NPR College of Engineering & Technology
NPR Nagar, Natham, Dindigul - 624401, Tamil Nadu, India.
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai.
An ISO 9001:2015 Certified Institution.
Phone No: 04544- 246 500, 246501, 246502.
Website : www.nprcolleges.org, www.nprcet.org, Email:nprcetprincipal@nprcolleges.org

CS8501

THEORY OF COMPUTATION

by,
C.Kalpana,
Assistant Professor,
Department of Computer Science and Engineering,
NPR College of Engineering & Technology

What is Computation?

Why we need Computation?

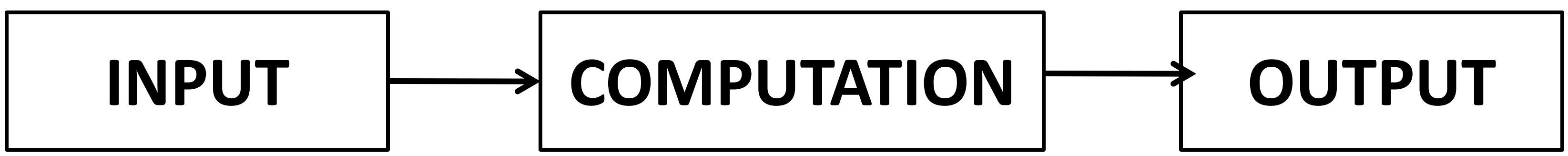
- Calculation.
- To Solve a problem.
- To get a desired result.

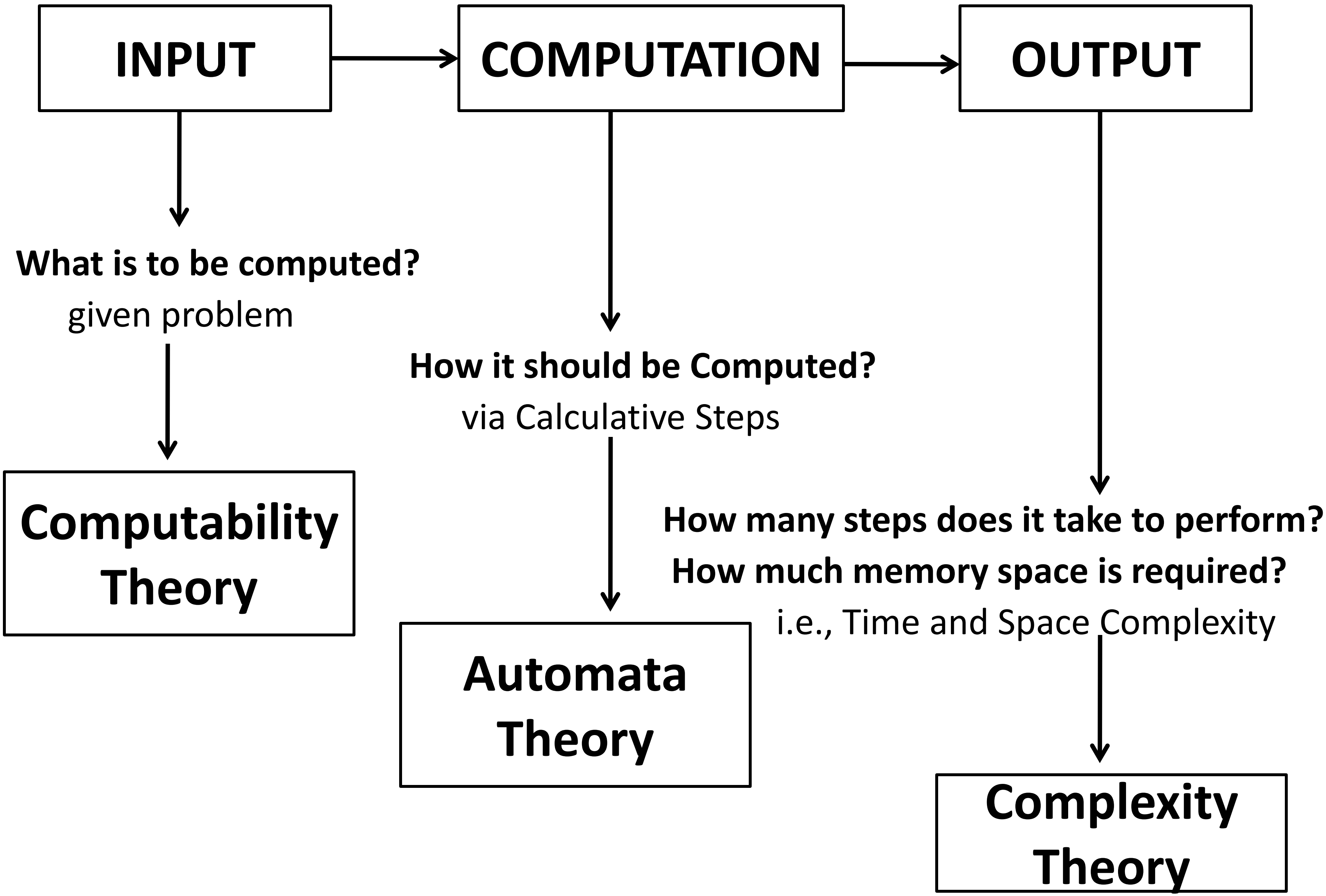
Sequence of steps

↓

Ex:- Algorithm

- Process of Computation




INPUT
COMPUTATION
OUTPUT
What is to be computed?
given problem
Computability Theory
How it should be Computed?
via Calculative Steps
Automata Theory
How many steps does it take to perform?
How much memory space is required?
i.e., Time and Space Complexity
Complexity Theory

- Computability Theory
- Automata Theory
- Complexity Theory

} Theory of Computation

Syllabus

UNIT I AUTOMATA FUNDAMENTALS

Introduction to formal proof – Additional forms of Proof – Inductive Proofs –Finite Automata – Deterministic Finite Automata – Non-deterministic Finite Automata – Finite Automata with Epsilon Transitions

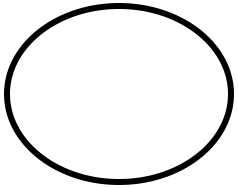
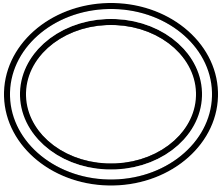
Definition of TOC

It is the branch that deals with "How efficiently the problems can be solved on a Model of computation using an Algorithm."

