

BCA Part-III

Paper-XX

Topic: Automata Theory

Prepared by: Dr. kiran Pandey
School of computer science

Email-id:

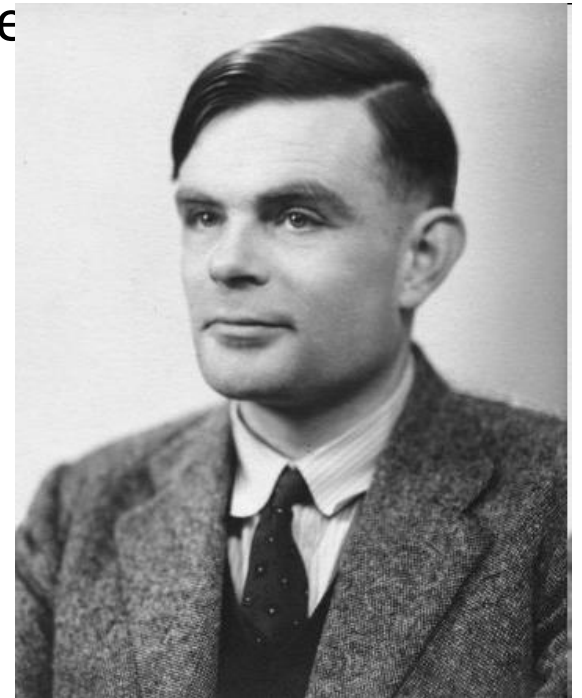
kiranpandey.nou@gmail.com

What is Automata Theory?

- *Study of abstract computing devices, or “machines”*
- **Automaton = an abstract computing device**
 - Note: A “device” need not even be a physical hardware!
- **A fundamental question in computer science:**
 - Find out what different models of machines can do and cannot do
 - The *theory of computation*
- Computability vs. Complexity

Alan Turing (1912-1954)

- Father of Modern Computer Science
- English mathematician
- Studied abstract machines called *Turing machines* even before computers existed
- Heard of the Turing test?



Theory of Computation(TOC): A Historical Perspective

1930s	<ul style="list-style-type: none">• Alan Turing studies Turing machines• Decidability• Halting problem
1940-1950s	<ul style="list-style-type: none">• “Finite automata” machines studied• Noam Chomsky proposes the “Chomsky Hierarchy” for formal languages
1969	Cook introduces “intractable” problems or “ NP-Hard ” problems
1970-	Modern computer science: compilers , computational & complexity theory evolve

Languages & Grammars

An **alphabet** is a set of symbols:

{0,1}

Or “**words**”

↓ **Sentences** are strings of symbols:

0,1,00,01,10,1,...

A **language** is a set of sentences:

$L = \{000,0100,0010,..\}$

A **grammar** is a finite list of rules defining a language.

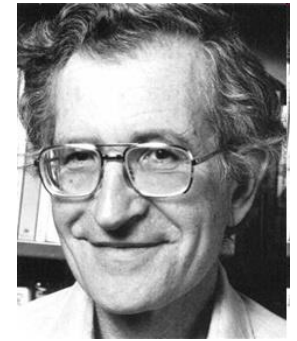
$S \longrightarrow 0A$ $B \longrightarrow 1B$

