DHANALAKSHMI SRINIVASAN
COLLEGE OF ENGINEERING AND TECHNOLOGY
ACADEMIC YEAR 2018-19 (ODD SEM)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
QUESTIONBANK

SUBJECT: THEORY OF COMPUTATION
SUB CODE:CS6503
SEM/YEAR:V/III

UNIT I FINITE AUTOMATA

PART-A

1. Define finite automata.
2. Write the difference between the + closure and * closure.
3. Define alphabet, string, powers of an alphabet and concatenation of strings.
4. Define language and Grammar give an example.
5. What is a transition table and transition graph?
6. Give the DFA accepting the language over the alphabet 0, 1 that has the set of all strings beginning with 101.
7. Give the DFA accepting the language over the alphabet 0,1 that have the set of all strings that either begins or end(or both) with 01.
8. Define NFA.
9. Difference between DFA and NFA.
10. Write the notations of DFA.
11. Define ε-NFA.
12. Define the language of NFA.
13. Is it true that the language accepted by any NFA is different from the regular language? Justify your Answer.
14. Define Regular Expression.
15. List the operators of Regular Expressions
16. State pumping lemma for regular languages
17. Construct a finite automaton for the regular expression 0*1*.
18. List out the applications of the pumping lemma.
19. Define Epsilon –Closures.

PART-B

1. a) If L is accepted by an NFA with ε-transition then show that L is accepted by an NFA without ε-transition.
   b) Construct a DFA equivalent to the NFA. M=({p,q,r},{0,1}, δ,p,{q,s})
   Where δis defined in the following table.

	0	1
p	{q,s}	{q}
q	{r}	{q,r}
r	{s}	{p}
s	-	{p}

2. a)Show that the set L={a^n b^n/n>=1} is not a regular. (6) b)Construct a DFA equivalent to the NFA given below: (10)

	0	1
p	{p,q}	P
q	r	R
r	s	-
s	s	S

3. a) Check whether the language L=(0^n1^n / n>=1) is regular or not? Justify your answer.

   b) Let L be a set accepted by a NFA then show that there exists aDFA that accepts L.

4. a) Convert the following NFA to a DFA(10)

δ	a	b
p	$\{p\}$	$\{p,q\}$
q	$\{r\}$	$\{r\}$
r	$\{\phi\}$	$\{\phi\}$

   b) Discuss on the relation between DFA and minimal DFA (6)

5. a) Construct a NDFA accepting all string in {a, b} with either two consecutive a‟s or two Consecutive b‟s.

   b) Give the DFA accepting the following language
   Set of all strings beginning with a 1 that when interpreted as a binary integer is a Multiple of 5.

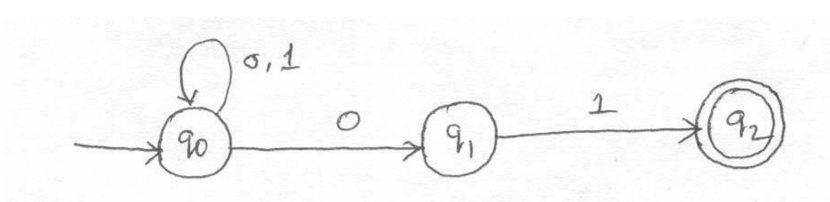
6. Draw the NFA to accept the following languages.

(i) Set of Strings over alphabet {0, 1,…….9} such that the final digit has appeared before. (8)

(ii)Set of strings of 0‟s and 1‟s such that there are two 0‟s separated by a number of positions that is a multiple of 4.

