Theory of Automata

Shakir Ullah Shah

Recommended Book(s)

- Text Book
  - Introduction to Computer Theory, by Daniel I. A Cohen, John Wiley and Sons, Inc., Second Edition
- Reference Book(s)
  - John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages and Computation, Second Edition. Addison-Wesley, 2001.
  - John C. Martin. Introduction to Languages and the Theory of Computation. McGraw-Hill, 2003.
  - Introduction to the Theory of Computation, Michael Sipser, 2nd edition, Thomson Course Technology, 2005

Objectives

- To introduce the basic parts of formal languages
- To describe the various methods to define a language
- To teach different formal models of computation such as Finite Automata and Turing Machines
- To build mathematical models and then to study their limitations
- To understand about the various Types of Languages, Including

Outcomes

- Understand a language and its basic parts
- How to define languages
- Finding the language
  - successful inputs of a machine
- Language processing machines, including FSA , TG , Mealy & Moore machines
- Comparison of various Languages

Tentative Class Policy

- Grading
  - Assignments 10%
  - Quizzes 10%
  - Class Participation 5%
  - Project 10%
  - Sessional Exams 25%
  - Final Exams. 40%

What does Theory of automata mean?

- The word “Theory” means that this subject is a more mathematical subject and less practical.
- It is not like your other courses such as programming. However, this subject is the foundation for many other practical subjects.
- Automata is the plural of the word Automaton which means “self-acting”
- In general, this subject focuses on the theoretical aspects of computer

Theory of Automa Applications

- This subject plays a major role in:
  - Theory of Computation
  - Compiler Construction
  - Parsing
  - Formal Verification
  - Defining computer languages

Background

- In this course we will consider a ***mathematical model*** of computing, called **machines**, and then to study their limitations by analyzing the types of **inputs** on which they can operate successfully.
- The collection of these **successful inputs** is called the **language** of the machine.
- These **theoretical models** helped/lead to the construction of

Background (cont.)

- Every time we introduce a new machine, we will learn its **language**; and every time we develop a new language, we will try to find a **machine** that corresponds to it.

- This interplay between languages and machines will be our way of investigating problems and their potential solutions by automatic procedures which we call

Background (cont.)

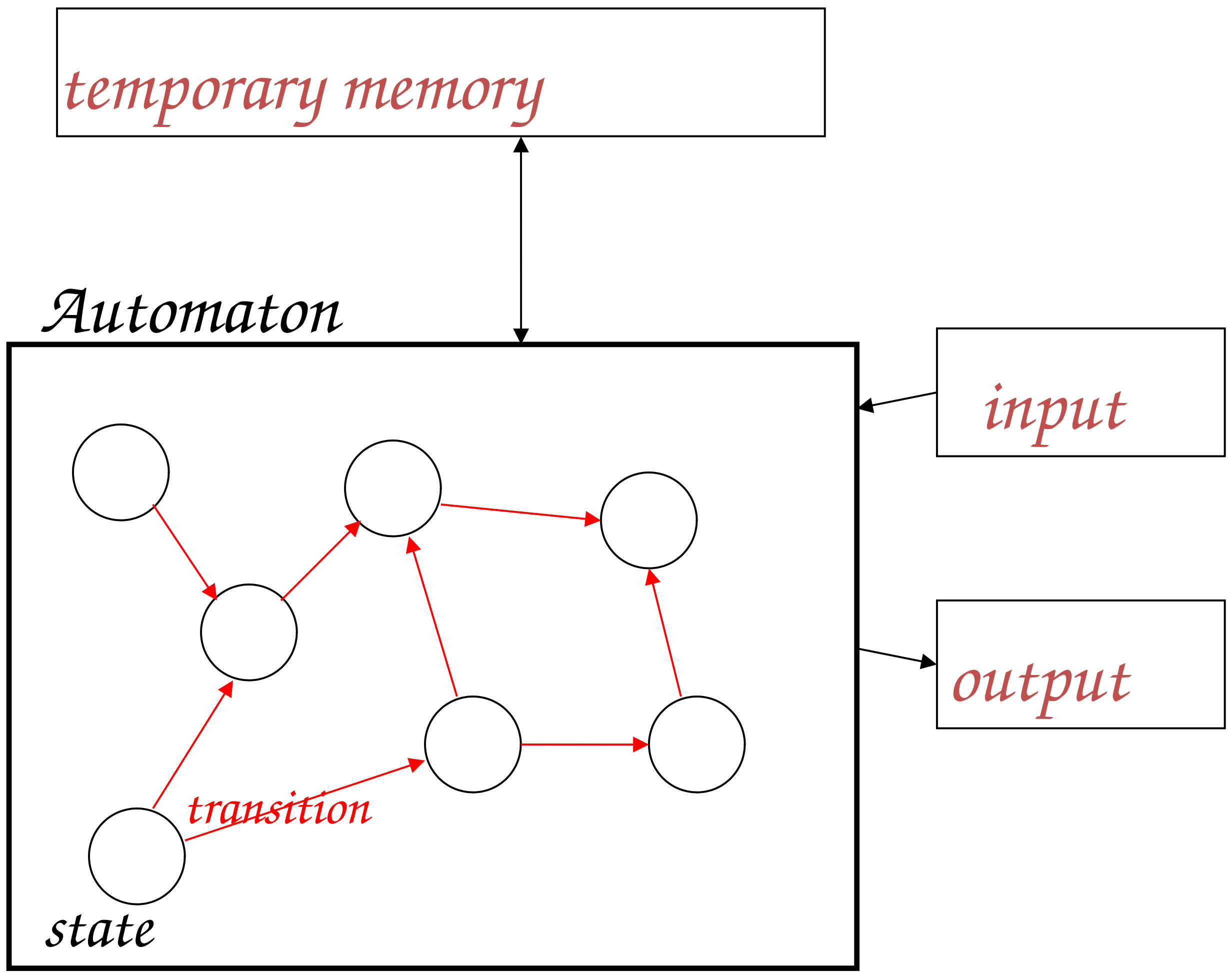
- We will arrive at what we may believe to be the most powerful machine possible. When we do, we will be surprised to find tasks that even such machine cannot perform.

