Formal Language & Automata Theory

Prof. Sankhadeep Chatterjee

Syllabus Outline

- Introduction
- Regular Languages and Finite Automata
- Context-Free Languages and Pushdown Automata
- Context-Sensitive Languages
- Turing Machines
- Undecidability

References

- “Introduction to Automata Theory, Languages, and Computation” Ullman, Motwani, Hopcroft (Pearson)
- “Automata and Computability” Dexter Kozen (Springer)
- “Introduction to the Theory of Computation” Michael Sipser (Cengage)
- “Introduction to Languages and the Theory of Computation” John C Martin (Tata Mcgraw-Hill)
- “Introduction to Theory of Computation” Bikash Kanti Sarkar and Ambuj Kumar (Universities Press)
- “An Introduction to Formal languages & Automata” Peter Linz (Jones & Bartlett)
- “Formal Languages and Automata Theory” Anami, Karibasappa (Wiley)
- “Switching and Finite Automata Theory” Kohavi, Jha (Cambridge University Press)

Introduction

- Why study Formal Language & Automata Theory (Theory of Computation)?
  - What a computing machine can do and what it can not do?
  - Which problems can be solved efficiently by computer?
  - Which problems are hard to solve (intractable)?

Introduction

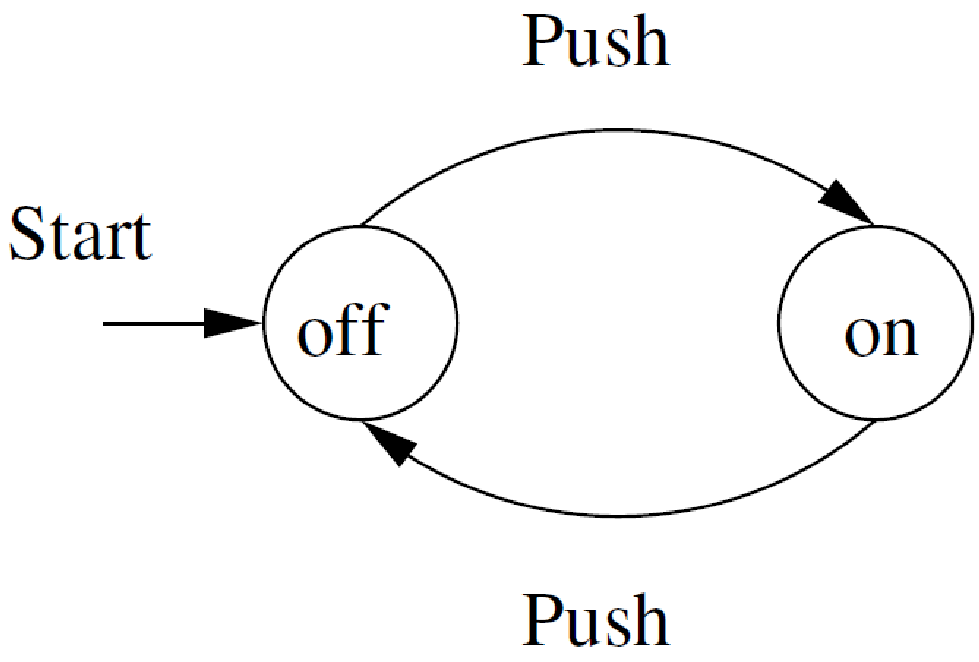
- Software for designing and checking the behaviour of digital circuits
- The “lexical analyser” of a typical compiler, i.e. the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation
- Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns
- Software for verifying systems of all types that have a finite number of distinct states e.g. communications protocols

Finite Automata

- Automata (Greek) – *Self Acting*
- Mathematical model of a computing device / software

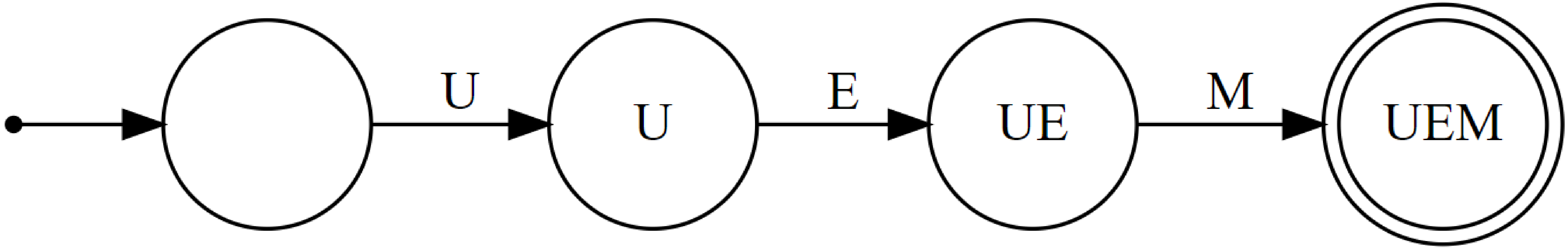
- Switch as an Finite Automaton

- The device remembers whether it is in the “on” state or the “off” state



Finite Automata

- Word recognition using finite automata
  - Find the word “UEM” in a piece of text



- It needs four states, each of which represents a different position in the word that has been reached so far

Central Concepts of Automata

- **Alphabets:** An alphabet is a finite, nonempty set of symbols. Conventionally, we use the symbol Σ for an alphabet.
  1. $\Sigma = \{0, 1\}$ is a binary alphabet
  2. $\Sigma = \{a, b, \dots, z\}$ set of all lower case letters
  3. The set of all ASCII characters, or the set of all printable ASCII characters
- **Strings:** A string (or sometimes word) is a finite sequence of symbols chosen from some alphabet.
  1. For example "01101" is a string from the binary alphabet $\Sigma = \{0, 1\}$.
  2. The string "111" is another string chosen from this alphabet.

Central Concepts of Automata

- **Empty String:** The empty string is the string with zero occurrences of symbols. Usually denoted by $\boldsymbol{\epsilon}$
- **Length:** The number of positions for symbols in the string.
  - The standard notation for the length of a string w is $|w|$
  - For example, $|011| = 3$ and $|\epsilon| = 0$
- **Powers of an Alphabet:** Σ^k to be the set of strings of length k, each of whose symbols is in Σ

  Considering $\Sigma = \{0, 1\}$,
  $\Sigma^0 = \{\epsilon\}$, $\Sigma^1 = \{0, 1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$
  $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Central Concepts of Automata

- The set of all strings over an alphabet Σ is conventionally denoted Σ^*
  - $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$
- In other words

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$$

- The non empty set of strings are denoted as follows

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \cdots$$

$$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$$

Central Concepts of Automata

- **Language:** A set of strings all of which are chosen from some Σ^*, where Σ is a particular alphabet, is called a language.
  - If Σ is an alphabet, and $L \subseteq \Sigma^*$ then L is a ***Language over Σ***
  - The language of all strings consisting of n 0's followed by n 1's, for some $n \geq 0$: $\{\epsilon,\ 01, 0011, 000111, \ldots\}$
  - The set of strings of 0's and 1's with an equal number of each: $\{\epsilon,\ 01, 10, 0011, 1010, 0101, \ldots\}$
  - Ø denotes **Empty Language**
  - $\{\epsilon\}$ is a language consisting of only empty string (Note: $Ø \neq \{\epsilon\}$)

Central Concepts of Automata

- **Problems:** A problem is the question of deciding whether a given string is a member of some particular language

- Given a string w in Σ^*decide whether or not w is in L