be a DFA. Define $N=\{Q, \Sigma, \delta_N, q_0,F\}$ to be the equivalent NFA.

Where, δ_N is defined by the rule:

If $\delta_D(q, a) = p$ then $\delta_N(q, a) = \{p\}$

It is then easy to show by induction on $|w|$, that if $\hat{\delta}_D(q_0, w) = p$ then $\hat{\delta}_N(q_0, w) = \{p\}$

As a consequence, ω is accepted by D if and only if it is accepted by N; i.e ., L(D)=L(N).

Subset construction method (with 'Lazy Evaluation') is used to convert NFA to DFA. In this method the transition functions are generated only for reachable states.

Method 1

Steps

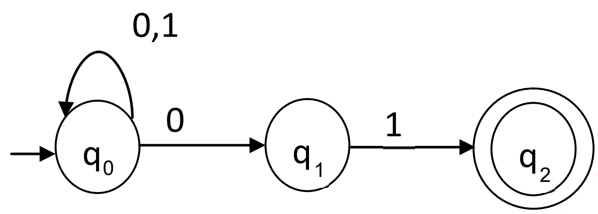
1. Include the starting state of NFA (q0) in DFA as starting state of DFA.
2. Find the transition for all the symbols from q0
3. If the output state is new state, include it in DFA and find the transition for all the symbols from that state.
4. Repeat step3 until there are no more new states.

5. The state which includes final state of NFA is the final state of DFA.

Problem	1.8

Construct the DFA for the L={w|w ends in 01}

Transition Diagram of NFA



Transition Table of NFA

δ_D	0	1	2
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_2\}$
* $\{q_0, q_2\}$	$\{q0, q_1\}$	$\{q_0\}$	$\{q_2\}$

Step 1:

Include q_0

Step 2:

Find transitions for 0,1 from q_0.

$\delta(q_0,0)=\{q_0, q_1\}$ - New state $\delta(q_0,1)=\{q_0\}$
- Existing state

Step 3:

Find transitions for 0,1 from new state.

$\delta(\{q\ q_{0},\ _{1}\},0) = \delta(q_0,0) \cup \delta(q_1,0)$

$= \{q\ q_{0},\ _{1}\} \cup = \phi\ \{q\ q_{0},\ _{1}\}$ Existing state $\delta(\{q$

$q_{0},\ _{1}\},1) = \delta(q_0,1) \cup \delta(q_1,1)$

$= \{q_0\} \cup \{q_2\} = \{q\ q_{0},\ _{2}\}$ New state

Step 4:

Repeat step 3 for new state (s).

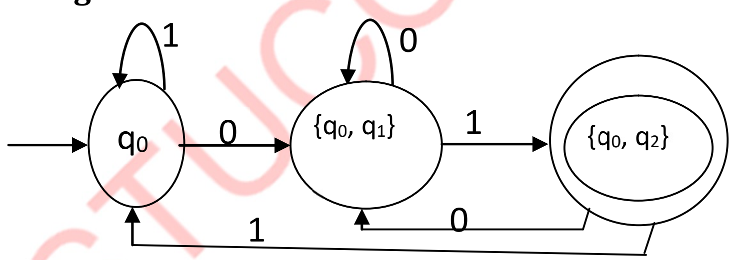
$\delta(\{q\ q_{0},\ _{2}\},0) = \delta(q_0,0) \cup \delta(q_2,0)$

$= \{q\ q_{0},\ _{1}\} \cup = \phi\ \{q\ q_{0},\ _{1}\}$ Existing state $\delta(\{q$

$q_{0},\ _{2}\},1) = \delta(q_0,1) \cup \delta(q_2,1)$

$= \{q_0\} \cup \phi\} = \{q_0\}$ New state

Transition Diagram of DFA



Method 2

Input: Transition table of NFA

Output: Transition table of DFA

Steps

1. Draw the transition table for NFA (if not given)
2. Copy the first row of NFA table (transition function of start state) to DFA table.
3. The entries are considered as states of DFA.

4. If there is any new state, find the transition function for that new state using the following formula:

$$d_D(\{q_1,\ldots,q_k\},a) = \bigcup_{i=1}^{k} d_N(q\ a_i,)$$

5. Continue Step 4 until no more new states.

Transition Table of DFA

δ_D	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\ \{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$

Procedure

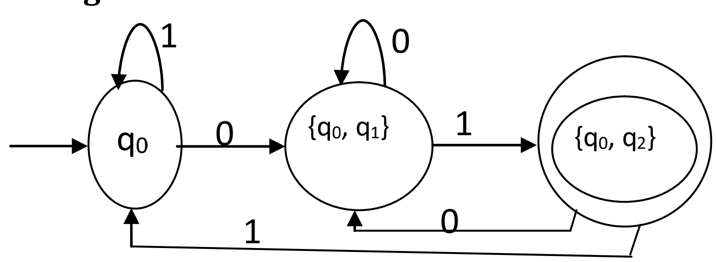
^ Copy the first row. $\{q_0, q_1\}$ is the new state.

^ Union of q_0 row and q_1 row. $\{q_0, q_2\}$ is the new state.

^ Union of q_0 row and q_2 row.

^ No more new states. So Stop

Transition Diagram of DFA



$Q = \{q_0, \{q_0,q_1\}, \{q_0,q_2\}\}$

$\Sigma = \{0,1\}$ $q_0 = q_0$

$F = \{q_0,q_2\}$

Problem	1.9

Consider the following NFA. Convert it into DFA.

