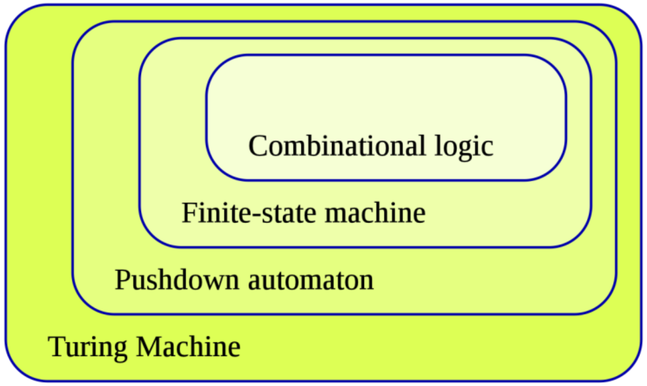
Introduction to Automata Theory

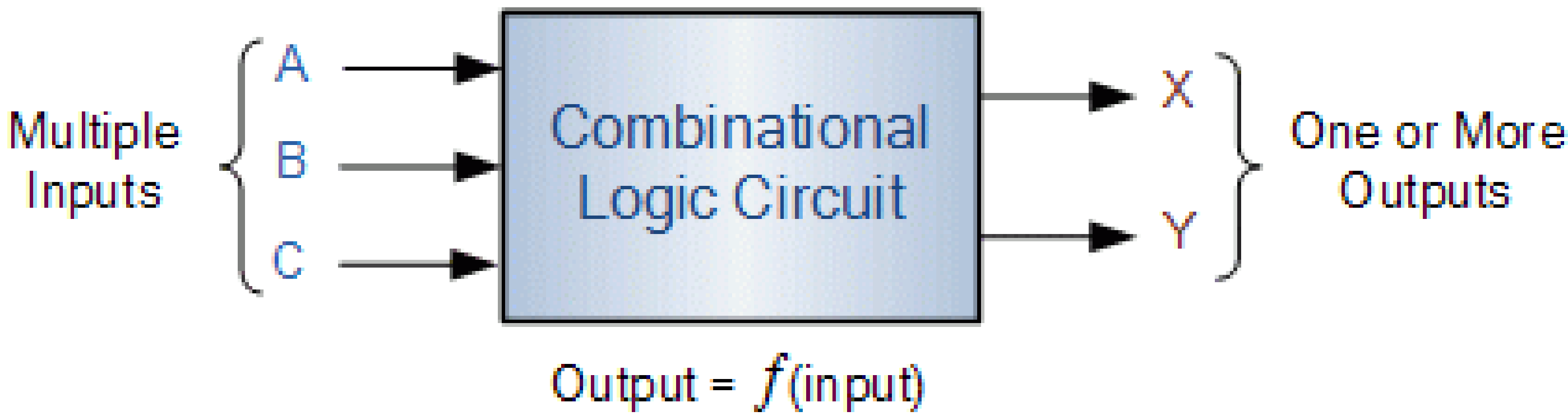
Dr. Ankit Rajpal
Assistant Professor
Department of Computer Science
University of Delhi

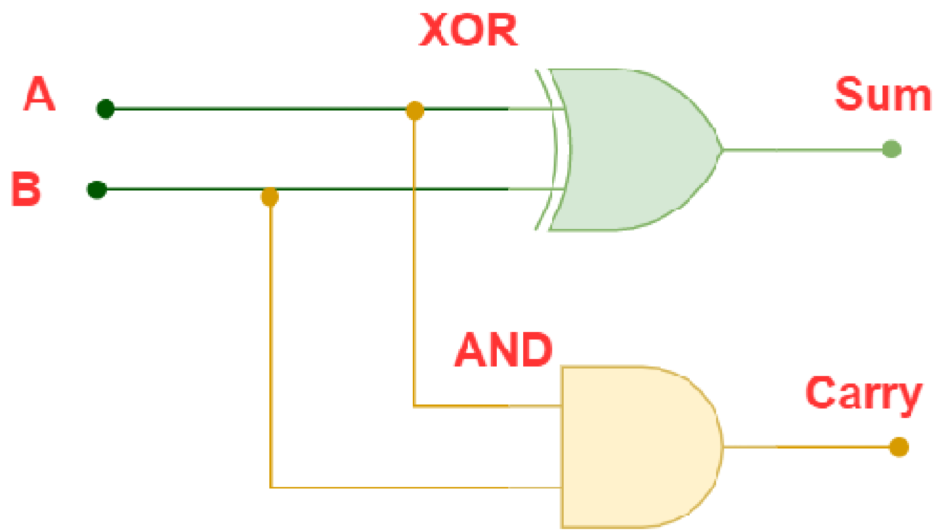
What is Automata Theory?

