Basic Definitions and Concepts

1

Alphabet : An alphabet (Σ) is defined as the finite set of input symbols.

eg. $\Sigma = \{0, 1, 2\}$

where, 0, 1 & 2 are symbols or letters

Basic Definitions and Concepts

<u>Alphabet</u> : An alphabet (Σ) is defined as the finite set of input symbols.

eg. $\Sigma = \{0, 1, 2\}$

where, 0, 1 & 2 are <u>symbols</u> or <u>letters</u>

As we say that in English Alphabet, there are 26 letters.

Basic Definitions and Concepts

2

Strings / Sentence / Word : It is defined as the finite sequence of symbols over the given alphabet (Σ).

eg. 0, 1, 2, 01, 02, 12, etc.

over $\Sigma = \{0, 1, 2\}$

Basic Definitions and Concepts

3

String Length: It is defined as the number of symbols present in a given string.

eg: $x = 2120$

then $|x| =$

String Length

Basic Definitions and Concepts

String Length: It is defined as the number of symbols present in a given string.

eg: $x = 2120$

then $|x| = 4$ Irrespective of repeating symbols.

String Length

Basic Definitions and Concepts

4

Empty String: The string of length zero (0), would be called as an empty string, denoted by the symbol ϵ also, called as Epsilon.

Basic Definitions and Concepts

5

Language: It is defined as the set of strings defined over the given alphabet, (Σ)

eg. $L = \{ X \mid X \text{ ends in "ba" over } \Sigma = \{a, b\} \}$

Basic Definitions and Concepts

Language : It is defined as the set of strings defined over the given alphabet, (Σ)

eg. $L = \{ x \mid \underbrace{x \text{ ends in "ba"}}_{\text{Condition}} \text{ over } \underbrace{\Sigma = \{a, b\}}_{\text{Alphabet}} \}$

Language

String

Condition

Alphabet

Basic Definitions and Concepts

⑥

Power of an Alphabet: If Σ is an alphabet, then $(\Sigma)^k$ is defined as the set of all the strings from the alphabet Σ of length k.

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

Basic Definitions and Concepts

Power of an Alphabet: If Σ is an alphabet, then $(\Sigma)^k$ is defined as the set of all the strings from the alphabet Σ of length k.

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

Σ^1 = Set of all strings of length 1 $= \{a, b\}$

Σ^2 = Set of all strings of length 2 $= \{aa, ab, ba, bb\}$

Basic Definitions and Concepts

Power of an Alphabet: If Σ is an alphabet, then $(\Sigma)^k$ is defined as the set of all the strings from the alphabet Σ of length k.

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

$\Sigma^1 =$ Set of all strings of length 1 $= \{a, b\}$

$\Sigma^2 =$ Set of all strings of length 2 $= \{aa, ab, ba, bb\}$

Also, $\Sigma^0 =$ Set of all strings of length 0 $= \{\ \} = \{\epsilon\}$

Basic Definitions and Concepts

7

Substring: Let U & W be two strings defined over the alphabet (Σ), then U is said to be the substring of W, if U occurs <u>as it is</u> in the same order.

eg. $W = \cancel{\{G, A, T, E\}}$ GATE

then $U = \{\epsilon, G, A, T, E, GA, GT, GE, AT, AE, TE, GAT, GTE, GAE, ATE, GATE\}$

For

Basic Definitions and Concepts

Substring : Let U & W be two strings defined over the alphabet (Σ), then U is said to be the substring of W, if U occurs as it is in the same order.

eg. W = { G, A, T, E }

then U = { E, G, A, T, E, GA, ~~GT~~, ~~GE~~, AT, ~~AE~~, TE, GAT, ~~GTE~~, ~~GAE~~, ATE, GATE }

For

Basic Definitions and Concepts

Substring: Let U & W be two strings defined over the alphabet (Σ), then U is said to be the substring of W, if U occurs <u>as it is</u> in the same order.

eg. $W = \{G, A, T, E\}$

then $U = \{\epsilon, G, A, T, E, GA, \cancel{GT}, \cancel{GE}, AT, \cancel{AE}, TE, GAT, \cancel{GTE}, \cancel{GAE}, ATE, GATE\}$

