Theory of Automata and Formal Language

Lecture-1

Dharmendra Kumar
(Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research, Prayagraj

March 30, 2021

Syllabus

Unit-I:

Basic Concepts and Automata Theory:Introduction to Theory of Computation-Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with -Transition, Equivalence of NFA's with and without -Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata, Myhill-Nerode Theorem, Simulation of DFA and NFA

Syllabus

Unit-II:

Regular Expressions and Languages:Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem, Algebraic Method Using Arden's Theorem, Regular and Non-Regular Languages- Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties, Finite Automata and Regular Languages, Regular Languages and Computers, Simulation of Transition Graph and Regular language.

Unit-III:

Regular and Non-Regular Grammars: Context Free Grammar(CFG)- Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars- Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms- Chomsky Normal Form(CNF), Greibach Normal Form (GNF), Chomsky Hierarchy, Programming problems based on the properties of CFGs.

Unit-IV:

Push Down Automata and Properties of Context Free Languages: Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA, Deterministic Pushdown Automata (DPDA) and Deterministic Context free Languages(DCFL), Pushdown Automata for Context Free Languages, Context Free grammars for Pushdown Automata, Two stack Pushdown Automata, Pumping Lemma for CFL, Closure properties of CFL, Decision Problems of CFL, Programming problems based on the properties of CFLs.

Unit-V:

Turing Machines and Recursive Function Theory: Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Turing Machine as Computer of Integer Functions, Universal Turing machine, Linear Bounded Automata, Church's Thesis, Recursive and Recursively Enumerable language, Halting Problem, Post's Correspondance Problem, Introduction to Recursive Function Theory.

Text books

1. Introduction to Formal Languages and Automata, Peter Linz
2. Theory of Computer Science : Automata, Languages and Computation, K.L.P. Mishra, N, Chandrasekaran and K.L.P. Mishra
3. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
4. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill
5. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI

Course Outcome

CO 1	Define and understanding of different types of languages and grammars.
CO 2	Understanding the concepts of finite automata and design finite automata for the languages.
CO 3	Describe the concepts of regular expression and apply these concepts to find regular expression for a set.
CO 4	Illustrate and apply the pumping lemma for the languages and demonstrate the normal form of the grammars.
CO 5	Understanding the concepts of pushdown automata and create pushdown automata for the languages.
CO 6	Understanding the concepts of Turing machine and create Turing machine for the languages and functions.

Basic Concepts

Alphabet

A finite set of symbols is said to be alphabet. We will denote it by set Σ .

String

This is the sequence of symbols from alphabet.

Example: If $\Sigma = \{a,b\}$, then abab, aaabab are strings on Σ.

Operations defined on strings

1. Concatenation

The concatenation of two strings w and v is the string obtained by appending the symbols of v to the right end of w, that is, if

w = $a_1 a_2 ....a_n$

and v = $b_1 b_2 ....b_m$

Then concatenation of w and v, denoted by wv, is

wv = $a_1 a_2 ....a_n b_1 b_2 ....b_m$

2. Reverse of a string

The reverse of a string is obtained by writing the symbols in reverse order. If w is a string then its reerse is denoted by w^R.

Example: If w = $a_1 a_2 ....a_n$ then

$w^R = a_n a_{n-1} ....a_2 a_1$

3. Length of a string

The length of a string is the number of symbols in the strings. If w is a string then it is denoted by $\mid w \mid$.

The string with length 0 is said to be empty string. And it is denoted by ϵ. It is also said to be null string.

Properties of empty string(ϵ)

1. $\mid \epsilon \mid = 0$
2. $\epsilon w = \mathrm{w} = \mathrm{w}\epsilon$, $\quad\forall$ string w.

.

Substring

Any string of consecutive characters in string w is said to be a substring of w. If

$$\mathrm{w} = \mathrm{vu}$$

Then the substrings v of u are said to be prefix and suffix of w respectively.

Properties of strings

1. $\mid uv \mid = \mid u \mid + \mid v \mid$
2. $w^n =$ wwwwwww............w(upto n times)
3. $w^0 = \epsilon$, $\forall w$

Kleene Closure(*-closure)

- If Σ is an alphabet, then Σ^* denote the kleene closure of Σ.
- Σ^* is the set of all the strings obtained by concatenating zero or more symbols from Σ.
- $\Sigma^* = \cup_{i=0}^{\infty} \Sigma^i = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots\ldots\ldots\ldots\ldots\ldots$

Positive Closure(+-closure)

- If Σ is an alphabet, then Σ^+ denote the kleene closure of Σ.
- Σ^+ is the set of all the strings obtained by concatenating one or more symbols from Σ.
- $\Sigma^+ = \cup_{i=1}^{\infty}\Sigma^i = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup$ ....
  $= \Sigma^*$ - $\{\epsilon\}$

Example: Consider $\Sigma = \{$a, b$\}$. Find Σ^2 and Σ^3.