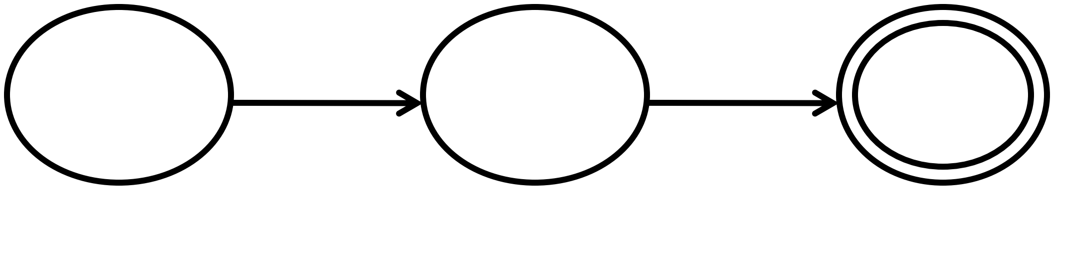
Definition of Automata

- Automaton or Automation
- Doing something by itself.
- It allows us to understand “How machines solve problems?

Automata

- Set of states and Transitions.
- States - Circles 
- Transitions - Arrows 
- End of State - Double Circles 

Automata



Basic Terminologies

- Symbols
- Alphabet
- String
- Language

Basic Terminologies

- **Symbols –** alphabet , letters

  Ex:- a,b,c,0,1…………

- **Alphabet (Σ) –** Set of symbols, which are always finite.

  Ex:- $\Sigma = \{0,1.....9\}$

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

  $\Sigma = \{ A,B,C.......\}$

Basic Terminologies

- **Strings(w)** – Finite sequence of strings from some alphabet.

  Length of string -|w|

  – Number of symbols in the string.

  Ex:- a,b,c,0,1…………

Basic Terminologies

- **Language(L) –** collection of appropriate string.
- Finite or Infinite.
- **Finite**

Ex:**-**

L1= { set of strings of length 2}

Answer: L1= {aa,bb,00,11}

- **Infinite**

Ex:**-**

L2= { set of all strings starts with a }

Answer : L2 = { aa,ab,ac,aab,aac,abc……}

Finite Automata (FA)

- Finite Automata consists of a finite set of states and a set of transitions from one state to another state.
- It occurs on input symbol chosen from an alphabet (Σ).

Language of Finite Automata

- 5 Tuples

$M = (Q, \Sigma , \delta, q0, F)$

Where ,

Q – Set of States.

