COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO-10

Sub: Theory of Computer Science

COURSE: T.E.	Year: 2020-2021	Semester: V
DEPT: Computer Engineering		
SUBJECT CODE: CSC504		DUE DATE: 27/11/2020

==

Roll No. 50	Name: Amey Thakur
Class: TE-Comps B	Date of Submission: 25/11/2020

Tutorial 10

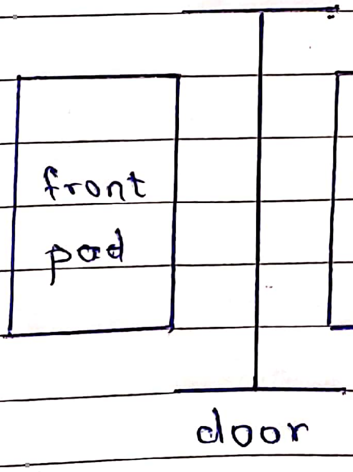
1. Discuss any real-life application based on FSM, PDA, TM, Grammar.

Q1 Discuss any real life application based on FSM, PDA, TM, Grammar.

Ans:

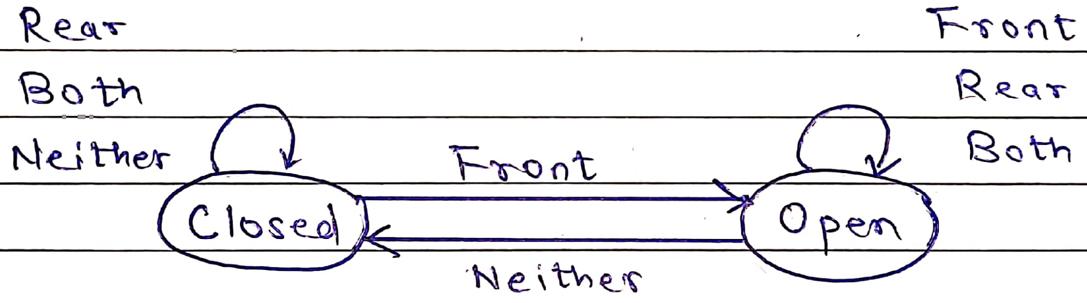
Finite State Machine (FSM)

- Finite automata are good models for computers with an extremely limited amount of memory.
- What can a computer do with such a small memory? Many useful things! In fact, we interact with such computers all the time, as they lie at the heart of various electromechanical devices.
- The controller for an automatic door is one example of such a device. Often found at supermarket entrances and exits, automatic doors swing open when sensing that a person is approaching. An automatic door has a pad in front to detect the presence of a person about to walk through a doorway Another pad is located to the rear of the doorway so that the controller can hold the door open long enough for the person to pass all the way through and also so that the door does not strike someone standing behind it as it opens.
- This configuration is as follows.



Top view of an automatic door

- The controller is in either of two states.:
  ① Open
  ② Closed
- There are four possible input conditions:
  ① Front-(Person is standing on the pad in front of the doorway).
  (II) Rear - (Person is standing on the pad to the rear of the doorway).
  (III) Both - (People are standing on both pads).
  (IV) Neither - (No one is standing on either pad).



- State diagram for automatic door controller.

	Neither	Front	Rear	Both
Closed	Closed	Open	Closed	Closed
Open	Closed	Open	Open	Open

Push-Down Automata (PDA)

- These automata are like nondeterministic finite automata but have an extra component called a stack.
- The stack provides additional memory beyond the finite amount available in the control.
- The stack allows pushdown automata to recognize some nonregular languages.
- Whenever your smartphone calculator calculates an arithmetic expression, it uses an implementation of the pushdown automata to evaluate it.
- It verifies that the parentheses are balanced. In the absence of parentheses, it makes sure that the multiplication operator has more precedence than the addition or subtraction operator. All of it can be summarized in three lines of grammar.

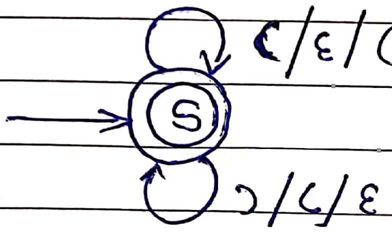
$$E \rightarrow E + T \mid E - T \mid T$$
$$T \rightarrow T * F \mid T / F \mid F$$
$$F \rightarrow x \mid y \mid (E) \mid -F$$

- The above grammar belongs to a class of languages called the Context Free Language. The variables E (Expression), T (Term) and F (Factor) produce more complex expressions or Terminal symbols (Numbers) in an arithmetic expression.
- The first rule is the starting point of any evaluation. Observe that the multiplication and division operations cannot be evaluated unless the PDA implementation sees T or F. This ensures the precedence of multiplication/division over addition/subtraction. Also observe that F produces another E (expression) only if it is wrapped in parentheses. This allows us to override the precedence of multiplication and division operations using parenthesis

Example: The Balanced Parentheses Language

Consider Balanced (B) = $\{w \in \{), (\}^*$: The parentheses are balanced$\}$.

- The following one state PDA M accepts B.
- M uses its stack to count the number of left parentheses that have not yet been matched.



⇒ If input is (, what's on stack top doesn't matter and a (is pushed to stack.

⇓

If input is) and (is on stack top, then (is popped and nothing is pushed to stack

M = (K, Σ, Γ, Δ, s, A), where:

K = {s} The states

Σ = {(,)} The input alphabets

Γ = {(} The stack alphabet

A = {s}

Δ contains:

((s, (, ε**), (s, ())

((s,), (), (s, ε))

** Important: This does not mean that the stack is empty

Turing Machine (TM)

- A much more powerful model, first proposed by Alan Turing in 1936, called Turing Machine.
- Similar to a finite automation but with an unlimited and unrestricted memory, a turing machine is a much more accurate model of a general purpose computer.
- A Turing machine can do everything that a real computer can do. Nonetheless, even a Turing machine cannot solve certain problems. In a very real sense, these problems are beyond the theoretical limits of Computation
- Turing machine are not used in real life programming It is a theoretical concept to define the notion of computability.
- :

(i) For solving any recursively enumerable problem
(ii) For understanding complexity theory
(iii) For implementation of Neural Networks
(iv) For implementation of Robotics applications.
(v) For implementation of Artificial Intelligence (AI)

Grammar

- Context Free Grammar are used in compilers and in particular parsing, taking a string based program and figuring out what it means.
- Typically, CFG are used to define the high level structure of a programming language.

Figuring out how a particular string was derived talks us about its structure and meaning.