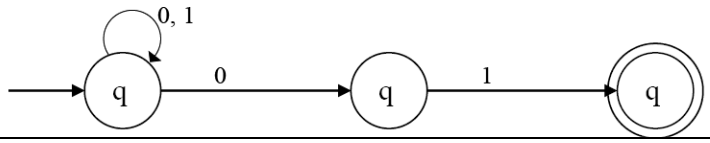
Department : **CSE**

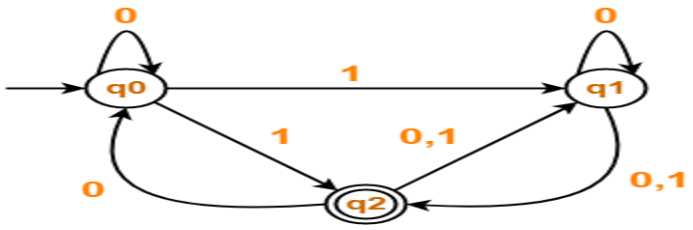
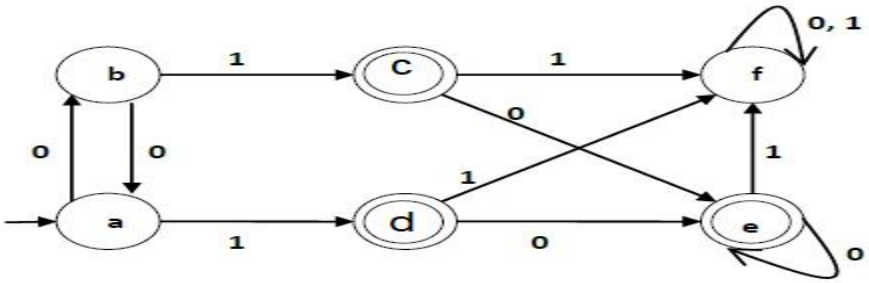
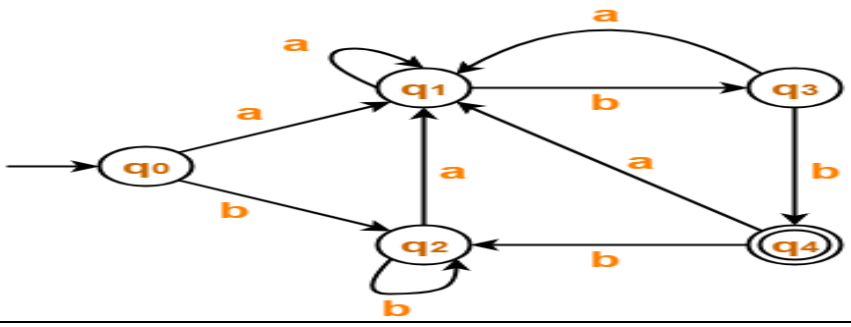
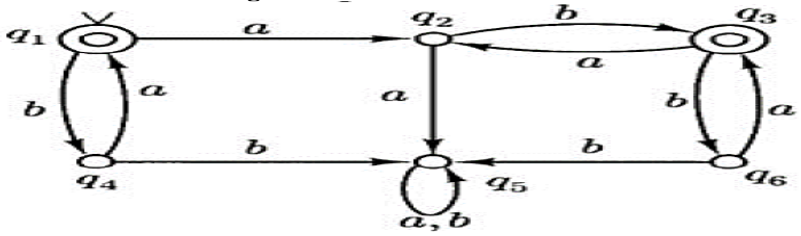
Year & Semester : **III Year V semester**

Sub Code & Sub Name : **23CSE351T, Automata Theory and Compiler Design**

Unit-I

S.No	Part-A Questions
1.	Differentiate DFA and NFA?
2.	Give the mathematical definition of Finite Automata
3.	List out the tuples of NFA
4.	Define Language accepted by DFA.
5.	Define the Regular expression with an example
6.	Write the language for the given regular expression r = (aa)*(bb)*b
7.	Describe kleen closuer operation in regular expression.
8.	Write Regular Expression for the set of strings over {0.1} that have at least one.
9.	Describe the following Regular expression in English (1+10)*
10.	Describe Chomsky hierarchy of Languages.
11.	Define Left Linear Grammar with a relevant example
12.	Define Right Linear Grammar with a relevant example
13.	Define Regular grammar.
14.	State the pumping lemma for regular languages.
15.	List any four closure properties of regular sets