- *Study of abstract computing devices, or "machines"*
- Automaton = an abstract computing device designed to respond to encoded instructions.
- An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically.
- 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



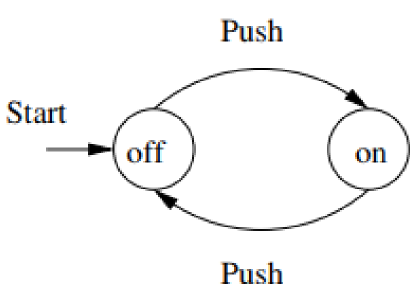
Combinational Logic Circuits





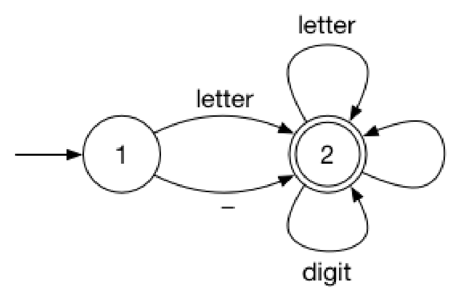
Finite Automaton/Finite State Machines

- Restricted model of an "actual computer" (why?)
  - It receives its input as a string on the input tape.
  - It delivers no output at all, except an indication of whether or not the input is considered acceptable.
  - Language Recognition Devices



A finite automaton modeling an on/off switch

```
letter       = [a-zA-Z]
digit        = [0-9]
identifier   = (letter | _) (letter | digit | _)*
```

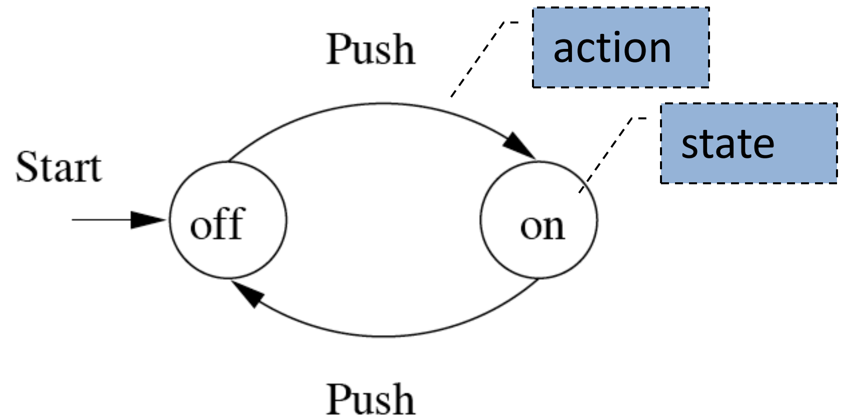


Finite Automata

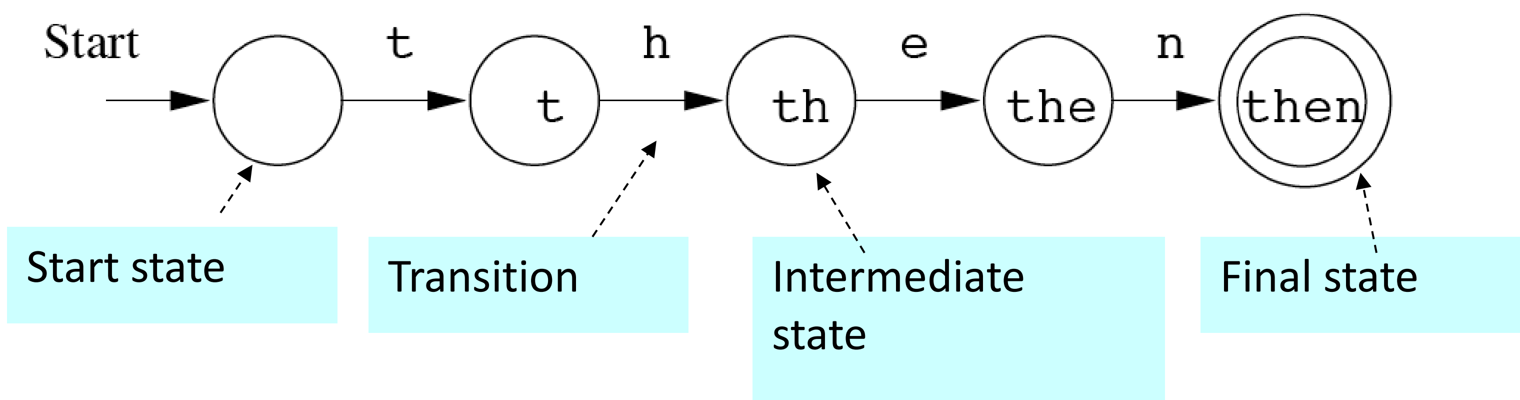
- Some Applications
  - Software for designing and checking the behavior of digital circuits
  - Lexical analyzer of a typical compiler
  - Software for scanning large bodies of text (e.g., web pages) for pattern finding
  - Software for verifying systems of all types that have a finite number of states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