For $K = 0 \rightarrow \epsilon$

$K = 1 \rightarrow G, A, T, E$

$K = 2 \rightarrow GA, GT, GE, AT, AE, TE$

$K = 3 \rightarrow GAT, GTE, GAE, ATE$

$K = 4 \rightarrow GATE$

Basic Definitions and Concepts

Substring:

Practice Problem:

1. $W = \cancel{\{}HELLO\cancel{\}}$, $U = ?$

2. $W = \cancel{\{}ROCKET\cancel{\}}$, $U = ?$

~~X~~ They are Strings, Not Sets

Basic Definitions and Concepts

Substring: Practice Problem (Solution)

① $W = \{HELLO\}$, $U = 16$

② $W = \{ROCKET\}$, $U = 22$

Shortcut: $\boxed{\frac{n(n+1)}{2} + 1}$

Basic Definitions and Concepts

10 **Types of Substring** : There are two types of substring

- Trivial (Improper)

$$\boxed{\frac{n(n+1)}{2} + 1}$$

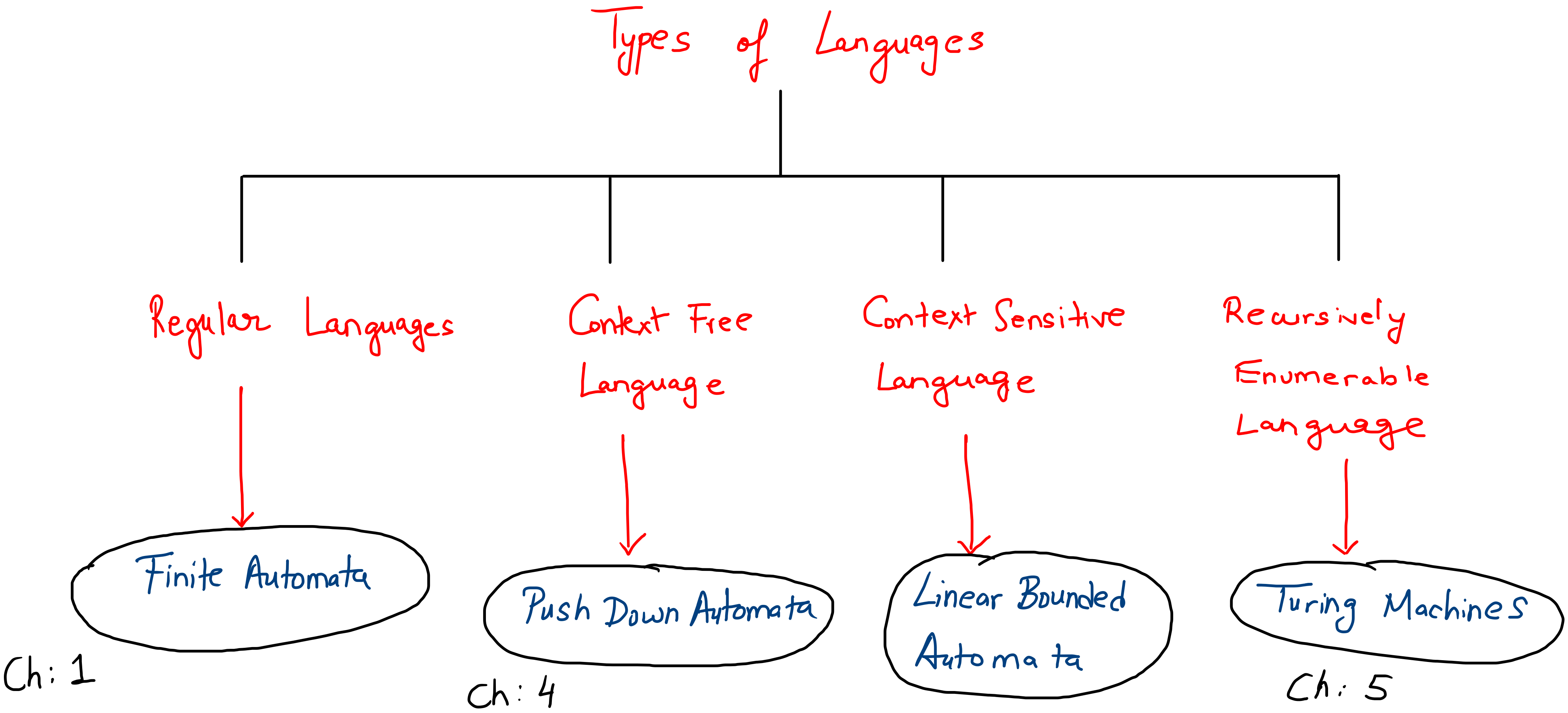
* Consists of ϵ & the complete string

- Non-Trivial (Proper)

$$\boxed{\frac{n(n+1)}{2} - 1}$$

* Does not contain ϵ & the complete string.