- Our ultimate result is that no matter what machine we build, there will always be questions that are simple to state and that the machine can not answer

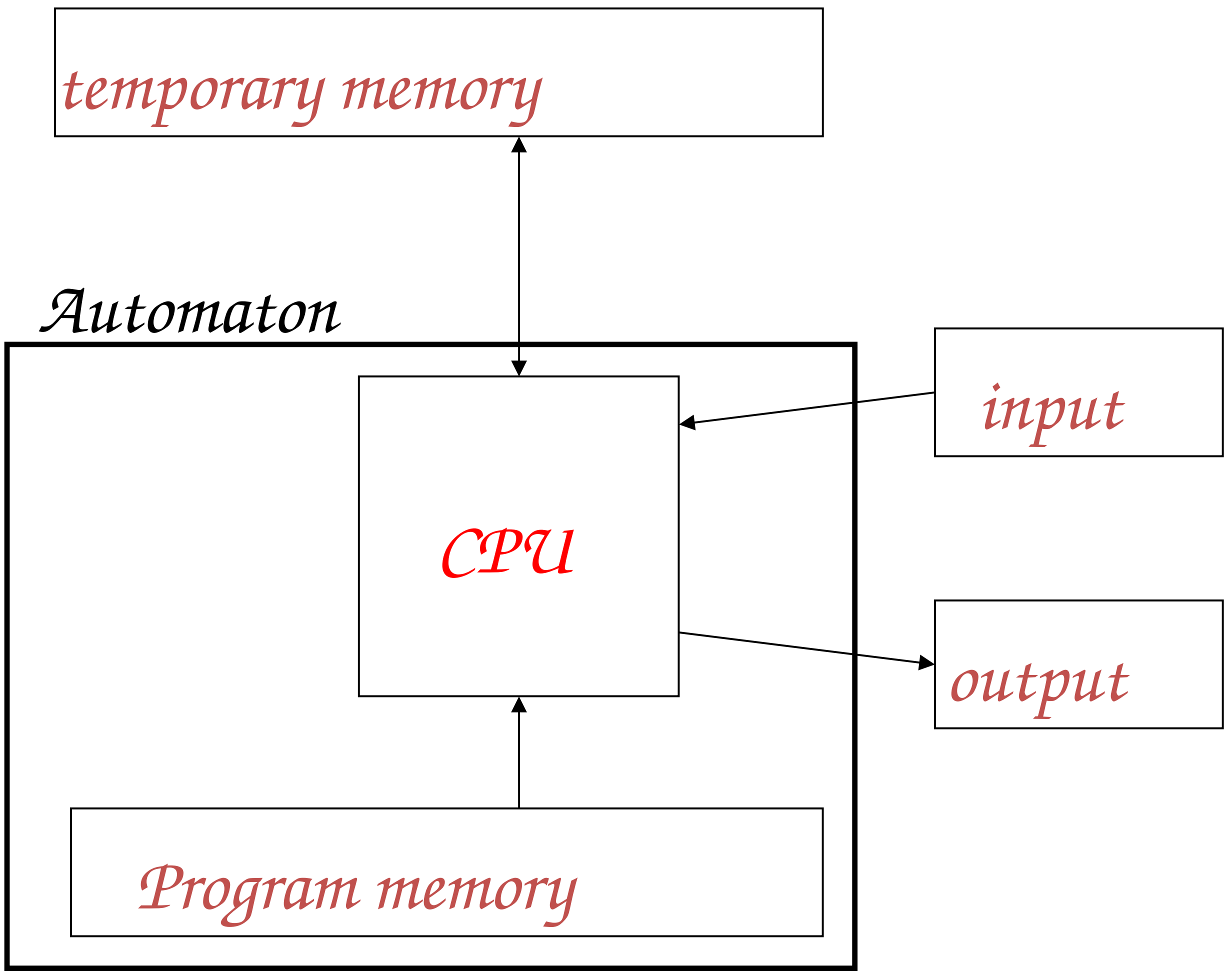
What do automata mean?