Finite Automata : Examples

- On/Off switch



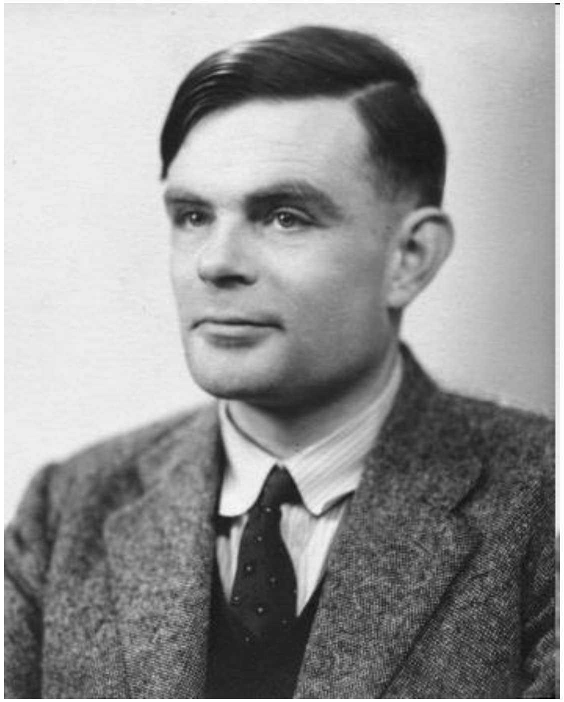
- Modeling recognition of the word *"then"*



(A pioneer of automata theory)

Alan Turing (1912-1954)

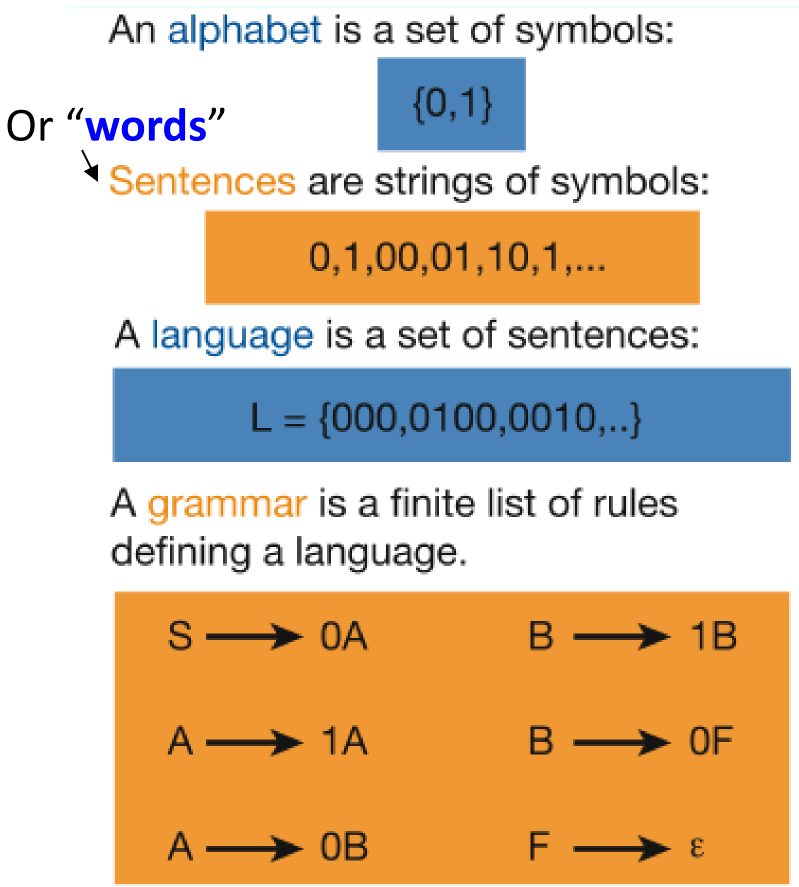
- Father of Modern Computer Science
- English mathematician
- Studied abstract machines called ***Turing machines*** even before computers existed
- Heard of the Turing test?