Finite State Machines and Finite Automata



Finite State Machines and Finite Automata

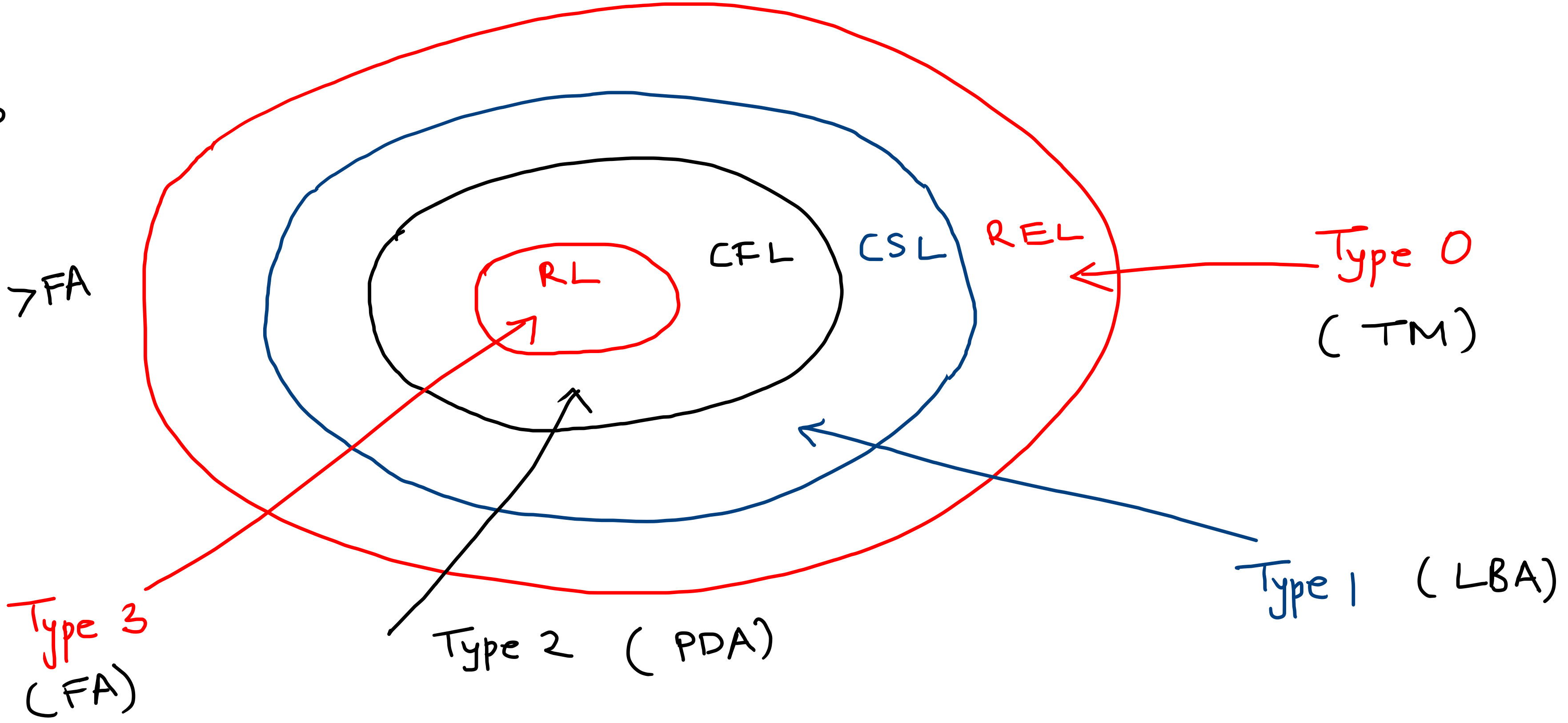
Chomsky's Hierarchy

Power :