$A \longrightarrow 1A$ $B \longrightarrow 0F$

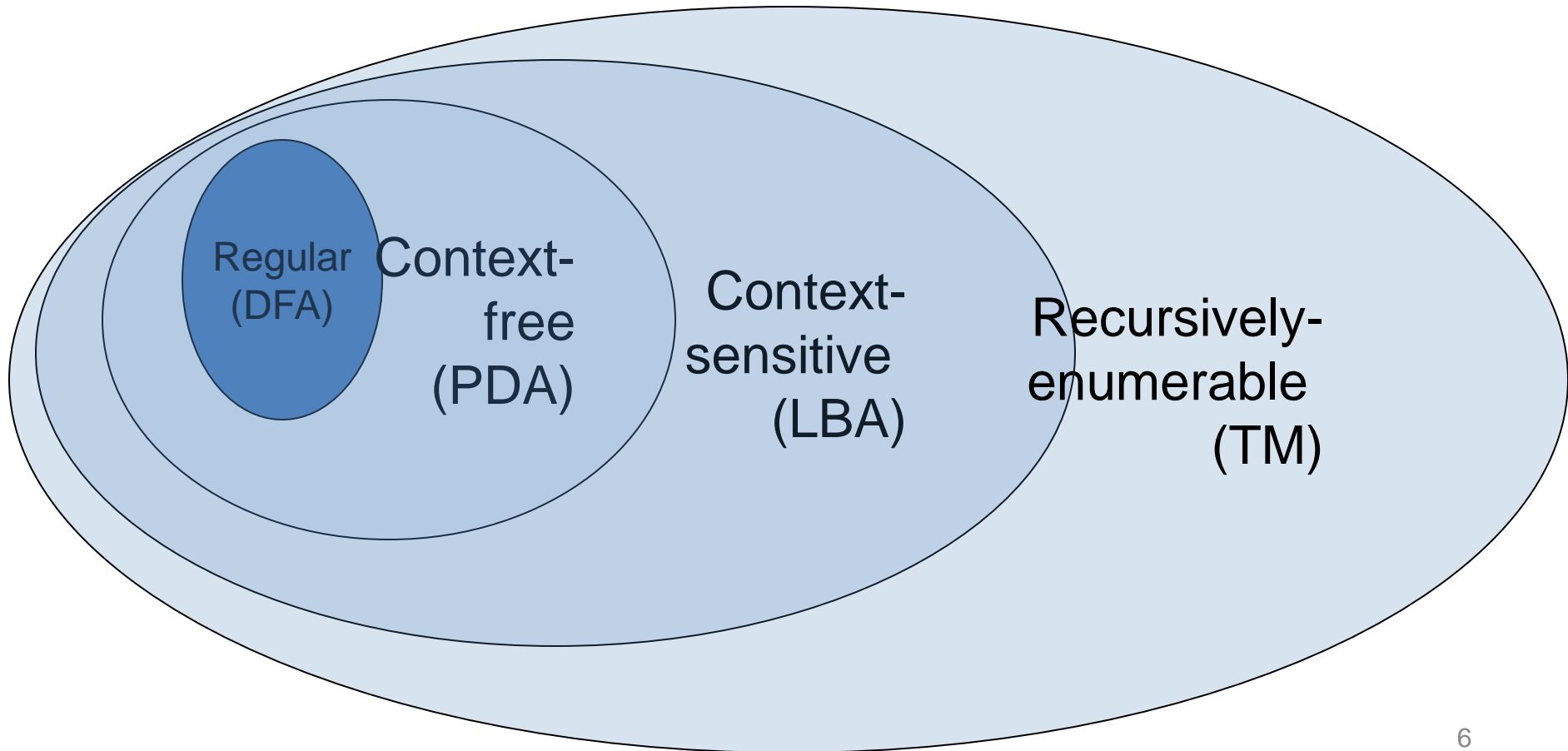
$A \longrightarrow 0B$ $F \longrightarrow \epsilon$

- Languages: “A language is a collection of sentences of finite length all constructed from a finite alphabet of symbols”
- Grammars: “A grammar can be regarded as a device that enumerates the sentences of a language” - nothing more, nothing less
- *N. Chomsky, Information and Control, Vol 2, 1959*

The Chomsky Hierarchy



- A containment hierarchy of classes of formal languages



BASIC CONCEPTS OF TOC

Alphabet

An alphabet is a finite, non-empty set of symbols

- We use the symbol Σ (sigma) to denote an alphabet
- Examples:
 - Binary: $\Sigma = \{0,1\}$
 - All lower case letters: $\Sigma = \{a,b,c,..z\}$
 - Alphanumeric: $\Sigma = \{a-z, A-Z, 0-9\}$
 - DNA molecule letters: $\Sigma = \{a,c,g,t\}$
 - ...

Strings

A string or word is a finite sequence of symbols chosen from Σ

- **Empty string is ε (or “epsilon”)**
- Length of a string w , denoted by “ $|w|$ ”, is equal to the *number of (non- ε) characters in the string*
 - E.g., $x = 010100$ $|x| = 6$
 - $x = 01 \varepsilon 0 \varepsilon 1 \varepsilon 00 \varepsilon$ $|x| = ?$
- xy = concatenation of two strings x and y

Powers of an alphabet

Let Σ be an alphabet.