S.No	Part-B Questions
1.	a. Design FA which accepts odd number of a's and even number of b's b. Construct DFA equivalent to the given NFA 
2.	Construct a DFA that accepts the following L={ x € {a,b}:|x|a = even and |x|b = odd}.
3.	Construct a DFA that accepts the following Binary strings such that the third symbol from the right end is 1.
4.	Construct DFA for the following: A) L={W/W has both an even numbers of 0's and even number of 1's} B) L={W/W is in the form of 'x01y' for some strings x and y consisting of 0's and 1's}
5.	Construct a DFA that accepts the following **i.** Set of all strings with three consecutive 0's **ii.** Set of all strings ending with aba
6.	Explain the procedure for converting NFA to DFA.
7.	Construct DFA equivalent to the given NFA

	
8.	Construct a ε-NFA for the following regular expression. (0+1)*(00+11)(0+1)*
9.	Construct a ε-NFA for the following regular expression. ((01+10 *) 00)*
10.	Construct the NFA with ϵ for the regular expression R = 01[((10)*+111)* + 0]*1
11.	Construct the NFA with ϵ for the regular expression (i) 0(0+1) * 100 and (ii). a(a + b)*b
12.	Define Regular Expression? Explain about the Properties of Regular Expressions?
13.	Minimize the following DFA 
14.	Minimize the following DFA 
15.	Minimize the following DFA 

Unit-II

S.No	Part-A Questions
1.	Identify the language generated by the following grammar G. G = ({S}, {0,1}, {S-> 0 / 1 / ϵ / 0S0 / 1S1}, S)
2.	List out the tuples of CFG
3.	Define Context Free Grammar
4.	Define ambiguity in CFG with an example.
5.	Define a derivation tree.
6.	Find Context Free Language for S→aSb|ab
7.	What are the unit productions.
8.	Describe Nullable variables.
9.	Define Push Down Automata
10.	A PDA is more powerful than a finite automaton. Justify them.
11.	List out the tuples of PDA
12.	Define Instantaneous description (ID) in PDA.
13.	Mention the two methods of accepting a language in PDA
14.	Write the formal definition of Deterministic PDA
15.	What is the additional feature of PDA, when compared with NFA?

S.No	Part-B Questions
1.	Consider the Grammar G whose productions are S -> 0B / 1A A -> 0 / 0S / 1AA B -> 1 / 1S / 0BB and the string 0110 a. Find the left most derivation and associated derivation tree. b. Find the right most derivation and associated derivation tree. c. Show that the G is ambiguous.
2.	Construct a CFG to generate the set of all palindromes over the alphabet{a,b}
3.	Define Ambiguous Grammar. Check whether the grammar S->aAB, A->bC/cd,C->cd, B->c/d is Ambiguous or not?
4.	Derive left and right most derivations for the input string a=b*c+d/e for the given Grammar. E->E+E|E-E|E*E E->E/E E->(E)|id
5.	a) Explain about Ambiguity in Grammars and Languages with example. b) Discuss in detail about leftmost and right most derivation tree with example.
6.	Remove null, unit productions and minimize the following CFG S->AaB, A->D, B-> bbA/ϵ, D->E, E->F, F->as
7.	Minimize the following CFG S->AB, A->a, B->C, B->b, C-> D, D->E
8.	Minimize the following CFG S->a/Ab/aBa A->b/ϵ B->b/A
9.	Construct a PDA that accept {WCW^R | W in (0+1)* } by empty stack
10.	Construct a PDA that accept {WW^R | W in (0+1)* } by empty stack
11.	Construct a PDA that accept {$a^n b^n$ | n>=1 } by empty stack

12.	Construct a PDA that accept $\{a^n b^{2n} \mid n>=1\}$ by empty stack
13.	Construct a PDA that will accept the language generated by the grammar G = ({S, A}, {a, b}, P, S) with the productions S ->AA / a, A ->SA / b and test whether "abbabb" is in N(M).
14.	Construct a PDA that will accept the language generated by the grammar G = ({S, A}, {a, b}, P, S) with the productions S ->0BB, B ->0S/1S/0 and test whether "010^4" is in N(M).
15.	Construct PDA from the following Grammar S-> aB; B-> bA/b; A-> aB

Unit-III

S.No	Part-A Questions
1.	Define Turing machine
2.	Describe the Turing machine model.
3.	Define ID of Turing machine.
4.	Define string acceptance.
5.	Define language acceptance.
6.	Describe Turing machine left move with example.
7.	Describe Turing machine right move with example.
8.	Define a compiler.
9.	Differentiate Compiler and Interpreter
10.	What are the supporting Phases of compiler.
11.	List the phases of compiler.
12.	Differentiate analysis and synthesis phase.
13.	What are the types of input buffering Schemes
14.	List the data structures for implementing symbol table.
15.	Why lexical and syntax analyzers are separated?