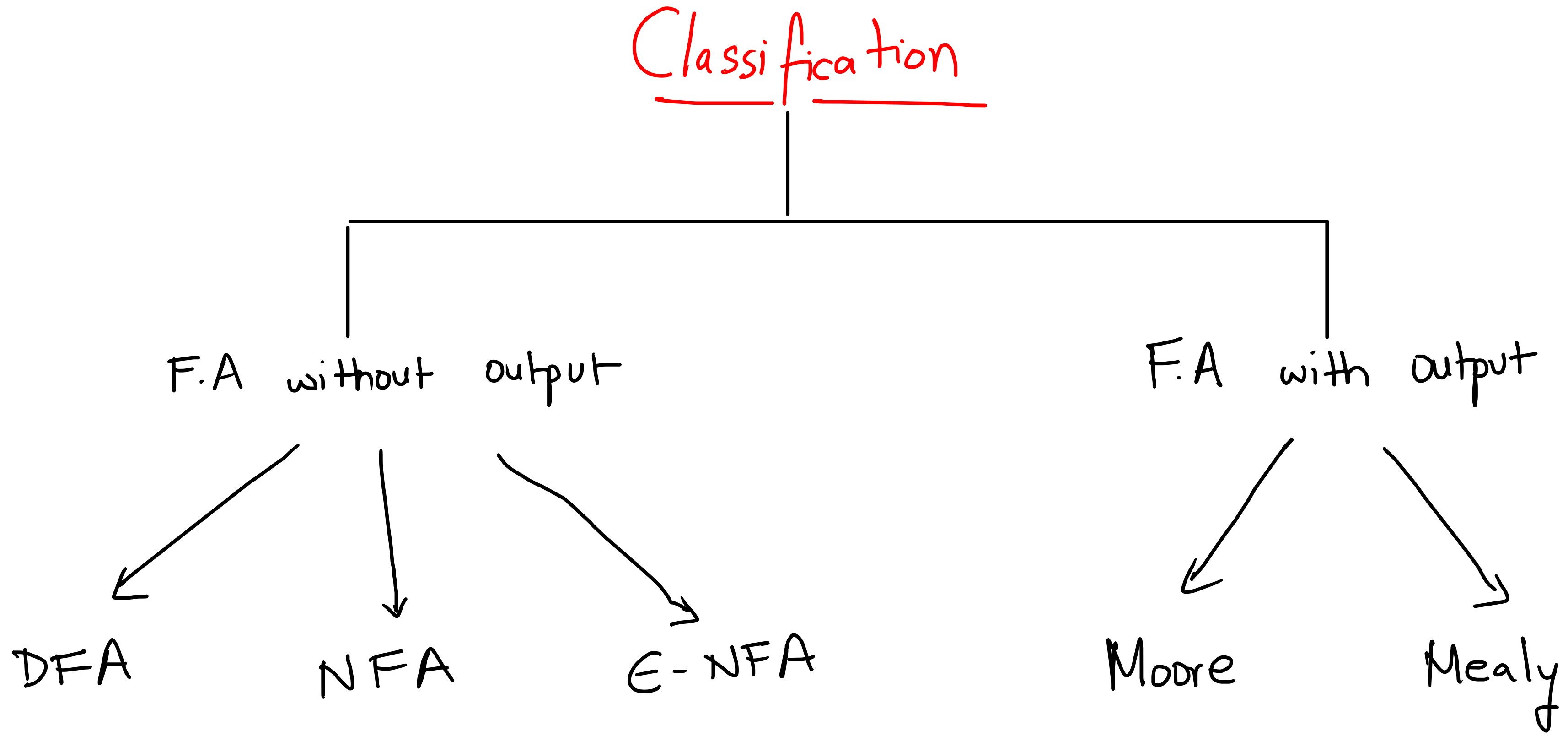
$T_0 > T_1 > T_2 > T_3$

$\therefore$

TM > LBA > PDA > FA



Finite State Machines and Finite Automata



Finite State Machines and Finite Automata

$$I_{nput} \longrightarrow \boxed{\text{FSM}} \longrightarrow O_{utput}$$

$$\Sigma \longrightarrow \boxed{\text{FA}} \longrightarrow F$$

3 Tuples

$I \longrightarrow$ Input

$O \longrightarrow$ Output

$S \longrightarrow$ States

5 Tuples

$\Sigma \longrightarrow$ Inputs

$Q \longrightarrow$ States

$q_0 \longrightarrow$ Initial State

$F \longrightarrow$ Final state

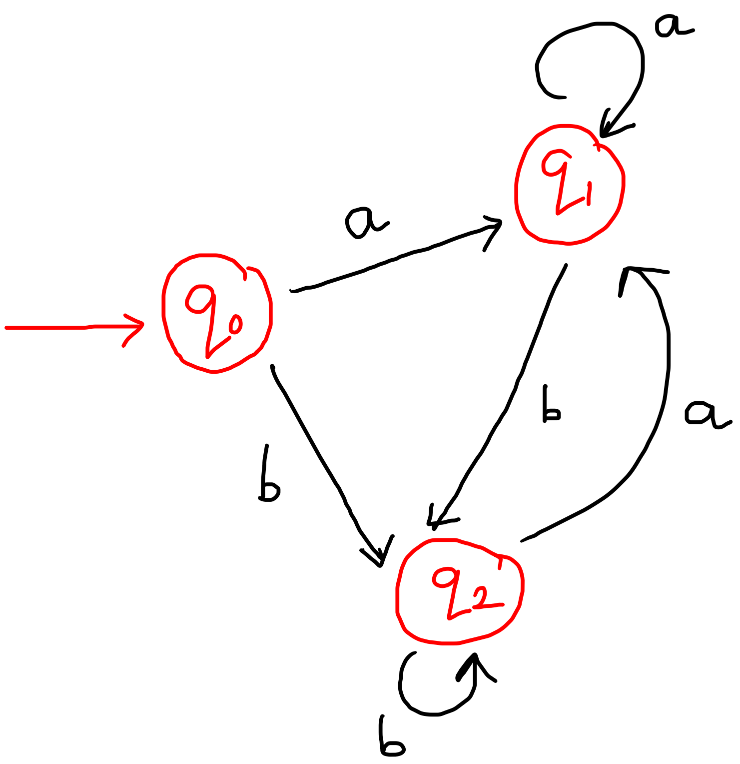