7. a) Let L be a set accepted by an NFA. Then prove that there exists a deterministic finite automaton that accepts L.Is the converse true? Justify your answer. (10)

   b)Construct DFA equivalent to the NFA given below: (6)



8. a) Prove that a language L is accepted by some ε–NFA if and only if L is accepted by some DFA. (8)

b) Consider the following ε–NFA. Compute the ε–closure of each state and find it"s equivalent DFA. (8)

	ε	A	b	C
p	{q}	{p}	Φ	Φ
q	{r}	ф	{q}	Φ
*r	Φ	ф	ф	{r}

9. a) Prove that a language L is accepted by some DFA if L is accepted by some NFA.

b) Convert the following NFA to it"s equivalent DFA

	0	1
p	{p,q}	{p}
q	{r}	{r}
r	{s}	ф
*s	{s}	{s}

10. a) Explain the construction of NFA with ε-transition from any given regular expres-sion.

b) Let A=(Q,∑, δ, q0 ,{qf) be a DFA and suppose that for all a in ∑we have δ(q0, a)= δ(qf ,a). Show that if x is a non empty string in L(A),then for all k>0,x^k is also in L(A).

11. Convert the following ε-NFA to DFA

states	ε	a	b	C
p	Φ	{p}	{q}	{r}
q	{p}	{q}	{r}	Φ
*r	{q}	{r}	ф	{p}

UNIT II GRAMMARS

PART-A

1. Define CFG.
2. Define production rule.
3. Define terminal and non terminal symbols.
4. Write about the types of grammars.
5. What is ambiguity?
6. Define sentential form.
7. Define parse tree.
8. What is a derivation?
9. What is a useless symbol and mention its types.
10. What is null production and unit production?
11. What are the two normal forms of CFG?
12. State Greibach normal form of CFG.
13. Mention the application of CFG.
14. Construct a CFG for the language of palindrome string over {a, b}.Write the CFG for the language, L=($a^n b^n$ | ≥n).
15. Construct a derivation tree for the string 0011000 using the grammar
    S->A0S |0 | SS , A-> S1A | 10.

16. Show that id+id*id can be generated by two distinct leftmost derivation in the grammar E->E+E | E*E | (E) | id .
17. Let G be the grammar S->aB/bA,A->a/aS/bAA,B->b/bS/aBB. obtain parse tree for the string aaabbabbba.
18. Find L(G)where G=({S},{0,1},{S->0S1,S->ε},S).
19. construct a context free Grammar for the given expression (a+b) $(a+b+0+1)^*$
20. Let the production of the grammar be S-> 0B | 1A, A-> 0 | 0S | 1AA, B-> 1|1S | 0BB.for the string 0110 find the right most derivation

PART-B

1. a. What are the closure properties of CFL? State the proof for any two properties. b. Construct a CFG representing the set of palindromes over (0+1)*.
2. a. if G is the grammar S → SbS | a show that G is ambiguous.
   b. Let G= (V,T, P,S) be a CFG. If the recursive inference procedure tells that terminal string w is in the language of variable A, then there is a parse tree with root A and yield w.
3. Discuss in detail about ambiguous grammar and removing ambiguity from grammar.
4. Discuss about eliminating useless symbols with example.
5. Explain about eliminating € productions with example.
6. What is a unit production and how will you eliminate it. Give example.
7. Prove that if G is a CFG whose language contains at least one string other than €, then there is a grammar G1 in Chomsky Normal Form such that L)G1) = L(G) –{€}.
8. Consider the grammar
   E → E + E | E*E | (E) | I
   I → a+b

   Show that the grammar is ambiguous and remove the ambiguity.

