Theory of Computation

By Prashant Gautam

Course of Study

1. Basic Foundations
    - Computation, theory of computation, brief history
    - Abstract model
2. Finite automata
    - DFA, NFA
3. Regular Expression and languages
4. Context Free Grammar
5. Push Down Automata
6. Turing Machine
7. Computational complexity
    - Undecidability
    - Intractability

Text Book

Introduction to Automata Theory, Languages and Computation

By John E. Hopcroft, Rajeev Motwani & Jeffry D. Ullman

Reference Books

Introduction to Languages and Theory of Computation.

By John Martin, Addison Wiley Print

Elements of the theory of computation

By Harry R. Lewis, Christos H. Papadimitriou

Theory of Computation,

By Adosh Kumar Pandey

Unit-1

Basic Foundations

Computation

- If it involves a computer, a program running on a computer and numbers going in and out then computation is likely happening.

Theory of computation

- Study of power and limits of computing.
- It has three interacting components:

  # Automata Theory

  # Computability Theory

  # Complexity Theory

Computability Theory

- What can be computed?
- Are there problems that no program can solve?

Complexity Theory

- What can be computed efficiently?
- Are there problems that no program can solve in a limited amount of time or space?

Automata Theory

- Study of abstract machine and their properties, providing a mathematical notion of “computer”
- Automata are abstract mathematical models of machines that perform computations on an input by moving through a series of states or configurations. If the computation of an automaton reaches an accepting configuration it accepts that input.

Study of Automata

- For software designing and checking behavior of digital circuits.
- For designing software for checking large body of text as a collection of web pages, to find occurrence of words, phrases, patters (i.e. pattern recognition, string matching, …)
- Designing “lexical analyzer” of a compiler, that breaks input text into logical units called “tokens”

Abstract Model

- An abstract model is a model of computer system (considered either as hardware or software) constructed to allow a detailed and precise analysis of how the computer system works.
- Such a model usually consists of input, output and operations that can be performed and so can be thought of as a processor.

E.g. an abstract machine that models a banking system can have operations like

“deposit”, “withdraw”, “transfer”, etc.

Brief History

- Before 1930's, no any computer were there and Alen Turing introduced an abstract machine that had all the capabilities of today's computers. This conclusion applies to today's real machines.
- Later in 1940's and 1950's, simple kinds of machines called finite automata were introduced by a number of researchers.
- In late 1950's the linguist N. Chomsky begun the study of formal grammar which are closely related to abstract automata.
- In 1969 S. Cook extended Turing's study of what could and what couldn't be computed and classified the problem as:
    - Decidable
    - Tractable/intractable

Basic Concept of Automata Theory

- The basic terms that pervade the theory of automata include “alphabets”, “strings”, “languages”, etc.

Alphabets: (Represented by 'Σ')

- Alphabet is a finite non-empty set of symbols. The symbols can be the letters such as {a, b, c}, bits {0, 1}, digits {0, 1, 2, 3… 9}.
- Common characters like $, #, etc.
- {0,1} – Binary alphabets
- {+, −, *} – Special symbols

Strings: - (Strings are denoted by lower case letters)

- String is a finite sequence of symbols taken from some alphabet. E.g. 0110 is a string from binary alphabet, “automata” is a string over alphabet {a, b, c … z}.

Empty String

- It is a string with zero occurrences of symbols. It is denoted by ‘ε’ (epsilon).

Length of String

- The length of a string w, denoted by | w |, is the number of positions for symbols in w. we have for every string s, length (s) ≥ 0.
- | ε | = 0 as empty string have no symbols.
- | 0110 | = 4

- The set of **Power of alphabet**
- all strings of certain length k from an alphabet is the kth power of that alphabet. i.e. $\sum^k = \{w \,/\, |w| = k\}$

If $\sum = \{0, 1\}$ then,

$\sum^0 = \{\varepsilon\}$

$\sum^1 = \{0, 1\}$

$\sum^2 = \{00, 01, 10, 11\}$

$\sum^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Kleen Closure

- The set of all the strings over an alphabet ∑ is called kleen closure of ∑ & is denoted by ∑*. Thus, kleen closure is set of all the strings over alphabet ∑ with length 0 or more.

- $\therefore \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots\ldots\ldots\ldots$
- E.g. A = {0}
- $A^* = \{0^n \,/\, n = 0, 1, 2, \ldots\}$

Positive Closure

- The set of all the strings over an alphabet $\sum$, except the empty string is called positive closure and is denoted by $\sum+$.
- $\therefore \sum+ = \sum 1 \ \cup \sum 2 \cup \sum 3 \cup$ ……………

Language

- A language L over an alphabet ∑ is subset of all the strings that can be formed out of ∑; i.e. a language is subset of kleen closure over an alphabet ∑; $L \subseteq \sum^*$. (Set of strings chosen from $\sum^*$ defines language).

For example;

- Set of all strings over ∑ = {0, 1} with equal number of 0’s & 1’s.

  L = {ε, 01, 0011, 000111, ………}

- ϕ is an empty language & is a language over any alphabet.
- {ε} is a language consisting of only empty string.
- Set of binary numbers whose value is a prime:

  L = {10, 11, 101, 111, 1011, ……}

Concatenation of Strings

- Let x & y be strings then xy denotes concatenation of x & y, i.e. the string formed by making a copy of x & following it by a copy of y.
- More precisely, if x is the string of i symbols as x = a1a2a3…ai & y is the string of j symbols as y = b1b2b3…bj then xy is the string of i + j symbols as xy = a1a2a3…aib1b2b3…bj.
- For example; x = 000 y = 111 xy = 000111 & yx = 111000
- Note: ‘ε’ is identity for concatenation; i.e. for any w, εw = wε = w

Suffix of a string

- A string s is called a suffix of a string w if it is obtained by removing 0 or more leading symbols in w. For example; w = abcd s = bcd is suffix of w.

Prefix of a string

- A string s is called a prefix of a string w if it is obtained by removing 0 or more trailing symbols of w. For example; w = abcd s = abc is prefix of w,

Substring

- A string s is called substring of a string w if it is obtained by removing 0 or more leading or trailing symbols in w. It is proper substring of w if $s \neq w$.
- If s is a string then Substr (s, i, j) is substring of s beginning at ith position & ending at jth position both inclusive.

Problem

- A problem is the question of deciding whether a given string is a member of some particular language.
- In other words, if ∑ is an alphabet & L is a language over ∑, then problem is
- Given a string w in ∑*, decide whether or not w is in L.