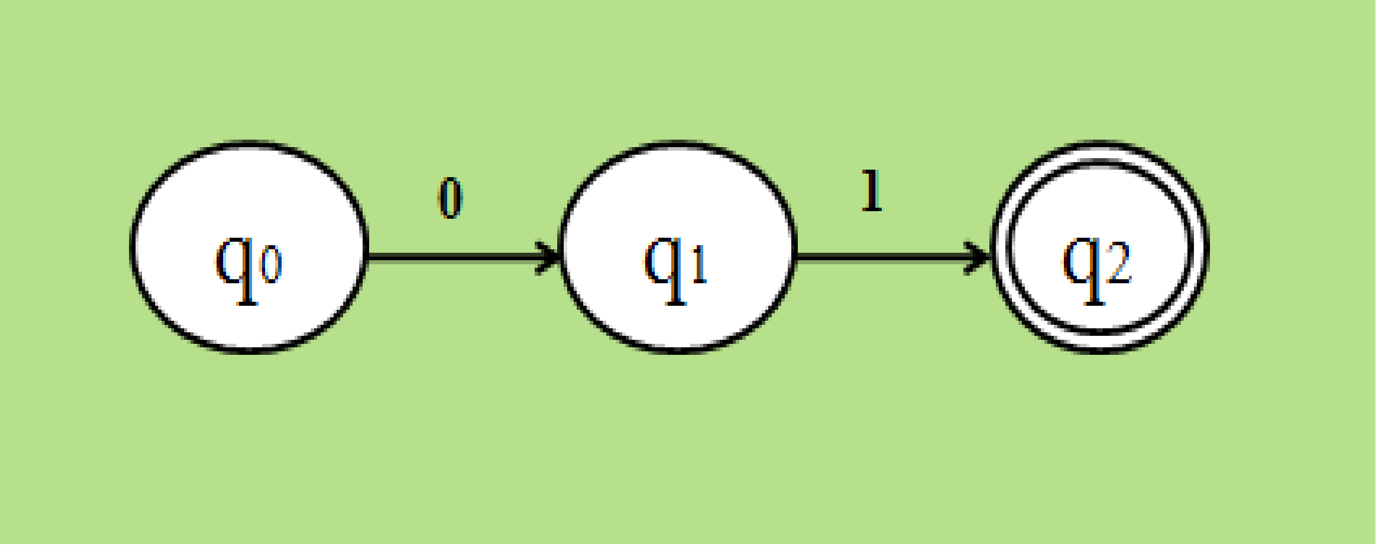
Σ - Input alphabets

δ - Transition Function

q0 - Start State

F- Final State

Finite Automata



Finite Automata

Two Types

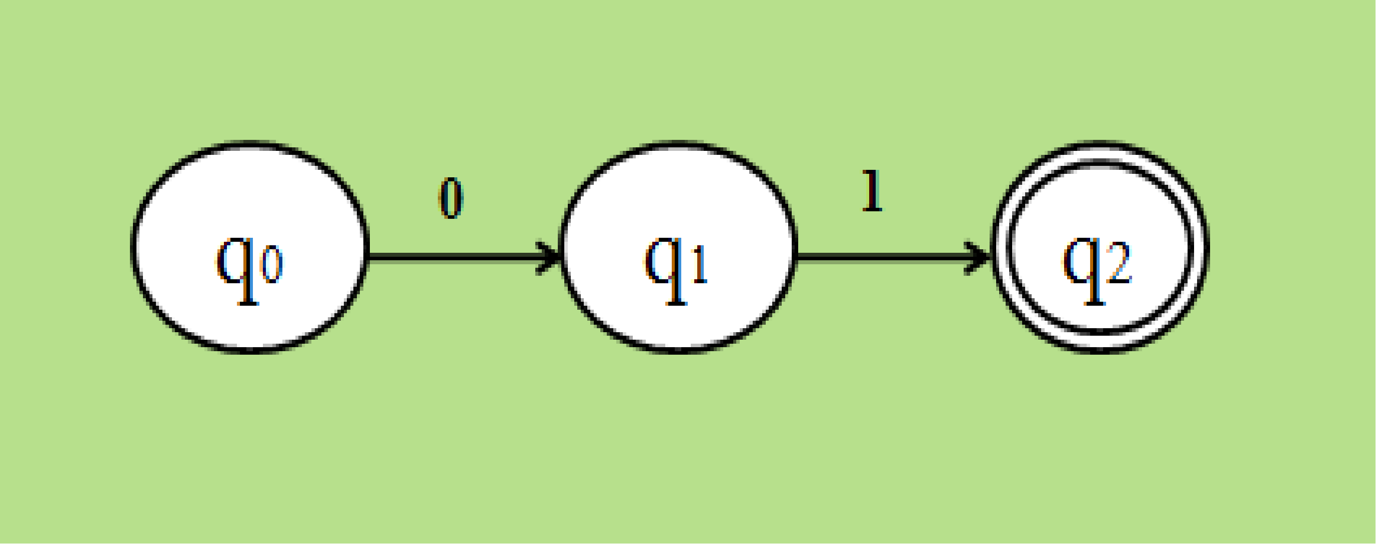
1.DFA

Deterministic Finite Automata

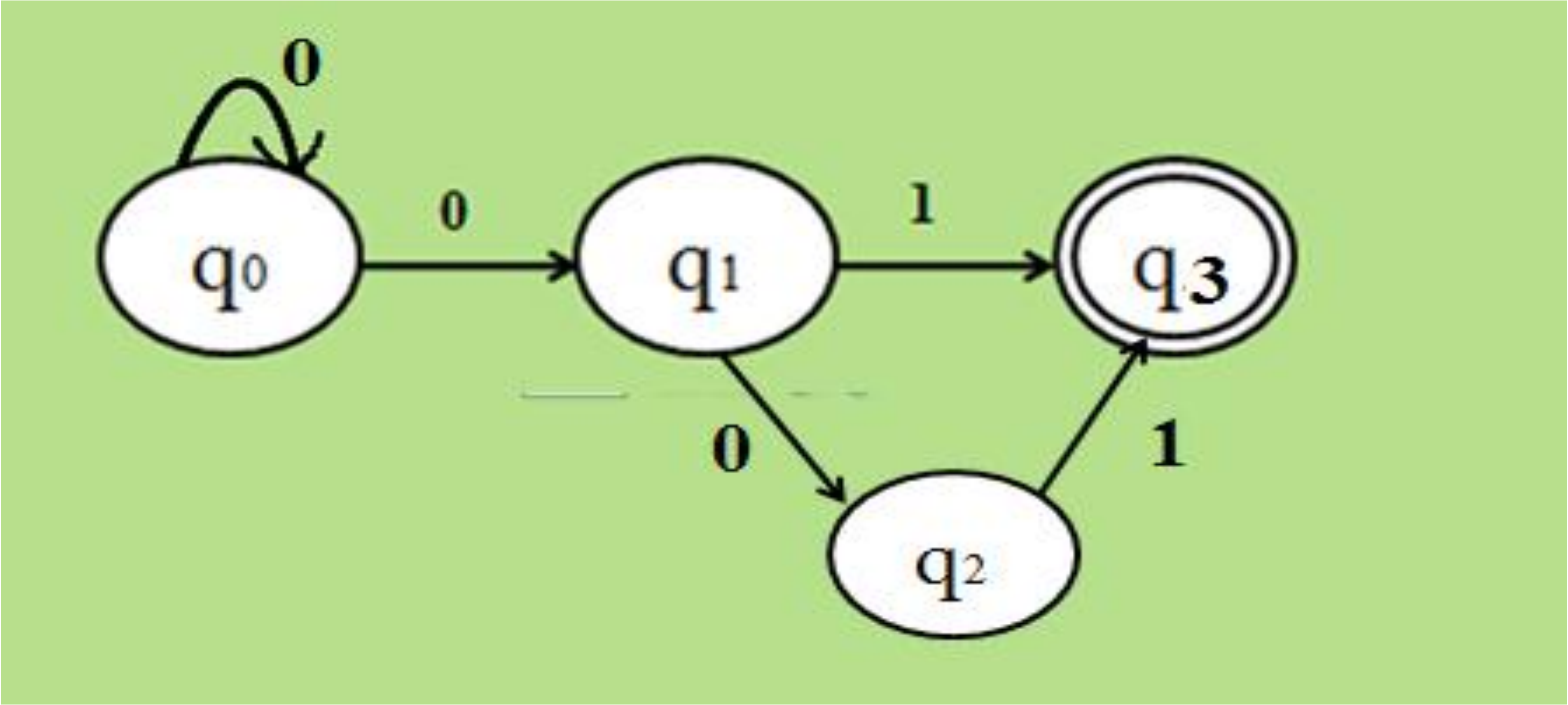
2.NFA

Non – Deterministic Finite Automata

DFA



NFA



Basic Terminologies

- Symbols
- Alphabet
- String
- Language

Basic Terminologies

- **Symbols –** alphabet , Numbers

  Ex:- a,b,c,0,1…………

- **Alphabet (Σ) –** Set of symbols, which are always finite.

  Ex:- $\Sigma = \{0,1.....9\}$

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

  $\Sigma = \{ A,B,C.......\}$

Basic Terminologies

- **Strings(w)** – Finite sequence of strings from some alphabet.

  Length of string -|w|

  – Number of symbols in the string.

  Ex:- a,b,c,0,1…………

Basic Terminologies

- **Language(L) –** collection of appropriate string.
- Finite or Infinite.
- **Finite**

  Ex:**-**

  L1= { set of strings of length 2}

  Answer: L1= {aa,bb,00,11}
- **Infinite**

  Ex:**-**

  L2= { set of all strings starts with a }

  Answer : L2 = { aa,ab,ac,aab,aac,abc……}

Finite Automata (FA)

- Finite Automata consists of a finite set of states and a set of transitions from one state to another state.
- It occurs on input symbol chosen from an alphabet (Σ).

Language of Finite Automata

- 5 Tuples

$M = (Q, \Sigma , \delta, q0, F)$

Where ,

Q – Set of States.