Transition Table of NFA

δ_N	**a**	**b**
$\rightarrow q_0$	$\{q_0,q_1\}$	$\{q_0\}$
q_1	q_2	q_1
q_2	q_3	q_3
$* q_3$	-	q_2

Procedure

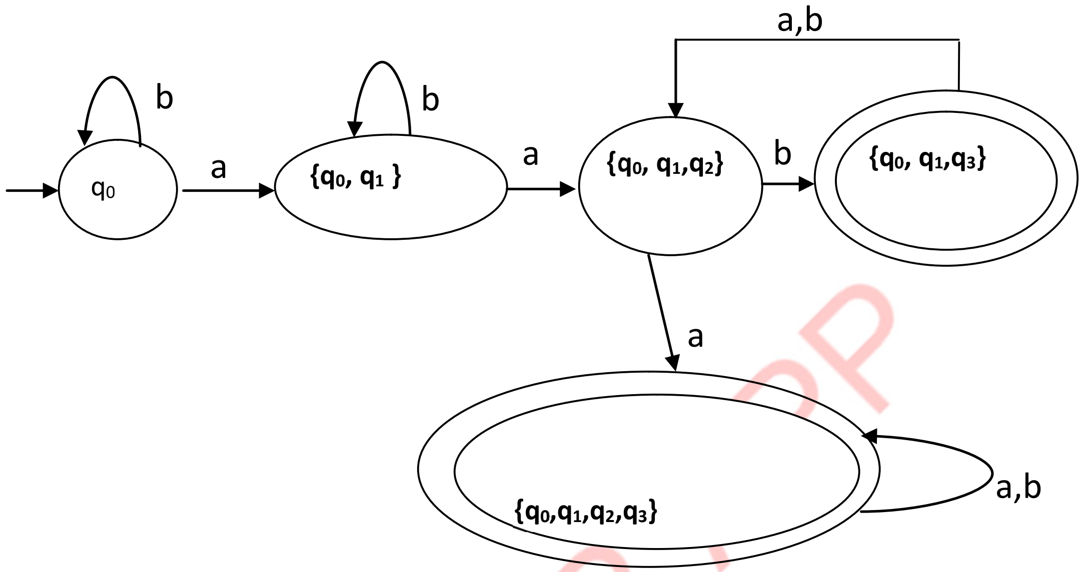
^ Copy the first row.

^ Identify the new state.

^ Find the transition for new state using Union operation.

^ Stop, if no more new states.

Transition Diagram of DFA



Problem	**1.10**

Convert to the DFA the following NFA.

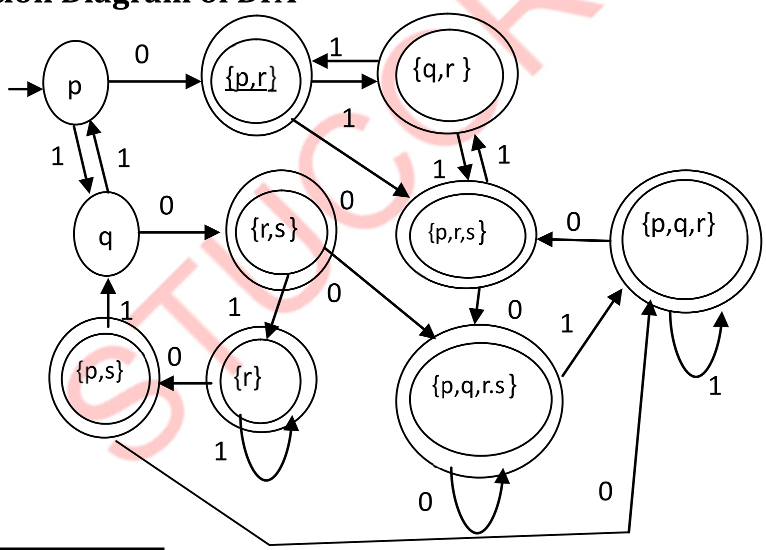
δ_N	**0**	**1**
→p	{ p,r }	{q}
q	{ r,s }	{ p}
*r	{ p,s}	{ r }
* s	{ q,r }	-

Transition Table of DFA

δ_D	**0**	**1**
→p	{ p,r }	{ q}
{ q}	{ r,s }	{ p}

*{ p,r }	{ p,r,s }	{ q,r }
*{ r,s }	{ p,q,r,s }	{ r}
*{ p,r,s }	{ p,q,r,s }	{ q,r }
*{ r}	{ p,s}	{ r }
*{ p,q,r,s }	{ p,q,r,s }	{ p,q,r }
*{ p,s}	{ p,q,r }	{ q}
*{ p,q,r }	{ p,r,s }	{ p,q,r }
*{ q,r }	{ p,r,s }	{ p,r }

Transition Diagram of DFA



Problem	1.11

Convert the following NFA to a DFA and informally describe the language it accepts.