Theory of Computation: A Historical Perspective

1930s	• Alan Turing studies Turing machines • Decidability • Halting problem
1940-1950s	• "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

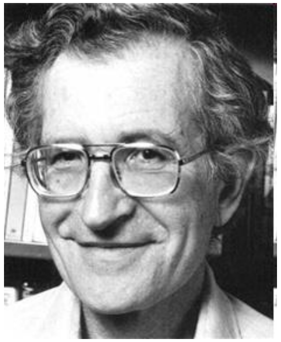


- <u>Languages</u>: *"A language is a collection of sentences of finite length all constructed from a finite alphabet of symbols"*
- <u>Grammars</u>: *"A grammar can be regarded as a device that enumerates the sentences of a language"* - nothing more, nothing less

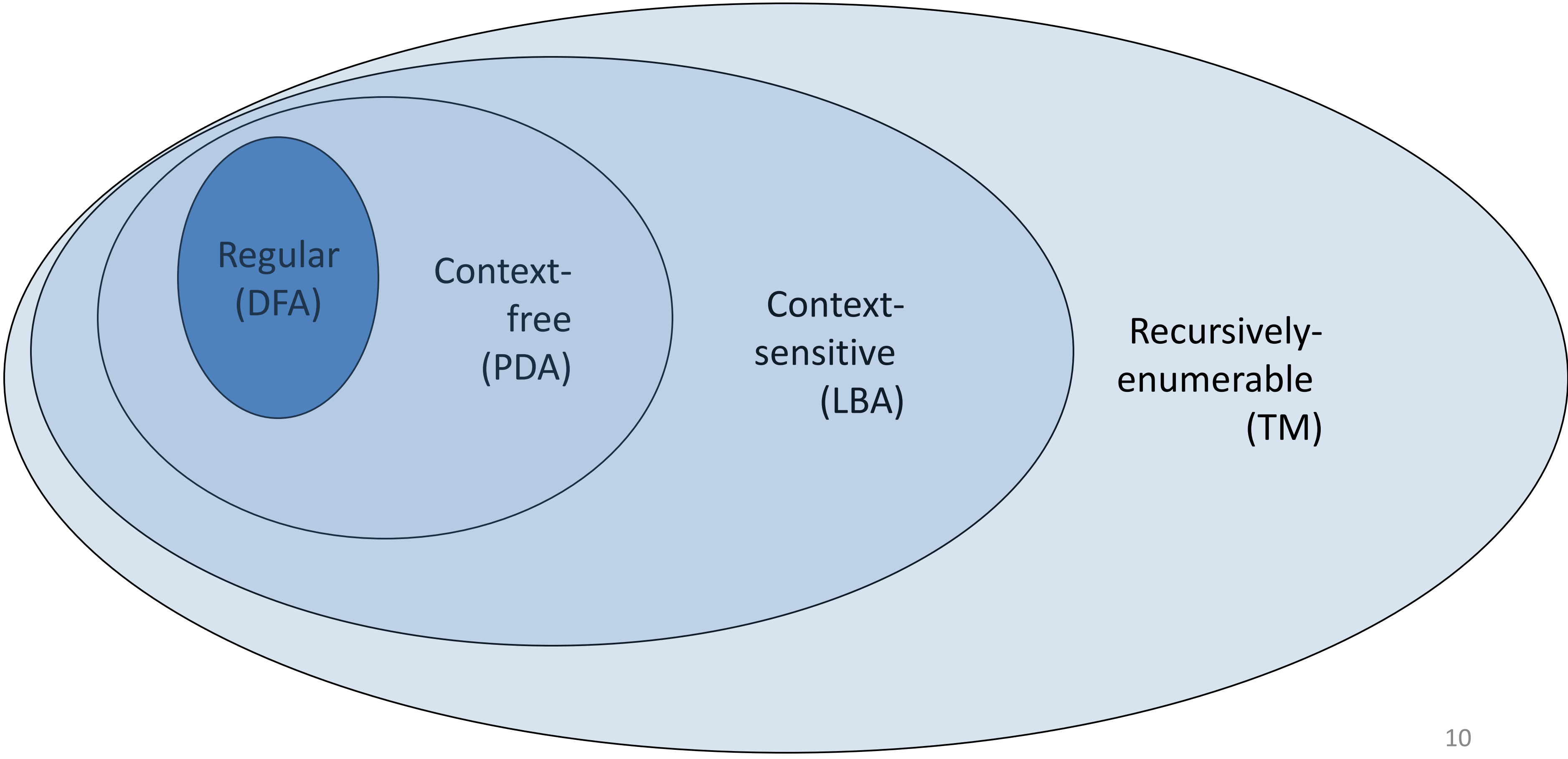
- *N. Chomsky, Information and Control, Vol 2, 1959*

Image source: Nowak et al. Nature, vol 417, 2002

The Chomsky Hierachy



- A containment hierarchy of classes of formal languages



Alphabet

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

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

Strings

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

- ***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 ε 0 ε 1 ε 00 ε* *|x| = ?*

– *xy = c*oncatenation 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

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

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

Languages

L is a 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:

   $L = \{\varepsilon, 01, 0011, 000111, \ldots\}$

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

   $L = \{\varepsilon, 01, 10, 0011, 1100, 0101, 1010, 1001, \ldots\}$

   Canonical ordering of strings in the language

Definition: **Ø denotes the Empty language**

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

NO

The Membership Problem

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

<u>Example:</u>

Let w = 100011

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