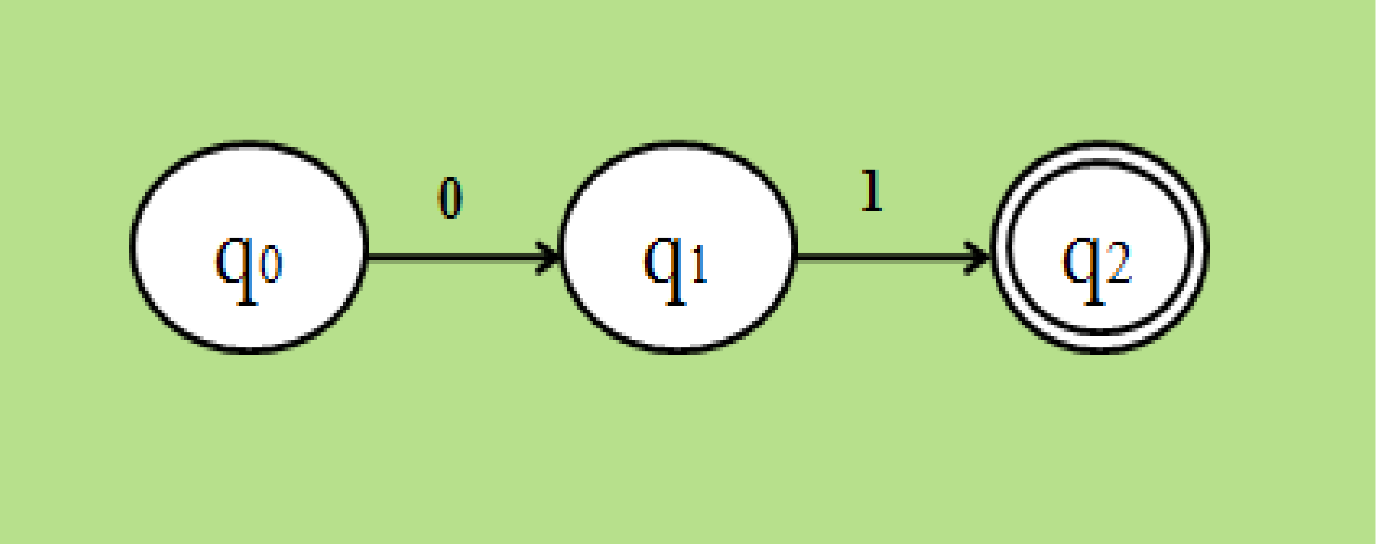
Σ - Input alphabets

δ - Transition Function

q0 - Start State

F- Final State

Finite Automata



Finite Automata

Two Types

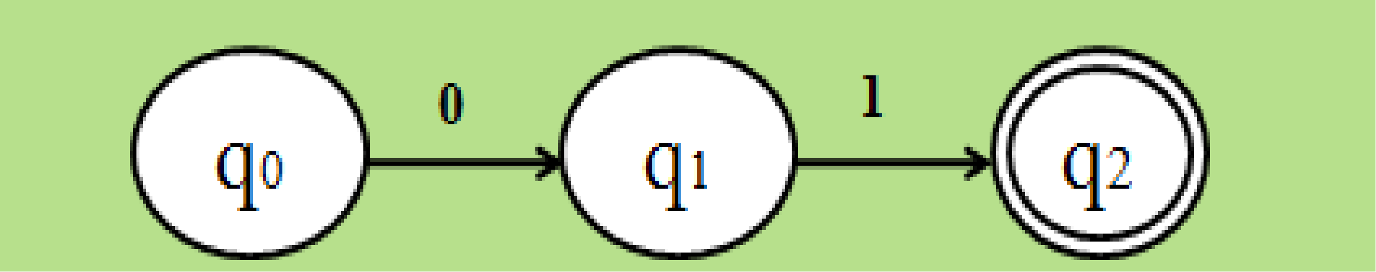
1.DFA

Deterministic Finite Automata

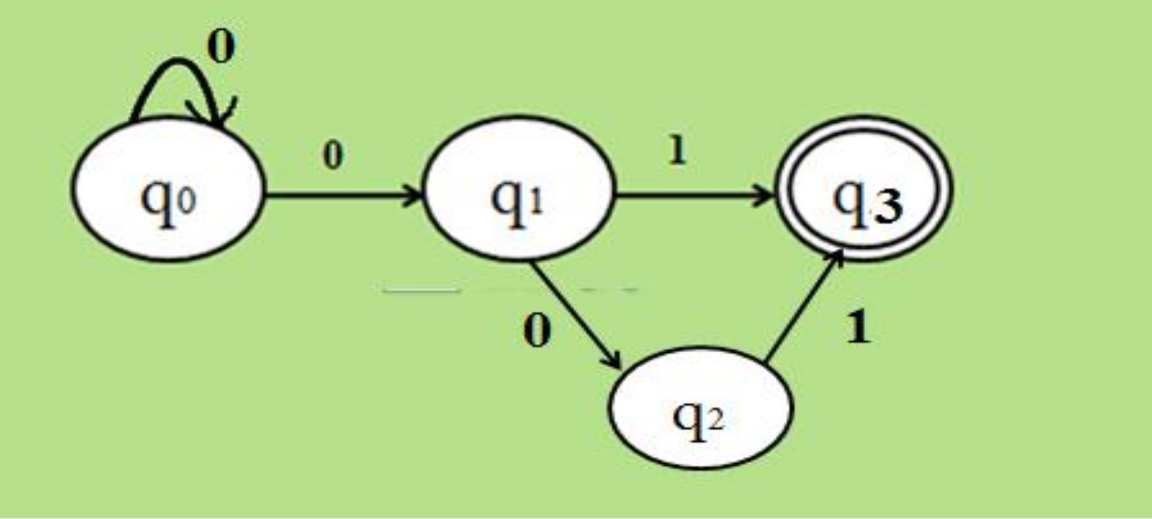
2.NFA

Non – Deterministic Finite Automata

DFA



NFA



Introduction to Formal Proofs

Two Types

1. Deductive Proof

2. Inductive Proof

Introduction to Formal Proofs

1.Deductive Proof

- Sequence of Statements given in logical order.
- To Prove Initial & Next Statement is True
- If H is True the C is also True
- H – Hypothesis
- C - Conclusion

Introduction to Formal Proofs

2.Inductive Proof

- Sequence of Parameterized Statements that uses the statement by itself.
- If A is True the B is also True
- A – Hypothesis
- B - Conclusion

Introduction to Formal Proofs

1.Inductive Proof

- Sequence of Parameterized Statements that uses the statement by itself.
- If A is True the B is also True
- A – Hypothesis
- B - Conclusion

Three Steps in Inductive Proof

- Basis → n = 0 or 1
- Inductive Hypothesis → n = k
- Inductive Step → n=k+1

Problem 1

1. Use mathematical induction to prove that
1 + 2 + 3 + ... + n = n (n + 1) / 2 for all positive integers n.

Solution

- Let the statement P (n) be

  1 + 2 + 3 + ... + n = n (n + 1) / 2

- **STEP 1:** We first show that p (1) is true.
- **Left Side =** 1
- **Right Side =** 1 (1 + 1) / 2 = 1
- Both sides of the statement are equal hence p (1) is true.