Transition table of given NFA

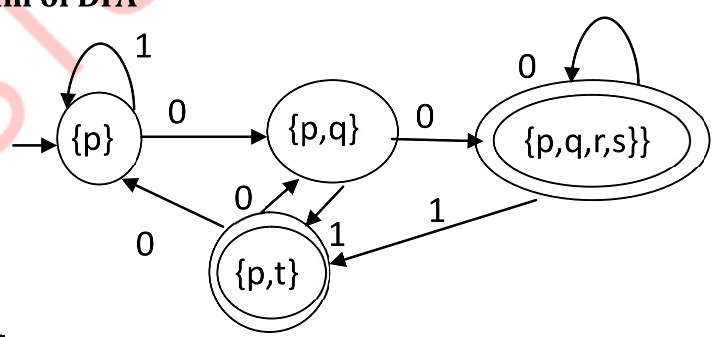
δ_N	0	1

→p	{ p,q }	{ p}
q	{ r,s }	{ t}
r	{ p,r}	{ t }
* s	-	-
*t	-	-

Transition table of DFA

δ_D	**0**	**1**
→p	{ p,q }	{ p}
{ p,q}	{p,q, r,s }	{ p,t}
*{ p,q,r,s }	{ p,q,r,s }	{ p,t }
*{p,t}	{ p,q }	{ p}

Transition Diagram of DFA



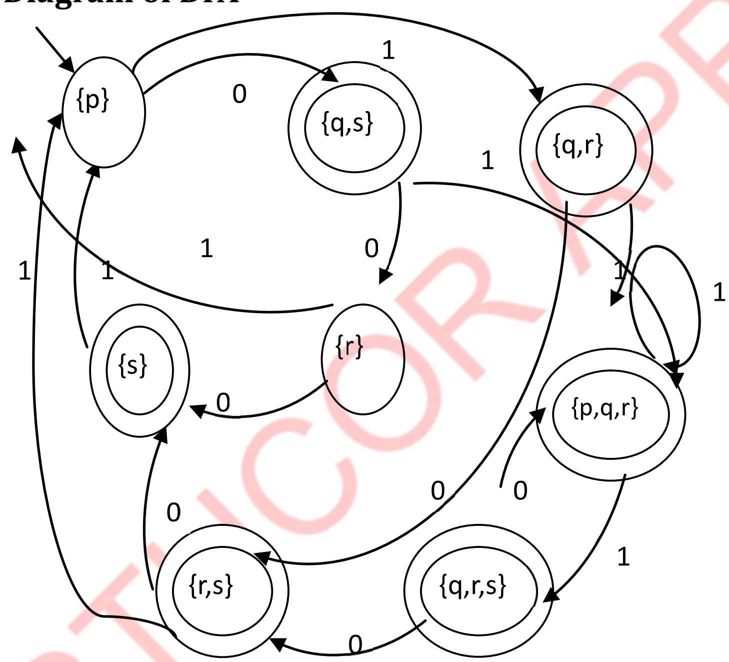
Problem	1.12

Convert to a DFA the following NFA.

	0	**1**

→p	{q,s}	{q}
*q	{r}	{q,r}
r	{s}	{p}
*s	-	{p}

Transition Diagram of DFA



Language of DFA à The language of a DFA

is defined by,

$$L(DFA)=\{w \mid \hat{\delta}(q_0,w \text{ is in } F\})$$

à And the language of a NFA is defined by,

$$L(NFA) = \{w \mid (\hat{\delta}\, q_0,w) \cap F \neq \phi\}$$

* Where q_0 is the start state * F is the set of final states

  and * *w* is a string.

* L(DFA) and L(NFA) are called Regular Languages.

1.8 FInIte AutomAtA wIth ePsIlon trAnsItIons

Finite Automata with Epsilon transitions is also called as ε-NFA . It contains epsilon edges. In transition table a column is allocated for epsilon and it gives the output for epsilon input.

+ A Non-Deterministic finite automaton with ε- Transitions (NFA) is represented by 5-tuples.

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

  * Q is a finite non-empty set of states.
  * Σ is a finite non-empty set of symbols (an alphabet)
  * $\delta : Q X \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$ is the transition function
  * $q_0 \in Q$ is the start state
  * $F \in Q$ is a set of final states

Transition Table of ε-NFA

δ_N	ε	a	b	c
→p	Φ	{p}	{q}	{r}
q	{p}	{q}	{r}	Φ
*r	{q}	{r}	Φ	{p}

ε-Closure

Epsilon closure of a state is the set of all states that are reachable by following the transition function from the given state through ε edge.

Problem	**1.13**

Consider the ε-NFA.Compute ε-Closure for each state.

δ_N	**ε**	**0**	**1**	**2**
$\rightarrow q_0$	q_1	q0	Φ	Φ
q_1	q_2	Φ	q_1	Φ
$*q_2$	Φ	Φ	Φ	q_2

r ε-Closure (q_0) = $\{ q_0 , q_1, q_2\}$

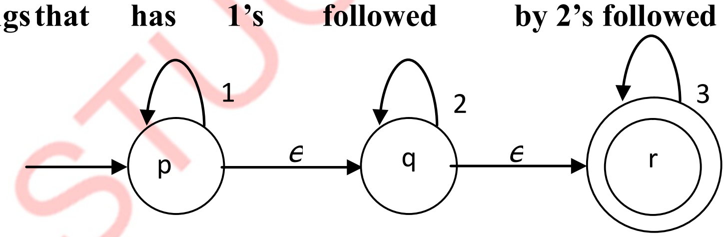
r ε-Closure (q_1) = $\{ q_1, q_2\}$ r

ε-Closure (q_2) = $\{q_2\}$

1.8.1 Designing an ε-NFA or NFA with ε-Transitions

Problem	**1.14**

Design an ε-NFA for the language which consists of strings that has 1's followed by 2's followed by 3's.



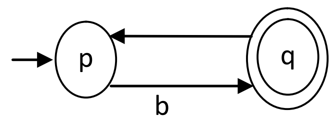
Q = {p,q,r}

Σ = {1,2,3}

Problem	**1.15**

Design an ε-NFA for the language b+.

e

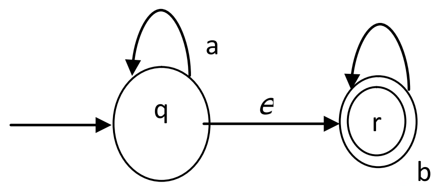


Q = {p,q}

Σ = {b}

Problem	1.16

Design an ε-NFA for the language which consists of strings with all a's followed by all b's.