$\delta \longrightarrow$ Transitions

Finite State Machines and Finite Automata

Deterministic Finite Automata (DFA)

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

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



☆ All the states have definite transitions for a given input.

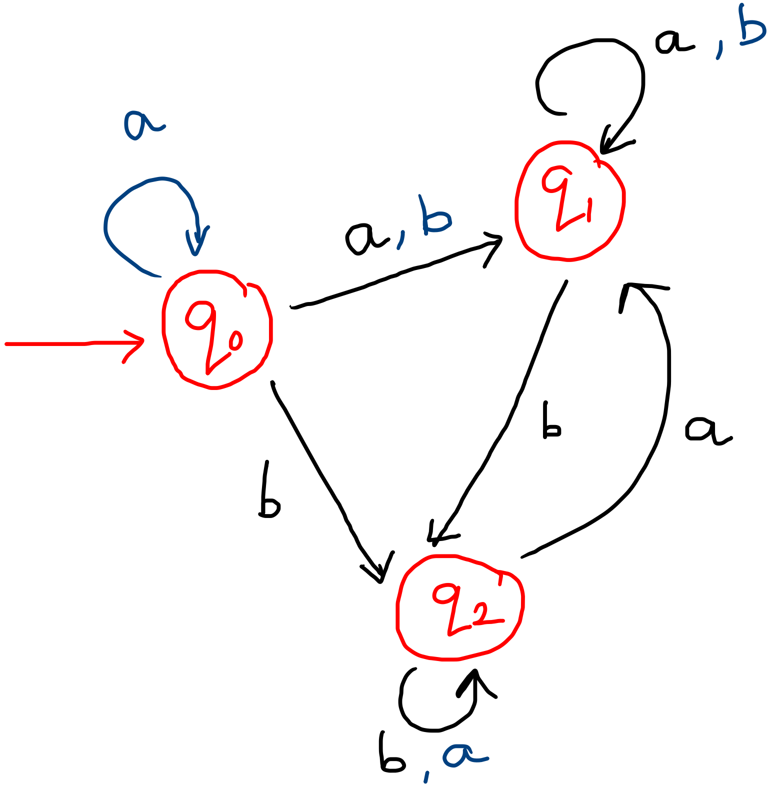
☆ There is no ambiguity

Finite State Machines and Finite Automata

Non Deterministic Finite Automata (NFA)

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

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



★ The states may or may not have define transitions for a given input.