– Σ^k = the set of all strings of length k

– $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$

– $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$

Languages

L is said to be a language over alphabet Σ , only if $L \subseteq \Sigma^$*

→ this is because Σ^* is the set of all strings (of all possible length including 0) over the given alphabet Σ

Examples:

1. Let L be *the* language of all strings consisting of n 0's followed by n 1's:

$$L = \{\epsilon, 01, 0011, 000111, \dots\}$$

2. Let L be *the* language of all strings of with equal number of 0's and 1's:

$$L = \{\epsilon, 01, 10, 0011, 1100, 0101, 1010, 1001, \dots\}$$

Definition: \emptyset denotes the Empty language

- Let $L = \{\epsilon\}$; Is $L = \emptyset$?

The Membership Problem

Given a string $w \in \Sigma^$ and a language L over Σ ,
decide whether or not $w \in L$.*

Example:

Let $w = 100011$

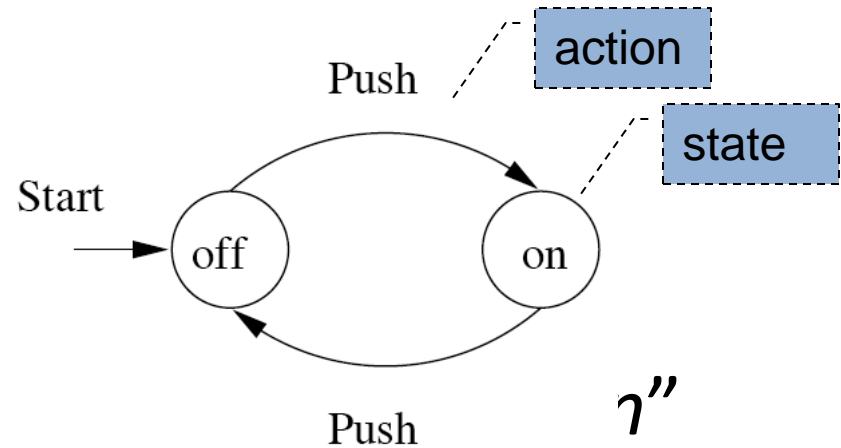
Q) Is $w \in$ the language of strings with equal number of 0s and 1s?

Finite Automata

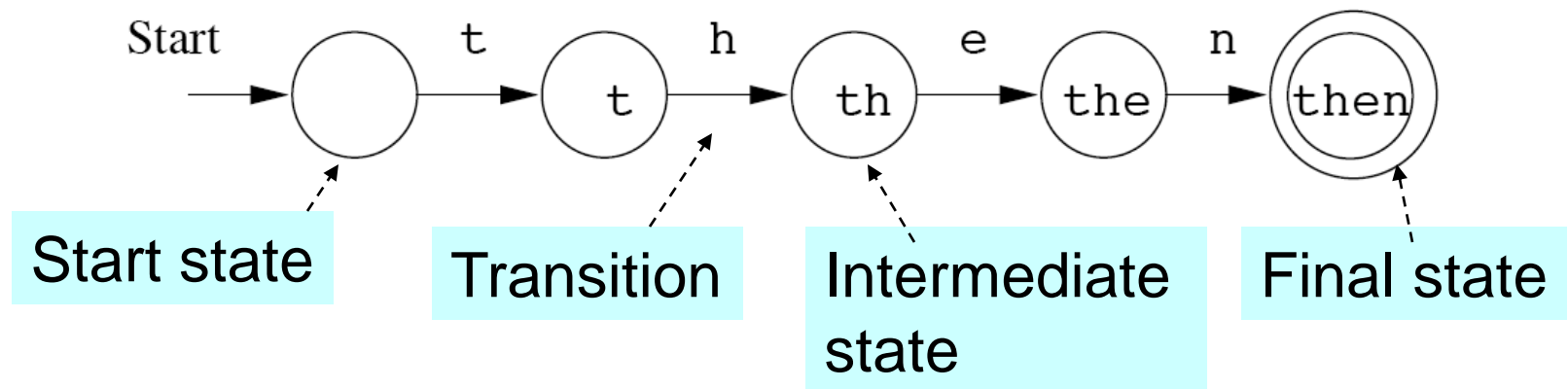
- Some Applications
 - Software for designing and checking the behavior of digital circuits
 - Lexical analyzer of a typical compiler
 - Software for scanning large bodies of text (e.g., web pages) for pattern finding
 - Software for verifying systems of all types that have a finite number of states (e.g., stock market transaction, communication/network protocol)

Finite Automata : Examples

- On/Off switch



- Modeling recognition



Regular expressions

- Grammars
- Regular expressions
 - E.g., regular expression to capture: “Patna”:
 - `[A-Z][a-z]*`