- **STEP 2:**
- We now assume that p (k) is true

  $1 + 2 + 3 + ... + k = k (k + 1) / 2$

  and show that p (k + 1) is true

- **<u>STEP 3</u> 1 + 2 + 3 + ... + k = k (k + 1) / 2**
- by adding k + 1 to both sides of the above statement
- 1 + 2 + 3 + ... + k + (k + 1) = k (k + 1) / 2 + (k + 1)

  = (k + 1)(k / 2 + 1)

  = (k + 1)(k + 2) / 2

  The last statement may be written as

  1 + 2 + 3 + ... + k + (k + 1) = (k + 1)(k + 2) / 2

  Which is the statement p(k + 1).

Additional Forms of Proofs

- Proofs by Sets
- Proofs by Contradiction
- Proofs by Counter Example

Finite Automata

- Finite Automata consists of set of states and transitions from one state to another state.
- It occurs from an input symbol chosen from alphabet Σ.

Formal Definition of FA
or
Language of FA

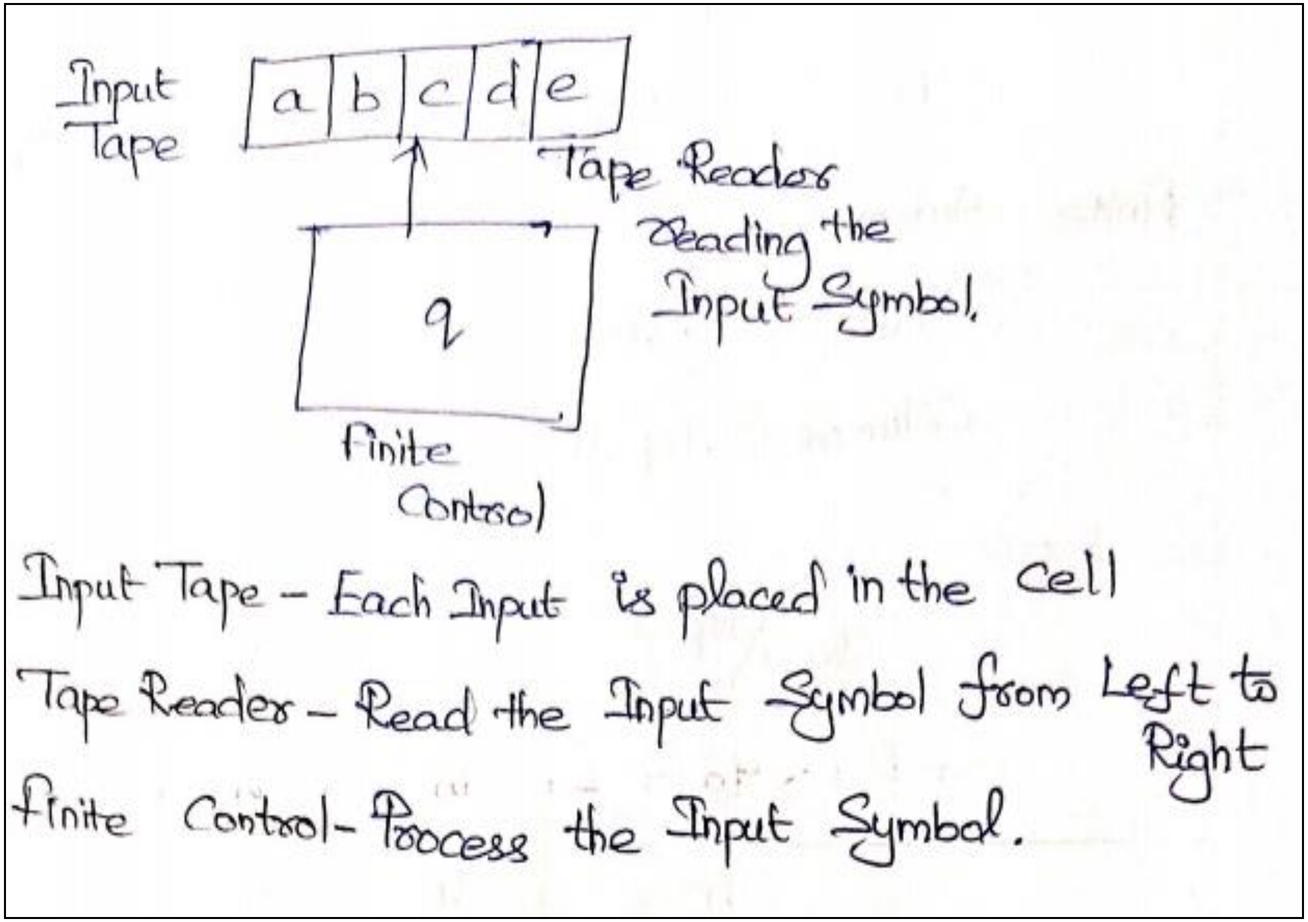
It has 5 Tuples

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

Where,

- Q - Set of States
- Σ - Input Alphabets
- δ - Transition function
- q_0 - Starting State
- F - Final State.

Finite Automata Model

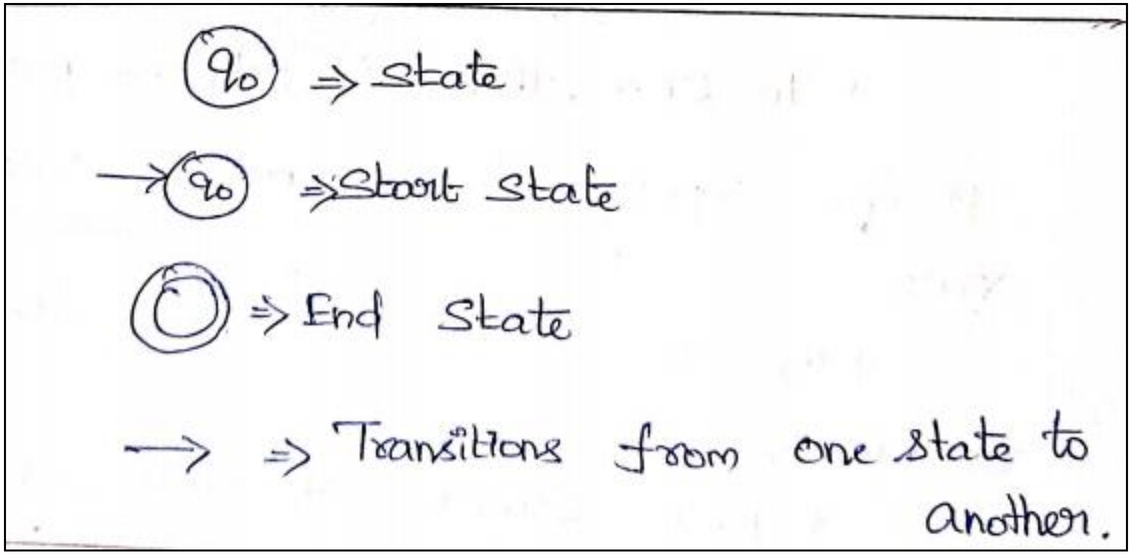


Representation of FA

1. Transition Diagram
2. Transition Table

Representation of FA

1. Transition Diagram
   - Graph
   - Finite set of states, Finite set of symbols and final states



Representation of FA

2. Transition Table

- Tabular representation

– Rows – States & Columns - Inputs

Types of Finite Automata

1. Deterministic Finite Automata (DFA)
2. Non-Deterministic Finite Automata(NFA)

DFA

- In DFA, for each input symbol, one can determine the state to which the machine will move.
- **Deterministic Finite Machine** (or)
- **Deterministic Finite Automaton.**