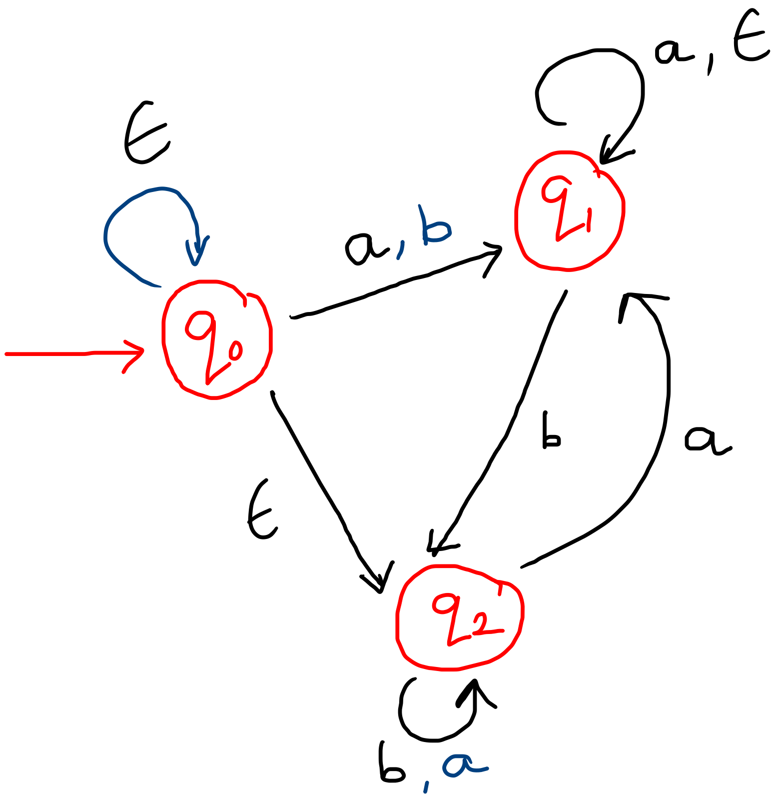
★ There is ambiguity

Finite State Machines and Finite Automata

NFA with ϵ-Moves (ϵ-NFA)

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

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



- ☆ The states may or may not have define transitions for a given input.
- ☆ There is ambiguity
- ☆ The states may have ϵ-transitions or null-transitions

Finite State Machines and Finite Automata

Problem No : 1

Q.1. Design a F.S.M to check whether a given decimal number is divisible by 3.

Steps :

- Step 1: Theory
- Step 2: Logic
- Step 3: Implementation
- Step 4: Transition Diagram
- Step 5: Simulation / Example

Finite State Machines and Finite Automata

Sol: Step 1: Theory

FSM consists of finite set of states (S), which on receiving inputs (I), alters them to produce output (O).