Q = {q,r}

Σ = {a,b}

1.8.2 String Processing in ε-NFA

Problem	1.17

For the ε-NFA M given in the following table, test whether the strings aabccand abba are accepted by M.

δ	ε	a	b	c
→p	{q}	{p}	Φ	Φ
q	{r}	Φ	{q}	Φ

*r	Φ	Φ	Φ	{r}

Step 1:

Compute ε-Closure [states that can be reached by traveling along zero or more ε arrows] for all states .

r ε-Closure (p) = {p,q,r} $\hat{\delta}(p,\epsilon)$ r

ε-Closure (q) = {q,r} $\hat{\delta}(q,\epsilon)$

r ε-Closure (r) = {r} $\hat{\delta}(r,\epsilon)$

Step2:

Start with ε-closure (p)= {p,q,r}

Where, p is the starting state of given ε –NFA.

1. (p)= {p,q,r}

$\hat{\delta}(\{p,q,r\},\underline{a}abcc)$

$= \hat{\delta}(\epsilon - closure((\delta(p,a) \cup \delta(q,a) \cup \delta(r,a)),\underline{a}bcc)$

$= \hat{\delta}(\epsilon - closure(p), abcc)$

$= \hat{\delta}(\{p,q,r\},\underline{a}bcc)$

$= \hat{\delta}(\epsilon - closure((\delta(p,a) \cup \delta(q,a) \cup \delta(r,a)),bcc)$

$= \hat{\delta}(\{p,q,r\},\underline{b}cc)$

$= \hat{\delta}(\epsilon - closure((\delta(p,b) \cup \delta(q,b) \cup \delta(r,b)),\underline{c}c)$

$= \hat{\delta}(\{q,r\},\underline{c}c)$

$= \hat{\delta}(\epsilon - closure((\delta(q,c) \cup \delta(r,c)),\underline{c})$

$= \hat{\delta}(\{q,r\},\underline{c})$

$= r \in F$

∴ Therefore the given string is accepted.

2. *w=abba*

$$\hat{\delta}(\{p,q,r\},\underline{a}bba)$$

$$= \hat{\delta}(\varepsilon-closure(\delta(p,a) \cup \delta(q,a) \cup \delta(r,a)),\underline{b}ba)$$

$$= \hat{\delta}(\varepsilon-closure(p), bba)$$

$$= \hat{\delta}(\{p,q,r\},\underline{b}ba)$$

$$= \hat{\delta}(\varepsilon-closure(\delta(p,b) \cup \delta(q,b) \cup \delta(r,b)),ba)$$

$$= \hat{\delta}(\{q,r\},\underline{b}a)$$

$$= \hat{\delta}(\varepsilon-closure(\delta(q,b) \cup \delta(r,b)),\underline{a})$$

$$= \hat{\delta}(\{q,r\},\underline{a})$$

$$= \varphi \notin F$$

∴ Therefore the given string is not accepted.

1.8.3 Equivalence of ε-NFA and DFA.

An ε-NFA can be converted into DFA. The subset construction method (with 'Lazy Evaluation') is used to convert ε-NFA to DFA. In this method the transition functions are generated only for reachable states.

Input: Transition table of ε-NFA

Output: Transition table of DFA

Theorem

A language L is accepted by some -NFA if and only if L is accepted by some DFA.

Proof

1. If part: If the L is accepted by some DFA then L is accepted by some -NFA

Suppose L=L(D)for some DFA. Turn D into an ε-NFA by adding transitions $d(q,e) = j$ for all states q of D. Technically we must also convert the transitions of D on input symbols, example, $d_D(q\ a,) = p$, into an NFA-transition to the set containing only p, that is, $d_E(q\ a,) = \{\ \}p$

Thus, the transitions of E and D are the same , but E explicitly states that there are no transitions out of any state on ε.

2. Only -If part: If the L is accepted by some -NFA then L is accepted by some DFA.

Let E = {Q_E, Σ, δ_E, q_0, F_E} be an ε -NFA. Apply the modified subset construction to produce the DFA.

D = {Q_D, Σ, δ_D, q_D, F_D}

^ We need to show that L(D) = L(E), and we do so by showing that the extended transition functions of E and D are the same.

^ Formally, we show $\hat{d}_E(q_0,w) = \hat{d}_D(q_D,w)$ by induction on the length of ω .

Basics

* If $|w|$then w=ε.

* We know $\hat{d}_E(q_0,)e = ECLOSURE\ q(_0)$

* We also know that $q_D = ECLOSURE\ q(_0)$, because that is how the start state of D is defined.

* Finally, for a DFA, we know that $\hat{d}(p,e) = p$ for any state p, so in particular $\hat{d}_D(q_D,)e = ECLOSURE\ q(_0)$.

* We have thus proved that $\hat{d}_E(q_0,)e = \hat{d}_D(q_D,)e$.

Induction

* Suppose w=xa.

r Where, a is the final symbol of w and assume that the statement holds for x.

* That is, $\hat{\delta}_E(q_0, x) = \hat{\delta}_D(q_D, x)$.

* Let both these sets of states be $\{p_1, p_2, \ldots p_k\}$. By the definition of δ for ε-NFA's, we compute $\delta_E(q_0, w)$ by,

  i. Let $\{r_1, r_2, \ldots r_m\}$ be $\bigcup_{i=1}^{k} \delta_E(p_i, a)$.

  ii. Then $\hat{\delta}_E(q_0, w) = \bigcup_{j=1}^{m} ECLOSURE(r_j)$

* If we examine the construction of DFA D in the modified subset construction, we see that $\delta_D(\{p_1, p_2, \ldots p_k\}, a)$ is constructed by the same above two steps (i) and (ii).