- It is the plural of automaton, and it means "something that works automatically"

Automaton


temporary memory
Automaton
input
output
transition
state

Automaton


temporary memory
Automaton
CPU
input
output
Program memory

Types of languages

- There are two types of languages
  - Formal Languages are used as a basis for defining computer languages
    - A predefined set of symbols and string
    - Formal language theory studies purely syntactical aspects of a language (e.g., word **abcd**)
  - Informal Languages such as English has many different versions.

Basic Element of a Formal Language – Alphabets

- Definition:
  A finite non-empty set of symbols (letters), is called an alphabet. It is denoted by Greek letter sigma Σ.
- Example:
  $\Sigma=\{1,2,3\}$
  $\Sigma=\{0,1\}$ //Binary digits
  $\Sigma=\{i,j,k\}$

Basic Element of a Formal Language – Alphabets

- Alphabet:

Alphabet	**Symbols**	**Symbol Name**	**String Name**
binary	01	bit	Bit string
Eng .Alph.	abcdefghijklm nopqrstuvwxy z ABCDEFGHIJKL MNOPQRSTUV WXYZ	letter	word
decimal	0123456789	digit	integer
special	~!@#$ %^&*()_- +={[}]|\\:;"'<, >.?/		
keyboard	Eng .Alph. + decimal + special+..	keystroke	typescript
ASCII	..	byte	String

String

- *A string over the alphabet Σ means a string all of whose symbols are in Σ*
- Example:
  If Σ= {a,b} then
  a, abab, aaabb, ababababababababab
- The set of all strings of length 2 over the alphabet {*a,b} is*
  - *{aa, ab, ba, bb}*

What is an EMPTY or NULL String

- A string with no symbol is denoted by (Small Greek letter Lambda) λ or (Capital Greek letter Lambda) Λ. It is called an empty string or null string.
- We will prefer Λ in this course. Please don't confuse it with logical operator 'and'.
- One important thing to note is that we never allow Λ to be part of alphabet of a language

Substring, prefix and suffix

- **Substring**:

any consecutive sequence of symbols that occurs anywhere in a string. For example,

  - ab and bc are substrings in abc while cb or ac are not.

- **Prefix and Suffix**:

A beginning of a string (upto any symbol) is called prefix and ending is called suffix, if w=xy with |x|,|y| >=0, then x is prefix and y is a suffix

Word

- Words are strings belonging to some language.

Example:

If Σ= {a} then a language L can be defined as

$L=\{a^n : n=1,2,3,.....\}$ or $L=\{a,aa,aaa,....\}$

Here a,aa,... are the words of L but not ab.

- All words are strings, but not all

Ambiguity (Cont'd…)

- Example: an alphabet may contain letters consisting of group of symbols for example
- Σ_1= {A, aA, bab, d}.

- Now consider an alphabet Σ_2= {A, Aa, bab, d} and a string AababA.

Ambiguity (Cont'd...)

- This string can be factored in two different ways
  - (Aa), (bab), (A)
  - (A), (abab), (A)

  Which shows that the second group cannot be identified as a string, defined over $\Sigma = \{a, b\}$.
- This is due to ambiguity in the defined alphabet Σ_2

Ambiguity (Cont'd...)

- **Why Ambiguity comes**: A computer program first scans A as a letter belonging to Σ_2, while for the second letter, the computer program would not be able to identify the symbols correctly.
- **Ambiguity Rule**:- The Alphabets should be defined in a way that letters consisting of more than one symbols should not start with a letter, already being used by some other letter.