Formal Definition of DFA

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

Where,

Q - Set of States

Σ - Input Symbols

δ - Transition function

q_0 - Start State

F - final State

Operations of DFA

- Initial State
- Input Process
- For each move, it accepts only one input symbol.
- If the input reaches the final state, then the given input will be accepted or rejected

Transition Function

TRANSITION FUNCTION

$$\delta(q, wa) = \delta(\delta(q, a), w)$$

Where w = Given string

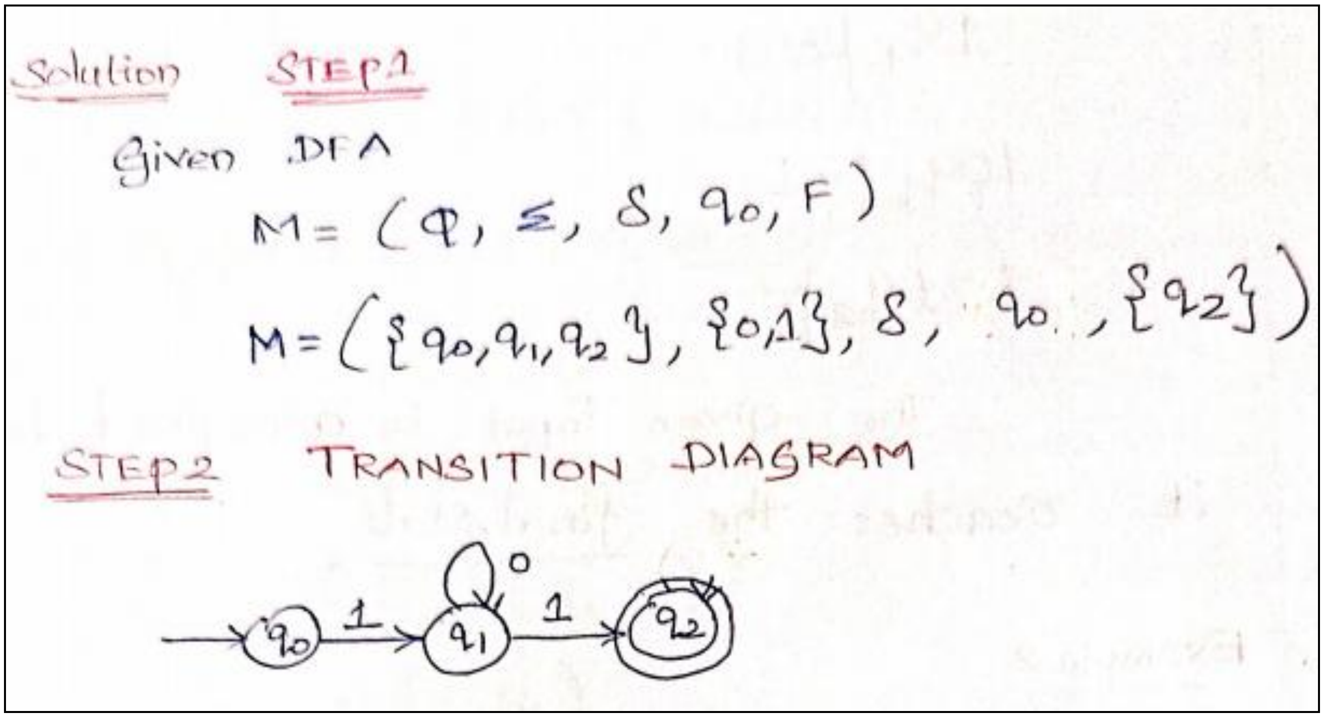
a - first Input Symbol.

DFA Example

1. **Design a DFA that accepts only the input 10 over the set {0,1}.**

DFA Example

2. Design a DFA that accepts only the input 101 over the set {0,1}.



STEP3 TRANSITION TAB

STATE	0	1
$\rightarrow q_0$	ϕ	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_2\}$
$*q_2$	ϕ	ϕ

STEP4 TRANSITION FUNCTION

$$\delta(q, wa) \Rightarrow \delta(q, aw) = \delta(\delta(q,a), w)$$

$w = 101$ (Given String)

$$= \delta(\delta(q_0, 1), 01)$$

$$= \delta(\delta(q_1), 01)$$

$$= \delta(\delta(q_1, 0), 1)$$

$$= \delta(\delta(q_1), 1)$$

$$= \delta(\delta(q_1, 1)) \Rightarrow \delta(q_2) \Rightarrow q_2$$ Accepted

15

METHOD 2 TRANSITION FUNCTION

First write the starting state.

$q_0 \vdash 101$

$1q_1 \vdash 01$

$10q_1 \vdash 1$

$101q_2 \vdash$

$\therefore$ The given input is accepted because it reaches the final state.

Equivalence of DFA & NFA

EXAMPLE PROBLEMS.

1. Construct DFA equivalent to the NFA

$M = (\{p, q, r\}, \{0, 1\}, \delta, p, \{q, s\})$

Where δ is defined in the following table.