* Thus, $\hat{\delta}_D(q_D, w)$, which is $\delta_D(\{p_1, p_2, \ldots p_k\}, a)$ is the same set as $\hat{\delta}_E(q_0, w)$.

* We have now proved that $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w)$ and completed the inductive part.

Steps to convert ε-NFA to DFA

a. Compute the ε-Closure for each state.

b. Draw the transition table for ε-NFA (if not given)

c. Start state of DFA is ε-Closure(q_0)

   • Where q_0 is the start state of ε-NFA .

d. Find the transition function for ε-Closure(q_0).

e. The entries are considered as states of DFA.

f. If there is any new state, find the transition function for that new state using the following formula:

$$d_D(\{q_1,...,q_k\},a) = \bigcup_{i=1}^{k} e-closure(d_N(q\ a_i,))$$

g. Continue the above step 'f' until no more new states.

1.8.4 Applications and Limitations of FA

1. Applications of FA

i. Text Search

a. **News Analyst** - Searches on-line news

b. **Shopping robot** – Searches current prices charged for an item

c. **Amazon.com** - Search some keywords

d. **Lexical analyzer of a compiler** - Identifies the token

* Verifying the working of a physical system

* Design and construction of Softwares ***ii.***

Advantages of Finite set of states in Automata

^ Implement a system with a fixed set of resources

^ Implementing a system within a hardware circuit

^ Complementing a system using software with a finite set of codes.

2. Limitations of FA

^ Some languages are not regular – i.e. we cannot construct FA ***Example:***

* $B = \{0^n 1^n \mid n \geq 0\}$ is NOT regular!

* $L=ww^R$ * $L=WW$

* $L=WCWR$

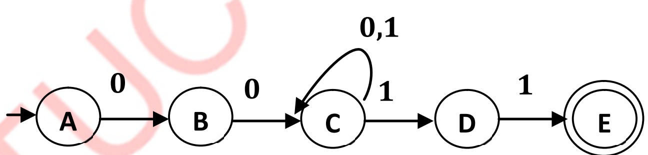
* $C = \{ w \mid w$ has equal number of 1s and 0s$\}$

1.8.5 Complex Problems

1. Design a NFA that accepts set of all strings that begins with 00 and ends with 11. Convert it into DFA.

Analysis

- Here we have two parts:
  - Begins with string1
  - Ends with string2
- Let string1 be considered as s1s2 and string 2 be considered as s3s4 where s1,s2,s3 and s4 are substrings.
- For all s2 and s3, if s2≠s3, we can easily construct the NFA.
- In this problem there is no such s2 and s3 where s2=s3. Therefore we can construct the NFA in one step as follows:



- The DFA of this machine is given below:

δ_D	0	1
→ {A}	{B}	-
{B}	{C}	-
{C}	{C}	{C,D}
{C,D}	{C}	{C,D,E}
* {C,D,E}	{C}	{C,D,E}

Note: It is difficult to draw the NFA for the following languages wheres2=s3.

^ Set of all strings that begins with 01 and ends with 11 [s2=1]

^ Set of all strings that begins with 01 and ends with 10 [s2=1]

^ Set of all strings that begins with 01 and ends with 01 [s2=01]

^ Set of all strings that begins with 10 and ends with 10 [s2=10]

^ Set of all strings that begins with 00 and ends with 00 [s2=00]

^ Set of all strings that begins with 11 and ends with 11 [s2=00]

For these kinds of problems we can use the intersection property of regular languages.

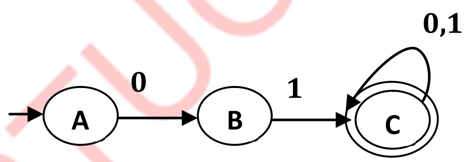
Problem	1.18

Design a DFA that accepts set of all strings that begins with 01 and ends with 11.

à There are three steps, that are given below.

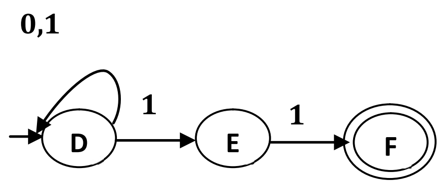
Step 1:

Design a DFA that accepts set of all strings that begins with 01



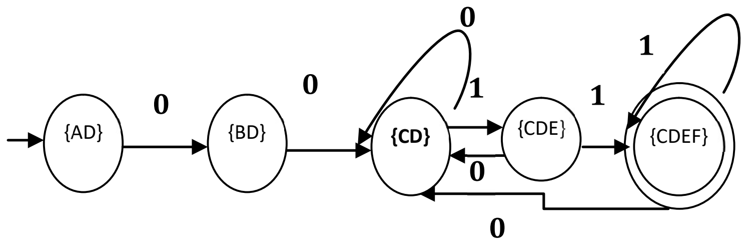
Step 2:

Design a DFA that accepts set of all strings that ends with 11.