9. Simplify the following grammar S → aAa | bBb | BB
   A → C
   B → S | A C
   → S | €

10. Construct a grammar in GNF which is equivalent to the grammar S → AA | a
    A → SS | b

UNIT III PUSHDOWN AUTOMATA

PART-A

1. Give an example of PDA.
2. Define the acceptance of a PDA by empty stack. Is it true that the language accepted by a PDA by empty stack or by that of final state is different languages?
3. What is additional feature PDA has when compared with NFA? Is PDA superior over NFA in the sense of language acceptance? Justify your answer.
4. Explain what actions take place in the PDA by the transitions (moves)
   a. δ(q,a,Z)={(p1,γ1),(p2, γ2),…..(pm, γm)} and δ(q, ε,Z)= {(p1,γ1),(p2, γ2),…..(pm,γm)}.
   b. What are the different ways in which a PDA accepts the language? Define them. Is a true that non deterministic PDA is more powerful than that of deterministic
   c. PDA? Justify your answer.
5. Explain acceptance of PDA with empty stack.
6. Is it true that deterministic push down automata and non deterministic push
   a. Down automata are equivalent in the sense of language of acceptances? Justify your answer.
7. Define instantaneous description of a PDA.
8. Give the formal definition of a PDA.
9. Define the languages generated by a PDA using final state of the PDA and empty stack of that PDA.
10. Define the language generated by a PDA using the two methods of accepting a language.
11. Define the language recognized by the PDA using empty stack.
12. For the Grammar G defined by the productions S → A/ B
    A → 0A/ ε
    B → 0B/ 1B/ ε
    Find the parse tree for the yields (i) 1001 (ii) 00101
13. Construct the Grammar with the productions
    E → E+E
    E→ id Check whether the yield id + id + id is having the parse tree with root E or not.
14. What is ambiguous and unambiguous Grammar?
15. Show that E→E+E/E*E/(E) / id is ambiguous.
16. S→aS/ aSbS/ ε is ambiguous and find the un ambiguous grammar.
17. Define the Instantaneous Descriptions (ID)

18. List out the applications of the pumping lemma for CFG.
19. State the pumping lemma for context-free languages.
20. Use the CFL pumping lemma to show each of these languages not to be context-free
    $\{ a^i b^j c^k | i<j<k\}$

PART-B

1. a) If L is Context free language then prove that there exists PDA M such that L=N(M).
   b) Explain different types of acceptance of a PDA.Are they equivalent in sense of language acceptance? Justify your answer.
2. Construct a PDA accepting $\{a^n b^m a^n / m, n>=1\}$ by empty stack. Also construct the corresponding context-free grammar accepting the same set.

3. a) Prove that L is L(M2) for some PDA M2 if and only if L is N(M1) for some PDA M.
   b) Define Deterministic Push Down Automata DPDA. Is it true that DPDA and PDA are equivalent in the sense of language acceptance is concern? Justify Your answer.
   c) Define a PDA. Give an Example for a language accepted byPDA by empty stack.
4. a) If L is Context free language then prove that there exists PDA M such that L=N(M).
   b) Explain different types of acceptance of a PDA. Are they equivalent in sense of language acceptance? Justify your answer
5. a) Construct the grammar for the following PDA.
   M=({q0, q1},{0,1},{X,z0},δ,q0,Z0,Φ) and where δis given by
   δ(q0,0,z0)={(q0,XZ0)}, δ(q0,0,X)={(q0,XX)},δ(q0,1,X)={(q1, ε)},
   δ(q1,1,X)={(q1, ε)},δ(q1, ε,X)={(q1, ε)}, δ(q1, ε, Z0)={(q1, ε)}. (12)
   b) Prove that if L is N(M1) for some PDA M1 then L is L(M2) for some PDA M2.
6. a) Construct a PDA that recognizes the language
   $\{a^i b^j c^k | i,j,k>0 \text{ and } i=j \text{ or } i=k\}$.
   b) Discuss about PDA acceptance
   1)From empty Stack to final state.
   2)From Final state to Empty Stack.
7. a) Show that E->E+E/E*E/(E)/id is ambiguous. (6)
   b) Construct a Context free grammar G which accepts N(M), where M=({q0, q1},{a,b},{z0,z},δ,q0,z0,Φ) and where δis given by
   δ(q0,b,z0)={(q0,zz0)}
   δ(q0, ε,z0)={(q0, ε)}
   δ(q0,b,z)={(q0,zz)}
   δ(q0,a,z)={(q1,z)}
   δ(q1,b,z)={(q1, ε)
   δ(q1,a,z0)={(q0,z0)}