δ	0	1
p	{q, s}	{q}
q	{r}	{q, r}
r	{s}	{p}
s	–	{p}

Solution

STEP1

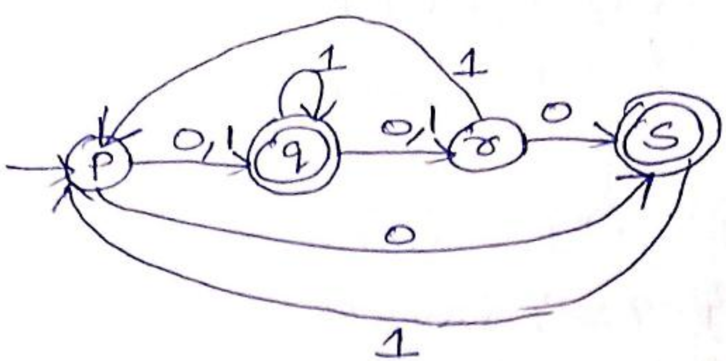
Let $M = (Q, \Sigma, \delta, q_0, F)$

$M = (\{p, q, r\}, \{0, 1\}, \delta, p, \{q, s\})$

Starting State = p

Final State = $\{q, s\}$

STEP2 TRANSITION DIAGRAM



STEP 3

Construct Subsets, **Formula = 2^n**

n = Total number of States = 4

$2^n = 2^4 = 2 \times 2 \times 2 \times 2 = 16$ States

$Q = (\{\phi, \{p\}, \{q\}, \{r\}, \{s\}, \{p,q\}, \{p,r\}, \{p,s\},$
$\{r,s\}, \{q,r\}, \{q,s\}, \{p,q,r\}, \{p,q,s\}, \{p,r,s\},$
$\{q,r,s\}, \{p,q,r,s\})$

STEP 4 Transition function for Subset Construction.

δ	0	1
φ	φ	φ
→ p	{q,s}	{q}
* (q)	{r}	{q,r}
r	{s}	{p}
* (s)	–	{p}
* ({p,q})	{q,s,r}	{q,r}
{p,r}	{q,s}	{p,q}

	δ	0	1
	φ	φ	φ
→	p	{q,s}	{q}
*	(q)	{r}	{q,r}
	r	{s}	{p}
*	(s)	–	{p}

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

δ	0	1
φ	φ	φ
→ p	{q,s}	{q}
* (q)	{r}	{q,r}
r	{s}	{p}
* (s)	-	{p}

STEP 5 To Construct DFA,

find transition function.

Here, the states will get minimized.

Start with starting state.

i) $\delta(p,0) = \{q,s\}$ → New state generated.

Note ⇒ [See the question it has only 4 states (p,q,r,s}. Here {q,s} is new state].

ii) $\delta(p,1) = \{q\}$

iii) $\delta(q,0) = \{r\}$

iv) $\delta(q,1) = \{q,r\}$ - New State

v) $\delta\{r,0\} = \{s\}$

vi) $\delta(r,1) = \{p\}$

vii) $\delta(s,0) = -$ (or) ϕ

viii) $\delta(s,1) = \{p\}$

(x) Find transitions for New state.

ix) $\delta(\{q,s\},0) = \{r\}$

x) $\delta(\{q,s\},1) = \{p,q,r\}$ - New State

New States $\{q,s\}$, $\{q,r\}$ till now.

xi) $\delta(\{q,r\},0) = \{r,s\}$ - New state

xii) $\delta(\{q,r\},1) = \{p,q,r\}$ - New State

xiii) $\delta(\{p,q,r\},0) = \{q,r,s\}$ - New State

xiv) $\delta(\{p,q,r\},1) = \{p,q,r\}$

xv) $\delta(\{r,s\},0) = \{s\}$

xvi) $\delta(\{r,s\},1) = \{p\}$

xvii) $\delta(\{q,r,s\},0) = \{r,s\}$

xviii) $\delta(\{q,r,s\},1) = \{p,q,r\}$

∴ We find transition function for all the New States.

Next Step ⇒ Draw transition table.

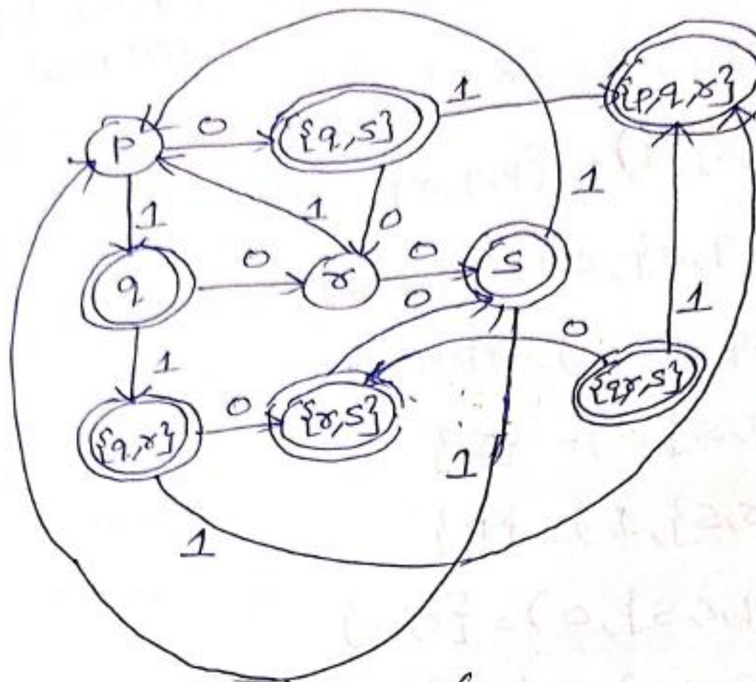
δ	0	1
ϕ	ϕ	ϕ
→ p	{q,s}	{q}
* (q)	{r}	{q,r}
r	{s}	{p}
* (s)	-	{p}

STEP 6 DFA TRANSITION TABLE

δ	0	1
→p	{q,s}	{q}
*(q)	{r}	{q,r}
r	{s}	{p}
*(s)	-	{p}
*({q,s})	{r}	{p,q,r}
*({q,r})	{r,s}	{p,q,r}
*({p,q,r})	{q,r,s}	{p,q,r}

δ	0	1
*{r,s}	{s}	{p}
*({q,r,s})	{r,s}	{p,q,r}

STEP 7 DFA TRANSITION DIAGRAM



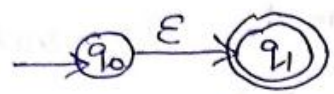
∴ Final States are ({q}, {s}, {q,r}, {q,s}, {r,s}, {q,r,s}, {p,q,r}).

FINITE AUTOMATA WITH EPSILON TRANSITIONS

ε - Epsilon

* The ε-transition in NFA, which helps to move from one state to another state without having an input symbol from the set Σ.

Ex:



* Here, the ε (epsilon) moves from the state q_0 to q_1.

* In ε-NFA, we can simply change the states from one state to another.

DEFINITION OF NFA with ε

* The language L accepted by NFA with ε, denoted by $M = (Q, \Sigma, \delta, q_0, F)$ can be defined as,

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a NFA with ε.

Where,

Q - Finite Set of States.