S.No	Part-B Questions
1.	Design a TM to recognize the language L ={ww^R ; w is (a+b)*}
2.	Design a TM to recognize the language L ={wcw^R ; w is (a+b)*}
3.	Design a TM to recognize the language L ={a^nb^n/n>=1}
4.	Construct a Turing Machine that will accept the Language consists of all palindromes of 0's and 1's?
5.	What are the different phases of compiler? Explain the phases in detail. Write down the output of each phase for the expression position : = initial + rate * 60.
6.	List and explain in detail about different phases of compilation with an example.
7.	Explain the Translation process with an example.
8.	Explain in detail about input buffering.
9.	Explain the role of lexical analyzer in detail.

Unit-IV

S.No	Part-A Questions
1.	Write about parser.
2.	What are the types of parser.
3.	List top down parsers.
4.	List bottom up parsers.
5.	Write the algorithm for FIRST in parser.
6.	Mention the functions in LR Parsers?
7.	Differentiate SLR and LALR?
8.	What is LL(1) grammar
9.	Write the algorithm for FIRST in parser.
10.	**What is Shift/Reduce Conflict**
11.	**What is Reduce/Reduce Conflict**
12.	**What are the Conflicts During Shift-Reduce Parsing**

S.No	Part-B Questions
1.	Find FIRST and FOLLOW for the following grammer. S->ACB|Ba|Cd A->et|ε B->gh|d c->d|e| ε
2.	Construct Predictive parsing table for the bellow given grammer. S->iEtS|iEtSeS|a E->b
3.	Check whether the following grammar if LL(1) grammar. S→ iEtS / iEtSeS /a E→ b
4.	Construct a predictive parsing table for the grammar E→E+T|T T→T*F|F F→E|id
5.	Construct SLR Parsing table for the grammar S→L=R/R, L→*R/id, R→L
6.	Construct the SLR parsing table for the bellow given grammer. E→E+T|T T→T*F|F F→id
7.	Consider the following grammar S→ AS| b A→ SA |a Construct the SLR parse table for the grammar. Show the action of parser for the input string “abab”
8.	Construct CLR parser for the following grammer. S→AaAb|BbBa A->ε B→ε

9.	Construct LALR parser for the following grammer. $S \rightarrow L=R$ $S \rightarrow R$ $L \rightarrow *R$ $L \rightarrow id$ $R \rightarrow L$
10.	Translate the expression a+a*(b-c)+(b-c)*d into quadruples, triples and indirect triples.
11.	What is a three address code? Mention its types. How would you implement these address statements? Explain with suitable example.
12.	What is an intermediate code? Explain different types of intermediate codes forms and represent the following statement in different forms: W = (A + B) – (C + D) + (A + B + C).

Unit-V

S.No	Part-A Questions
1.	What is DAG?
2.	List the issues in designing a code generator?
3.	What is code motion?
4.	Give example of redundant sub expression elimination.
5.	Write the rules for finding Leaders while constructing basic blocks?
6.	What is dead code?
7.	What is common sub expression elimination?
8.	What is strength reduction? Give an example?
9.	Discuss about Instruction Selection and Register allocation.
10.	What is the use of DAG?

S.No	Part-B Questions
1.	Discuss the various Peephole Optimization techniques in detail.
2.	What are the basic blocks and flow graphs? write an algorithm to partition the three address instructions into basic blocks.
3.	Explain the brief about different principal sources of Optimization with suitable examples.
4.	Explain the principle sources of code optimization in detail.
5.	Discuss about the following: a) Copy Propagation b) Dead code Elimination and c) Code motion.
6.	What are the object code forms? Explain the issues in code generation
7.	Discuss various techniques of Structure preserving transformations for code optimization
8.	Explain in brief about the issues in the design of code generator.
9.	Explain the simple code generator algorithm, with an example.