UNIT IV TURING MACHINES

PART-A

1. Define a Turing Machine.
2. Define multi tape Turing Machine.
3. Explain the Basic Turing Machine model and explain in one move. What are the actions take place in TM?
4. Explain how a Turing Machine can be regarded as a computing device to compute integer functions.
5. Describe the non deterministic Turing Machine model. Is it true the non deterministic
6. Turing Machine models are more powerful than the basic Turing Machines? (In the sense of language Acceptance).
7. Explain the multi tape Turing Machine mode. Is it more power than the basic turing machine? Justify your answer.
8. Using Pumping lemma Show that the language L={ $a^n b^n c^n$ |n>=1} is not a CFL.
9. What is meant by a Turing Machine with two way infinite tape.
10. Define instantaneous description of a Turing Machine.
11. What is the class of language for which the TM has both accepting and rejecting configuration? Can this be called a Context free Language?
12. The binary equivalent of a positive integer is stored in a tape. Write the necessary transition to multiply that integer by 2.
13. What is the role of checking off symbols in a Turing Machine?
14. Mention any two problems which can only be solved by TM.
15. Draw a transition diagram for a Turing machine to compute n mod 2.
16. Difference between multi head and multi tape Turing machine.
17. Define Halting Problem.
18. Define LBA.
19. List out the hierarchy summarized in the Chomsky hierarchy.
20. Draw a transition diagram for a Turing machine accepting of the following languages.

PART-B

1. Explain in detail notes on Turing Machine with example.
2. Consider the language L={a,b}*{aba}{a,b}*={x ε{a,b}*|x containing the substring aba}. L is the regular language, and we can draw an FA recognizing L.
3. Design a Turing Machine M to implement the function “multiplication” using the subroutine „copy‟.
4. Explain how a Turing Machine with the multiple tracks of the tape can be used to determine the given number is prime or not.
5. Design a Turing Machine to compute f(m+n)=m+n, V m,n>=0 and simulate their action on the input 0100.
6. Define Turing machine for computing f(m, n)=m-n (proper subtraction).
7. Explain how the multiple tracks in a Turing Machine can be used for testing given positive integer is a prime or not.
8. Explain in detail” The Turing Machine as a Computer of integer functions”.
9. Design a Turing Machine to accept the language L={0^n 1^n/n>=1}
10. What is the role of checking off symbols in a Turing Machine?
11. Construct a Turing Machine that recognizes the language {wcw / w € {a, b} + }
12. Design a TM with no more than three states that accepts the language. a(a+b) *.Assume €={a,b}
13. Design a TM to implement the function f(x)= x+1.
14. Design a TM to accept the set of all strings {0,1} with 010 as substring.
15. Design a TM to accept the language LE={$a^n b^n c^n$ | n > 1 }

UNIT V UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS

PART-A

1. When a recursively enumerable language is said to be recursive.
2. Is it true that the language accepted by a non deterministic Turing Machine is different from recursively enumerable language?
3. When we say a problem is decidable? Give an example of undecidable problem?
4. Give two properties of recursively enumerable sets which are undecidable.
5. Is it true that complement of a recursive language is recursive? Justify your answer.
6. When a language is said to be recursive or recursively enumerable?
7. When a language is said to be recursive? Is it true that every regular set is not recursive?
8. State the Language NSA and SA.
9. What do you mean by universal Turing Machine?
10. Show that the union of recursive language is recursive.
11. Show that the union of two recursively enumerable languages is recursively enumerable.
12. What is undecidability problem?
13. Show that the following problem is undecidable.“Given two CFG‟s G1 and G2, is L(G1)∩L(G2)=Φ?”.
14. Define recursively enumerable language.
15. Give an example for a non recursively enumerable language.
16. Differentiate between recursive and recursively enumerable languages.
17. Mention any two undecidability properties for recursively enumerable language.
18. Difference between Initial and composition function .
19. Give an example for an undecidable problem.
20. Define MPCP.

PART-B

1. Describe the recursively Enumerable Language with example.
2. Explain in detail notes on computable functions with suitable example.
3. Explain in detail notes on primitive recursive functions with examples.
4. Discuss in detail notes on Enumerating a Language with example
5. Explain in detail notes on universal Turing machines with example.
6. Discuss the Measuring and Classifying Complexity.
7. Describe the Tractable and possibly intractable problems P and NP Completeness.
8. Explain in detail Time and Space Computing of a Turing Machine.