Σ - Input Symbols

δ - Transition function $Q \times \{\Sigma \cup \varepsilon\}$ to 2^Q

q_0 - Start State

F - Final State

It has 5 Steps,

1. Formal Notation
2. Epsilon Closures
3. Extended Transition for epsilon (ε-NFA).
4. Eliminating ε- Transition.
5. Language of ε-NFA.

1. FORMAL NOTATION:

Let ε-NFA can be represented as,

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

Where δ has the arguments,

i) A state in Q

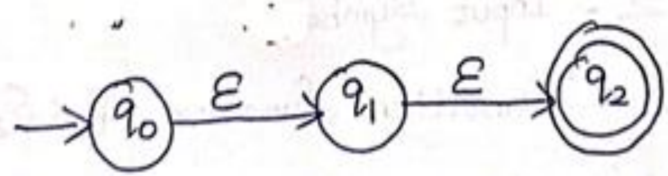
ii) A New Input Symbol $\underline{\underline{\varepsilon}}$ is added to the Set Σ.

2. EPSILON CLOSURE (ε-CLOSURE)

* Epsilon (ε) means present state can goto other state without any input.

* Epsilon closure is finding all the states which can be reached from the present state on one or more epsilon transitions.

Ex:



ε-closure$(q_0) = \{q_0, q_1, q_2\}$ → ε-moves

ε-closure$(q_1) = \{q_1, q_2\}$

ε-closure$(q_2) = \{q_2\}$

3. EXTENDED TRANSITION FUNCTION for ε-NFA.

* Let $E = (Q, \Sigma, \delta, q_0, F)$ be the ε-NFA, then $\hat{\delta}$ is the extended transition function.

* The extended transition function $\hat{\delta}$ is defined as,

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

ii) for ω in Σ^* and a in Σ,

$$\hat{\delta}(q, \omega a) = \varepsilon\text{-closure}(P)$$

Where,

$$P = \{ p \mid \text{for some } r \text{ in } \hat{\delta}(q, \omega),\ p \text{ is in } \delta(r, a) \}$$

iii) $\delta(R, a) = \bigcup_{q \text{ in } R} \delta(q, a)$

iv) $\delta(R, \omega) = \bigcup_{q \text{ in } R} \hat{\delta}(q, \omega)$.

4. ELIMINATING $\mathcal{E}$-TRANSITIONS.

* It is possible to construct DFA D that accepts the same language as $\mathcal{E}$-NFA.

Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$.

* Then the equivalent DFA $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ is constructed as follows.

1. Q_D is the subset of Q_E.
2. $q_D = \mathcal{E}\text{-closure}(q_0)$ is the start state of D.
3. F_D is the final state that contain atleast one accepting state of E.

i.e., $F_D = \{S \mid S \text{ in } Q_D \text{ and } S \cap F \neq \Phi\}$

4. $\delta_D(S, a)$ is computed by,

i) Let $S = \{P_1, P_2 \dots P_K\}$

ii) Compute $\bigcup_{i=1}^{K} \delta(P_i, a) = \{r_1, r_2 \dots r_m\}$

iii) $\delta_D(S, a) = \bigcup_{j=1}^{M} \varepsilon\text{-closure}(r_j)$

5. LANGUAGE OF ε-NFA.

* The language of an ε-NFA is

$E = (Q, \Sigma, \delta, q_0, F)$ is

$$L(E) = \{w \mid \delta(q_0, w) \cap F \neq \Phi\}$$

i.e., the language of E is the set of strings w that take the start state to atleast one accepting state.

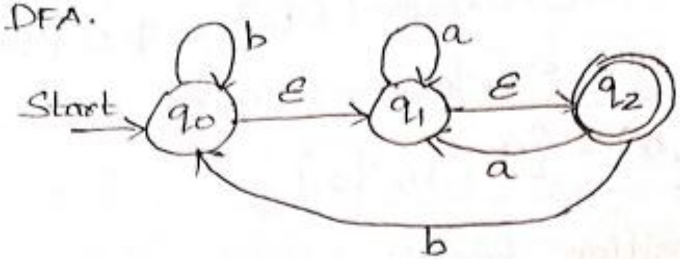
CONVERSION OF NFA with ε to DFA.

PROBLEM STEPS.

1. Find ε-Closure.
2. Find Transition function from ε-closure of Start State.
3. Find Transition function for each and every New State.
4. Draw Transition → Table / Diagram for DFA.
5. Write the final states. (Mention).

EXAMPLE PROBLEMS

1. Convert the following NFA with ε to equivalent DFA.



Solution

STEP 1 Find ε-Closures.

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

ε-Closure $\{q_1\} = \{q_1, q_2\}$

ε-Closure $\{q_2\} = \{q_2\}$

STEP 2

Let us start from ε-Closure of start state

ε-Closure $\{q_0\} = \{q_0, q_1, q_2\}$ - New state

STEP3

Find Transitions for New state to Input a, b

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

Refer diagram

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

$= \varepsilon\text{-closure}(\phi \cup \{q_1\} \cup \{q_1\})$

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

$\boxed{\delta(\{q_0,q_1,q_2\},a) = \{q_1,q_2\}}$ - NewState

ε - closure $\{q_0\} = \{q_0, q_1, q_2\}$

ε - closure $\{q_1\} = \{q_1, q_2\}$

ε - closure $\{q_2\} = \{q_2\}$

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

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

$= \varepsilon\text{-closure}(\{q_0\} \cup \phi \cup \{q_0\})$

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

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

iii) Find transition for New State $\{q_1, q_2\}$

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

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

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

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

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

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

ε - Closure $\{q_1\} = \{q_1, q_2\}$

ε - Closure $\{q_2\} = \{q_2\}$

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

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

$= \varepsilon\text{closure}(\phi \cup \{q_0\})$

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

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

STEP 4 Hence, the generated DFA is

	STATES / INPUT	a	b
*	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
*	$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$

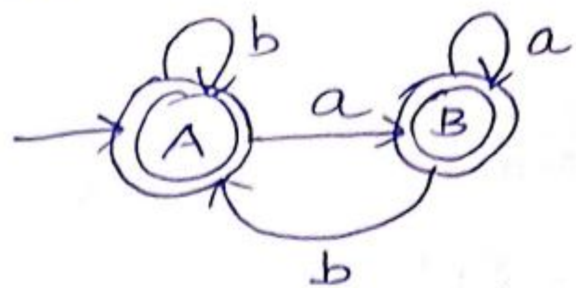
Assume $\{q_0, q_1, q_2\}$ as A

$\{q_1, q_2\}$ as B

	States / Inputs	a	b
⇒ *	A	B	A
*	B	B	A

STEP 5 Draw Transition diagram for DFA.

Mark start state as well as final state



Final states are ({A}, {B}).

Thank You