FSM is defined by 2 functions :-

1) State Transition Function (STF) : $S \times I \rightarrow S$

2) Machine Application Function (MAF) : $S \times I \rightarrow O$

Finite State Machines and Finite Automata

Step 2: Logic

$$I = \{0, 1, 2, \ldots\ldots 9\}$$

$$O = \{Y, N\}$$

$$S = \{q_s, q_0, q_1, q_2\}$$

Finite State Machines and Finite Automata

Step 3: Implementation

S \ I				

Finite State Machines and Finite Automata

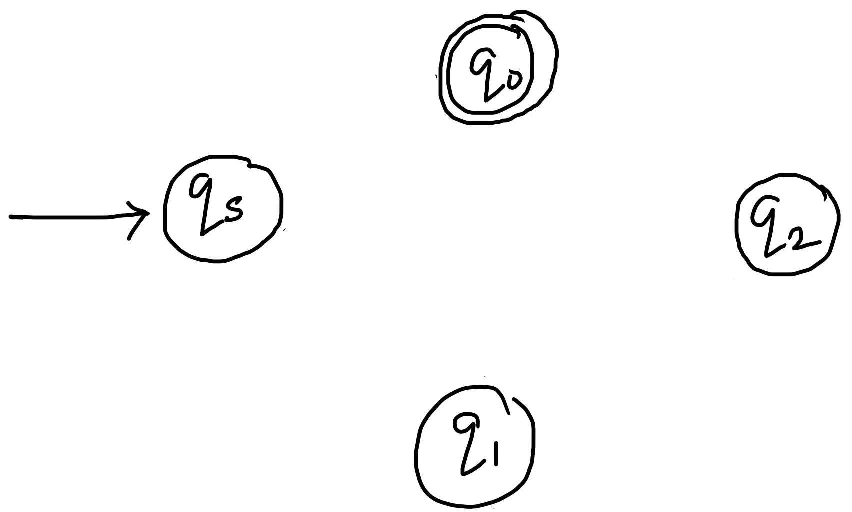
Step 3: Implementation

0, 3, 6, 9 → 0
1, 4, 7 → 1
2, 5, 8 → 2

S \ I	0 {0,3,6,9}	1 {1,4,7}	2 {2,5,8}	
→ q_s	q_0	q_1	q_2	
q_0^*	q_0	q_1	q_2	
q_1	q_1	q_2	q_0	
q_2	q_2	q_0	q_1	

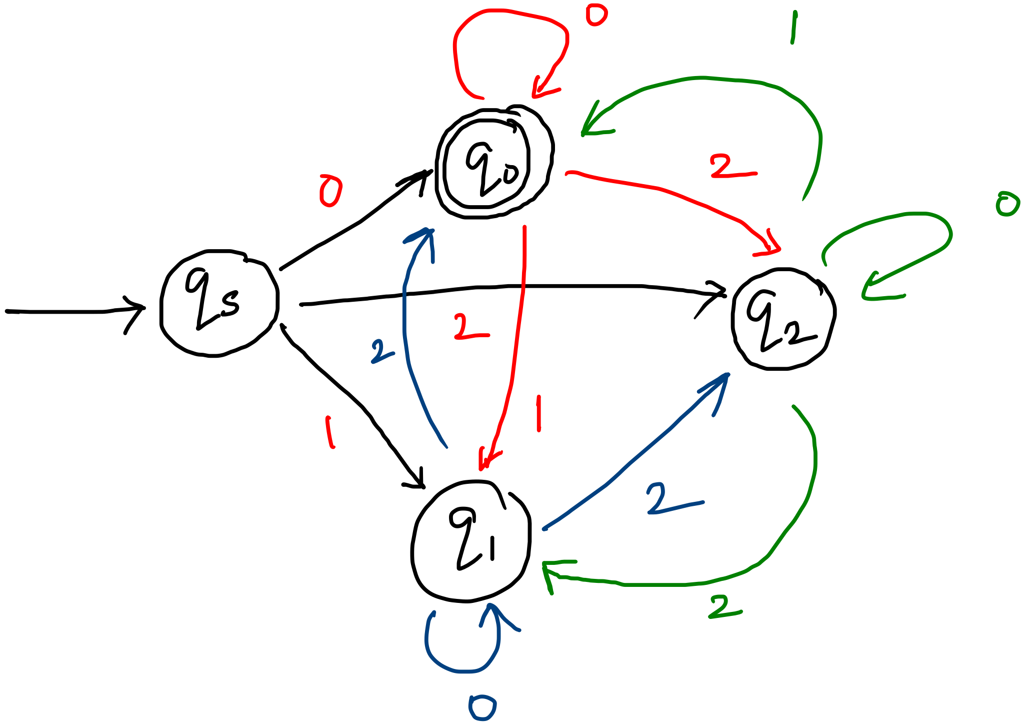
Finite State Machines and Finite Automata

Step 4: Transition Diagram :



Finite State Machines and Finite Automata

Step 4: Transition Diagram :



Finite State Machines and Finite Automata

Step 5: Simulation / Example

$\delta(q_s, 312)$

$\delta(\qquad)$

$\delta(\qquad)$

$\delta(\qquad)$

$\delta(q_s, 7258)$

$\delta(\qquad)$

$\delta(\qquad)$

$\delta(\qquad)$

$\delta(\qquad)$

Finite State Machines and Finite Automata

Step 5: Simulation / Example

$\delta(q_s, 312)$

$\delta(q_0, 12)$

$\delta(q_1, 2)$

$\delta(q_0, \epsilon)$

Accept

$\delta(q_s, 7258)$

$\delta(q_1, 258)$

$\delta(q_0, 58)$

$\delta(q_2, 8)$

$\delta(q_1, \epsilon)$

Reject