Step 3:

Intersection between two DFAs (Lazy Evaluation-processing only reachable nodes)



1.8.6 PROBLEMS

1. **Consider the following ε-NFA.Covertit into DFA Transition Table of ε-NFA**

δ_N	ε	a	b	c
→p	Φ	{p}	{q}	{r}
q	{p}	{q}	{r}	Φ
*r	{q}	{r}	Φ	{p}

Step 1:

Compute ε-Closure [states that can be reached by traveling along zero or more ε arrows] for all states.

r ε-Closure (p) = {p} $\hat{d}(p,e)$

r ε-Closure (q) = {p,q} $\hat{d}(q,e)$

r ε-Closure (r) = {p,q,r} $\hat{d}(r,e)$

Step 2:

Start with ε-closure (p)= {p}

Where, p is the starting state of given ε –NFA.

Step 3:

Find the transition for{p} $\delta_D(\{\ \},$

$)p\ a = -\epsilon\ closure(\delta_N(p\ a,))$

$= -\epsilon\ closure\ p(\ )$

$= \{\ \}p$ *New State*

$\delta_D(\{\ \},)p\ b = -\epsilon\ closure(\delta_N(p\ b,))$

$= -\epsilon\ closure\ q(\ )$

$= \{p\ q$ *New State* $, \}$

$\delta_D(\{\ \},)pc = -\epsilon\ closure(\delta_N(p\ c,))$

$= -\epsilon\ closure\ r(\ )$

$= \{p\ q\ r,,\}$

Step 4:

Find the transition for {p,q}

$\delta_D(\{p\ q,\},a) = -\epsilon\ closure(\delta_N(p\ a,) \cup \delta_N(q\ a,))$

$= -\epsilon\ closure\ p\ q(\ ,)$

$= \{p\ q,\}$

$\delta_D(\{p\ q,\},b) = -\epsilon\ closure(\delta_N(p\ b,) \cup \delta_N(q\ b,))$

$= -\epsilon\ closure\ q\ r(\ ,)$

$= \{p\ q\ r,,\}$

$\delta_D(\{p\ q,\},)c = -\epsilon\ closure(\delta_N(p\ c,) \cup \delta_N(q\ c,))$

$= -\epsilon\ closure\ r(\ )$

$= \{p\ q\ r,,\}$

Step 5:

Find the transition for {p,q,r}

$\delta_D(\{p\ q\ r,,\},a) = -\epsilon\ closure(\delta_N(p\ a,) \cup \delta_N(q\ a,) \cup \delta_N(r\ a,))$

$= -\epsilon\ closure\ p\ q\ r(\ ,,)$

$= \{p\ q\ r,,\}$

$\delta_D(\{p\ q\ r,,\},b) = -\epsilon\ closure(\delta_N(p\ b,) \cup \delta_N(q\ b,) \cup \delta_N(r\ b,))$

$= -\epsilon\ closure\ q(\ ,r)$

$$= \{p, q, r\}$$

$$\delta_D(\{p, q, r\}, c) = \epsilon\text{-}closure(\delta_N(p, c) \cup \delta_N(q, c) \cup \delta_N(r, c))$$

$$= \epsilon\text{-}closure(p, r)$$

$$= \{p, q, r\}$$

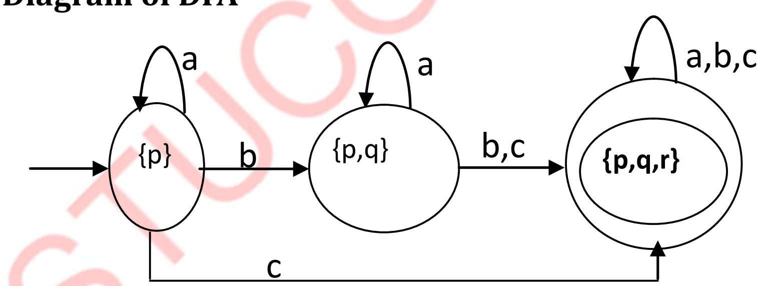
Step 6:

No more new states. Stop the process.

Transition Table of DFA

δ_D	a	b	c
→{p}	{p}	{p,q}	{p,q,r}
{p,q}	{p,q}	{p,q,r}	{p,q,r}
*{p,q,r}	{p,q,r}	{p,q,r}	{p,q,r}

Transition Diagram of DFA



2. Consider the following ε-NFA.Covertit into DFA

δ	ε	a	b	c
→p	{q,r}	Φ	{q}	{r}
q	Φ	{p}	{r}	{p,q}
*r	Φ	Φ	Φ	Φ

Step 1:

Compute ε-Closure [states that can be reached by traveling along zero or more ε arrows] for all states .

r ε-Closure (p) = {p,q,r} $\hat{\delta}(p,\epsilon)$

r ε-Closure (q) = {q} $\hat{\delta}(q,\epsilon)$

r ε-Closure (r) = {r} $\hat{\delta}(r,\epsilon)$

Step 2:

Start with ε-closure (p)= {p,q,r}

Where, p is the starting state of given ε - NFA

Step 3:

Find the transition for {p,q,r}

$$\delta_D(\{p,q,r\},a) = \epsilon - closure(\delta_N(p,a) \cup \delta_N(q,a) \cup \delta_N(r,a))$$

$$= \epsilon - closure(p)$$

$$= \{p,q,r\}$$

$$\delta_D(\{p,q,r\},b) = \epsilon - closure(\delta_N(p,b) \cup \delta_N(q,b) \cup \delta_N(r,b))$$

$$= \epsilon - closure(q,r)$$

$$= \{q,r\}$$ New State δ_D

$$(\{p,q,r\},c) = \epsilon - closure(\delta_N(p,c) \cup \delta_N(q,c) \cup \delta_N(r,c))$$

$$= \epsilon - closure(p,q,r)$$

$$= \{p,q,r\}$$

Step 4:

Find the transition for {q,r}

$$\delta_D(\{q,r\},a) = \epsilon - closure(\delta_N(q,a) \cup \delta_N(r,a))$$

$$= \epsilon - closure(p)$$

$$= \{p, q, r\}$$

$$\delta_D(\{q, r\}, b) = \varepsilon\text{-}closure(\delta_N(q, b) \cup \delta_N(r, b))$$

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

$$= \{r\}$$ New State

$$\delta_D(\{q, r\}, c) = \varepsilon\text{-}closure(\delta_N(q, c) \cup \delta_N(r, c))$$

$$= \varepsilon\text{-}closure(p, q)$$

$$= \{p, q, r\}$$

Step 5:

Find the transition for {r}

$$\delta_D(\{r\}, a) = \varepsilon\text{-}closure(\delta_N(r, a))$$

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

$$= \varphi$$ Dead State δ_D

$$(\{r\}, b) = \varepsilon\text{-}closure(\delta_N(r, b))$$

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

$$= \varphi$$

$$\delta_D(\{r\}, c) = \varepsilon\text{-}closure(\delta_N(r, c))$$

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

$$= \varphi$$

Step 6:

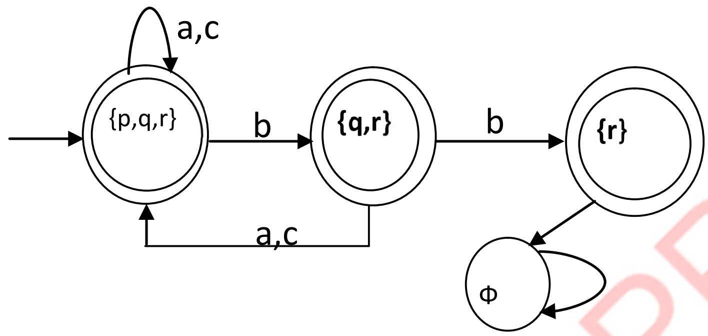
No more new states. Stop the process.

Transition Table of DFA

δ_D	a	b	c
→*{p,q,r}	{p,q,r}	{q,r}	{p,q,r}
*{q,r}	{p,q,r}	{r}	{p,q,r}

*{r}	Φ	Φ	Φ

Transition Diagram of DFA



3. Consider the following ε-NFA.Covert a,b,c it into DFA. Transition Table of ε-NFA

δ_N	ε	0	1	2
$\rightarrow q_0$	q_1	q_0	Φ	Φ
q_1	q_2	Φ	q_1	Φ
$*q_2$	Φ	Φ	Φ	q_2

Step 1:

Compute ε-Closure [states that can be reached by traveling along zero or more ε arrows] for all states .

r ε-Closure (q_0) = { q_0, q_1, q_2} $\hat{d}(q_0,\ )e$

r ε-Closure (q_1) = { q_1 ,q_2} $\hat{d}(q_1,\ )e$

r ε-Closure (q_2) = { q_2} $\hat{d}(q_2,\ )e$

Step 2:

Start with ε-closure (q_0)= { q_0, q_1, q_2}

Where, q_0 is the starting state of given ε –NFA.

Step 3:

Find the transition for { q_0, q_1, q_2}

$$d_D(\{q\ q\ q_0,{}_1,{}_2\},0) = -e\ closure(d_N(q_0,0) \cup d_N(q_1,0) \cup d_N(q_2,0))$$

$$= -e\ closure\ q(\,{}_0)$$

$$= \{q\ q\ q_0,{}_1,{}_2\}$$

$$d_D(\{q\ q\ q_0,{}_1,{}_2\},1) = -e\ closure(d_N(q_0,1) \cup d_N(q_1,1) \cup d_N(q_2,1))$$

$$= -e\ closure\ q(\,{}_1)$$

$$= \{q\ q_1,{}_2\}$$ New State

$$d_D(\{q\ q\ q_0,{}_1,{}_2\},2) = -e\ closure(d_N(q_0,2) \cup d_N(q_1,2) \cup d_N(q_2,2))$$

$$= -e\ closure\ q(\,{}_2)$$

$$= \{q_2\}$$ New State

Step 4:

Find the transition for {q_1,q_2}

$$d_D(\{q\ q_1,{}_2\},0) = -e\ closure(d_N(q_1,0) \cup d_N(q_2,0))$$

$$= -e\ closure(\,)j$$

$$= j$$ Dead State d_D

$$(\{q\ q_1,{}_2\},1) = -e\ closure(d_N(q_1,1) \cup d_N(q_2,1))$$

$$= -e\ closure\ q(\,{}_1)$$

$$= \{q\ q_1,{}_2\}$$

$$d_D(\{q\ r,\ \},2) = -e\ closure(d_N(q_1,2) \cup d_N(q_2,2))$$

$$= -e\ closure\ q(\,{}_2)$$

$$= \{q_2\}$$

Step 5:

Find the transition for $\{q_2\}$

$$d_D(\{q_2\},0) = -e\,closure(d_N(q_2,0))$$
$$= -e\,closure(\ )j$$
$$= j \qquad \text{Dead State } d_D$$
$$(\{q_2\},1) = -e\,closure(d_N(q_2,1))$$
$$= -e\,closure(\ )j$$
$$= j \qquad \text{Dead State } d_D$$
$$(\{q_2\},2) = -e\,closure(d_N(q_2,2))$$
$$= -e\,closure\ q(\ _2)$$
$$= \{q_2\}$$