Ambiguity Examples

- $\Sigma_1 = \{A, aA, bab, d\}$
- $\Sigma_2 = \{A, Aa, bab, d\}$

Σ_1 is a valid alphabet while Σ_2 is an in-valid alphabet.

Similarly,

- $\Sigma_1 = \{a, ab, ac\}$
- $\Sigma_2 = \{a, ba, ca\}$

In this case, Σ_1 is a invalid alphabet while Σ_2 is a valid alphabet.

String Operations

- Length

We define the function **length** of a string to be the number of letters in the string s, denoted by |s|.

- Example:

Σ={a,b}

s=ababa

|s|=5

In any language that includes the null word **Λ**, then length(**Λ**) = 0

For any word w in any language, if length(w) = 0 then w = **Λ**.

Word Length Example

- Example:
  $\Sigma = \{A, aA, bab, d\}$
  s=AaAbabAd
  Factoring=(A), (aA), (bab), (A), (d)
  $|s|=5$
- One important point to note here is that aA has a length 1 and not 2.

Length of strings over n alphabets

- **Formula:** Number of strings of length 'm' defined over alphabet of 'n' letters is $\mathbf{n^m}$
- Examples:
  - The language of strings of length 2, defined over Σ={a,b}is L={aa, ab, ba, bb} *i.e.* number of strings = 2^2
  - The language of strings of length 3, defined over Σ={a,b} is L={aaa, aab, aba, baa, abb, bab, bba, bbb} *i.e.* number of strings = 2^3

String Operations

- Concatenation: wy, w^K

Let $w = w_1 \ldots w_k$ and $y = y_1 \ldots y_k$ be two strings over some alphabet Σ . Then the concatenation of w and y (in symbols $w \cdot y$, or just wy) is the string $w_1 \ldots w_k y_1 \ldots Y_{k,}$

- If $w = a1 \ldots a_n$ and $y= b1 \ldots b_m$ then w.y (or wy) $= a1 \ldots a_n b1 \ldots b_m$

$$\underbrace{www \ldots w}_{k \text{ times}}.$$

Notation: by w^K we denote w

String Operations

The reverse of a string s denoted by Rev(s) or s^R, is defined as as follows:

If s= Λ then

s^R= Λ

otherwise

If $s=s_1 \ldots s_k$ then

$s^R=s_k \ldots s_1$

s^R is obtained by writing the letters of s in reverse order.

Example 1:

If s=abc is a string defined over Σ={a,b,c} then

Rev(s) or s^R = cba

String Operations

- Example:
  Σ= {A, aA, bab, d}
  s=AaAbabAd
  Rev(s)=dAbabaAA or
  Rev(s)= dAbabAaA

Which one is correct?

About Null

- The empty set Ø is a language which has no strings.
- Let L= Ø then It is not true that **Λ** is a word in the language ø since this language has no words at all.

- The set {**Λ** } is a language which has one string, namely **Λ** . So it is not empty.
- If a certain language L does not contain the word **Λ** and we wish to

Defining Languages

- The rules for defining a language can be of two kinds:
  - They can tell us how to test if a string of alphabet letters is a valid word, or
  - They can tell us how to construct all the words in the language by some clear procedures.

Defining Languages (contd.)

- The languages can be defined in different ways, such as

1. Descriptive definition,
2. Recursive definition,
3. Regular Expressions(RE)
4. Finite Automaton(FA) etc.

- Descriptive Definition:

The language is defined by describing the conditions imposed on its words.

Descriptive definition of language

Examples

1. The language L of strings of odd length, defined over Σ={a}, can be written as

   L1={a, aaa, aaaaa,…..}

2. The language L of strings that does not start with a, defined over Σ ={a,b,c}

   L2 ={Λ, b, c, ba, bb, bc, ca, cb, cc, …}

3. The language L of strings of length

Descriptive definition of language

4. The language L of strings ending in 0, defined over Σ ={0,1}, can be written as

L4={0,00,10,000,010,100,110, …}

5. The language **EQUAL**, of strings with number of a’s equal to number of b’s, defined over Σ={a,b}

L5={Λ ,ab,aabb,abab,baba,abba, …}

6. The language **EVEN-EVEN** of strings

Descriptive definition of language

7. The language $\{a^n b^n\}$, of strings defined over $\Sigma=\{a,b\}$, as $\{a^n b^n: n=1,2,3,\ldots\}$

L7={ab, aabb, aaabbb,aaaabbbb, …}

8. The language FACTORIAL, of strings defined over $\Sigma=\{a\}$, as $\{a^{n!} : n=1,2,3, \ldots\}$, can be written as

L8={a,aa,aaaaaa,…}.