Date: 27-July-2024

B.Sc. (Hons) Computer Science
DISCIPLINE SPECIFIC CORE- Theory of Computation (Guidelines)
Sem V (July 2024 Onwards)

Unit	Topic	Chapter	Reference	Total Hours
1. Introduction	Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.	2	[1]	4
2. Finite Automata and Regular Languages	Regular expressions	4	[1]	18
	Deterministic Finite Automata (DFA)	5	[1]	
	Transition graphs (TG)	6	[1]	
	Non-deterministic finite automata (NFA), Relationship between NFA and DFA	7	[1]	
	The relationship between regular languages and finite automata (Converting RE into FA and vice-versa), Kleene's Theorem	7	[1]	
	Properties of regular languages (Proof of Theorem 12 using De Morgan's law is to be done intuitively only)	9 (Excluding Pages 175-179)	[1]	
	Pumping lemma for regular grammars (Excluding Myhil-Nerode theorem & Quotient Languages)	10 (Till Page 195)	[1]	
3. Context-Free Languages (CFL)	Context-free grammars (CFG), Parse trees (Excluding Lukasiewicz notation), Ambiguities in grammars	12 (except pages 246-249)	[1]	15
	Deterministic and non-deterministic Pushdown Automata (PDA)	14		
	Chomsky Normal Form, Leftmost derivation,	13 (Page 275 onwards)		
	Properties of CFL (Excluding mixing context-free and regular languages)	17 (upto Pg 388)		
	Pumping lemma for CFL	16 (Till page 370)		
4. Turing Machines and Models of Computations	Turing machine as a model of computation, configuration of Turing machine, Recursive and recursively enumerable languages	4 (upto 4.2)	[2]	8
	Church Turing Thesis, Universal Turing Machine, decidability, Halting problem	5 (upto 5.3 except page 252)		

References

1. Cohen, D. I. A. (2011). **Introduction to Computer Theory**. 2nd edition. Wiley India.

2. Lewis, H.R. & Papadimitriou, H. R. (2002). **Elements of the Theory of Computation**. 2nd edition. Prentice Hall of India (PHI)

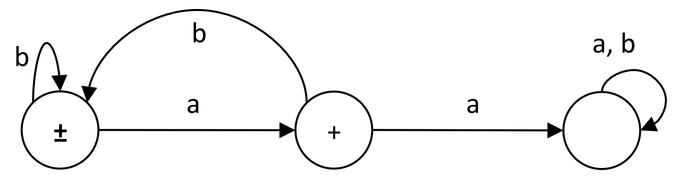
List of Practicals:

Section A

(The students should be given a regular language for designing and simulating FA. The machine is to be simulated using C++/Java.)

For Example: Design an FA that accepts a language over Σ={a, b} consisting of strings that do not contain a double a. Write a program to simulate this FA.

FA:



Pseudocode:

```
main(){
      w = user entered string
      State1(w, 0)
}
State1(string w, int i){
 print('State 1')
 if (i==w.length)
      print('String is accepted') //String ending in final state
      return;
 else
      if (w[i]=='b') State1(w, i+1)
      if (w[i]=='a') State2(w, i+1)
}
State2(string w, int i){
 print('State 2')
 if (i==w.length)
      print('String is accepted') //String ending in final state
      return;
 else
      if (w[i]=='b') State1(w, i+1)
      if (w[i]=='a') State3(w, i+1)
}
State3(string w, int i){
 print('State 3')
 if (i==w.length)
      print('String is rejected').  //String ending in non-final state
      return;
 else
      if (w[i]=='b') State3(w, i+1)
      if (w[i]=='a') State3(w, i+1)
}
```

Suggested List:

1. Design a Finite Automata (FA) that accepts all strings over Σ={0, 1} having three consecutive 1's as a substring. Write a program to simulate this FA.
2. Design a Finite Automata (FA) that accepts all strings over Σ={0, 1} having either exactly two **1**'s or exactly three **1**'s, not more nor less. Write a program to simulate this FA.

3. Design a Finite Automata (FA) that accepts language L_1, over $\Sigma=\{a, b\}$, comprising of all strings (of length 4 or more) having first two characters same as the last two. Write a program to simulate this FA.
4. Design a Finite Automata (FA) that accepts language L_2, over $\Sigma= \{a, b\}$ where $L_2= a(a+b)^*b$. Write a program to simulate this FA.
5. Design a Finite Automata (FA) that accepts language EVEN-EVEN over $\Sigma=\{a, b\}$. Write a program to simulate this FA
6. Write a program to simulate an FA that accepts
    a. Union of the languages L_1 and L_2
    b. Intersection of the languages L_1 and L_2
    c. Language L_1 L_2 (concatenation)

Section B

(The students should be given a context free language for designing and simulating PDA. The machine is to be simulated using C++/Java.)

Suggested List:

7. Design a PDA and write a program for simulating the machine which accepts the language $\{\mathbf{a^n b^n}$ where $n>0$, $\Sigma= \{a, b\}\}$.
8. Design a PDA and write a program for simulating the machine which accepts the language $\{wXw^r|$ w is any string over $\Sigma=\{a, b\}$ and w^r is reverse of that string and X is a special symbol $\}$.

Section C

(The students should be given a language for designing and simulating Turing machine. The machine is to be simulated using C++/Java.)

Suggested List:

9. Design and simulate a Turing Machine that accepts the language $a^nb^nc^n$ where $n >0$.
10. Design and simulate a Turing Machine which will increment the given binary number by 1.

Dr. Veenu Bhasin | Prof. Dr. Neeraj Kumar Sharma | Dr. Sonika Thakral