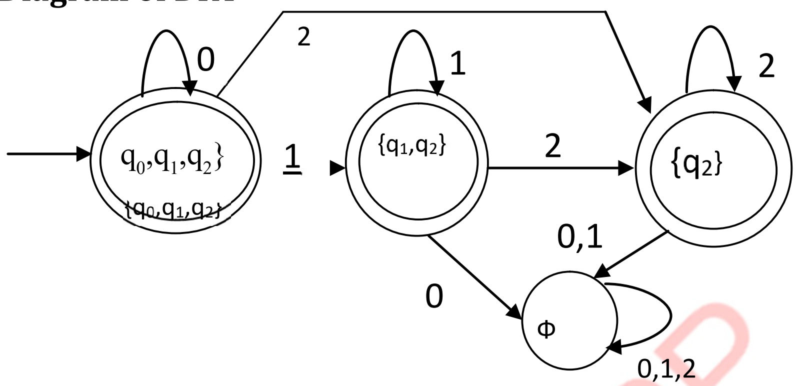
Step 6:

No more new states. Stop the process.

Transition Table of DFA

δ_D	**0**	**1**	**2**
$\rightarrow *\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$*\{q_1, q_2\}$	Φ	$\{q_1, q_2\}$	$\{q_2\}$
$*\{q_2\}$	Φ	Φ	$\{q_2\}$

Transition Diagram of DFA



$Q = \{\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}\}$

$\Sigma = \{0,1,2\}$ $q_0 = \{q_0, q_1, q_2\}$

$F = \{\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}\}$

4. Consider the following ε-NFA.Covertit into DFA

δ_N	**ε**	**a**	**b**
→ p	{r}	{q}	{p,r}
q	Φ	{p}	Φ
*r	{p,q}	{r}	{p}

Step 1:

Compute ε-Closure [states that can be reached by traveling along zero or more ε arrows] for all states.

r ε-Closure (p) = {p,q,r} $\hat{d}(p,e)$

r ε-Closure (q) = {q} $\hat{d}(q,e)$

r ε-Closure (r) = {p,q,r} $\hat{d}(r,e)$

Step 2:

Start with ε-closure (p)= { p, q, r}

Where, p is the starting state of given ε –NFA

Step 3:

Find the transition for { p,q,r}

$$d_D(\{p\ q\ r, , \},a) = -e\ closure(d_N(p\ a,)\cup d_N(q\ a,)\cup d_N(r\ a,))$$
$$= -e\ closure\ q(\ \cup\ \cup p\ r)$$
$$= -e\ closure\ p\ q\ r(\ , ,)$$
$$= \{p\ q\ r, , \}$$
$$d_D(\{p\ q\ r, , \},b) = -e\ closure(d_N(p\ b,)\cup d_N(q\ b,)\cup d_N(r\ b,))$$
$$= -e\ closure(\{\{\ , \}p\ r \cup \cup j\ \{\ \}\})p$$
$$= -e\ closure(\{p\ r, \})$$
$$= \{p\ q\ r, , \}$$

Transition Table of DFA

δ_D	a	b
→*{ p,q,r}	{ p,q,r}	{ p,q,r}

REVIEW QUESTIONS

1. Convert the following NFAs to a DFA . a.

	a	b
→p	{p,q}	p
q	r	r
r	s	-
*s	s	s

b.

δ	a	b
→ p	{q,s}	{q}
* q	{r }	{q,r }
r	{s}	{p}
* s	ϕ	{p}

c.

δ	a	b
→ p	{p,q}	{p}
q	{r,s }	{t}
r	{p,r}	{t}
* s	ϕ	ϕ
* t	ϕ	ϕ

2. Consider the following ε- NFA. Compute the ε- Closure of each state and find it's equivalent DFA .

a.

δ	ε	a	b	c
→p	{q,r}	-	{q}	{r}
q	-	{p}	{r}	{p,q}
*r	-	-	-	{r}

b.

δ	ε	a	b	c
→p	ϕ	{p}	{q}	{r}
q	{p}	{q}	{r}	ϕ
*r	{q}	{r}	ϕ	{p}

3. Construct a minimized DFA for the DFA given below. a.

δ	0	1
→ A	B	E
B	C	F
*C	D	H
D	E	H
E	F	I
*F	G	B
G	H	B
H	I	C
*I	A	E

b.

δ	0	1
→ A	B	A
B	A	C
C	D	B
*D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

4. Construct (DFA) an Automata for the following Language

a. D = { w | w has equal number of occurrences of 01 and 10}

b. D ={{ w | w begins with 10 and ends with 10}

c. D ={{ w | w begins with 01 and ends with 01}

d. D ={{ w | w begins with 10 and ends with 00}

D ={{ w | w begins with 10 and ends with 01}

5. Consider the following ε-NFA.

δ	ε	0	1
→ p	{r}	{q}	{p,r}
q	Φ	{p }	Φ
* r	{p,q}	{r}	{p}

a. Compute the ε-closure of each state.

b. List all the possible strings of length 3 or less accepted by the automaton.

c. Convert the automaton to a DFA.

d. Compute $\hat{d}(q_0,0110)$, where q_0 is the start state.

6. Obtain the DFA equivalent to the following ε-NFA.

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

7. Let L be a language accepted by a NFA then show that there exists a DFA that accepts L.

8. Design a NFA that accepts set of all strings that begins with bb and ends with aa. Convert it into DFA.

9. Construct a minimized DFA for the DFA given below.

δ	0	1
→ a	b	c
b	c	d
c	c	d
*d	d	d
*e	e	e
*f	f	e

10. Design a NFA that accepts empty string or string starts and ends with 0. Convert it into DFA.

11. Define NFA. Explain its significance. Convert the given NFA to DFA. Prove that both NFA and DFA accepts the string 0110.