

# **SRM VALLIAMMAI ENGINEERING COLLEGE**

(An Autonomous Institution)

**SRM Nagar, Kattankulathur–603203**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**QUESTION BANK**



**IV SEMESTER**

**CS3461-Theory of Computation**

**Regulation–2023**

**Academic Year 2025–26(Even Sem)**

*Prepared by*

**Ms. N. Leo Bright Tennisson, Assistant Professor (Sr. G) /CSE**

**Ms. G. Sathya, Assistant Professor (O.G) / CSE**

**Ms. M. Priyadharshini, Assistant Professor (O.G)/CSE**



# SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)  
SRM Nagar, Kattankulathur-603203.



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### QUESTION BANK

**SUBJECT: CS3461-Theory of Computation**

**SEM/ YEAR: IV/ II**

#### UNIT I AUTOMATA FUNDAMENTALS

Introduction - Languages: Alphabets and Strings - Finite Automata - Deterministic Finite Automata - Non-deterministic Finite Automata – Equivalence of NFA and DFA – Finite Automata with Epsilon Transitions.

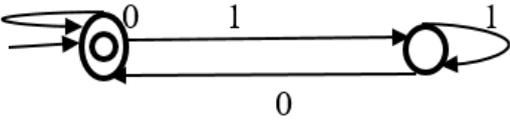
#### PART- A

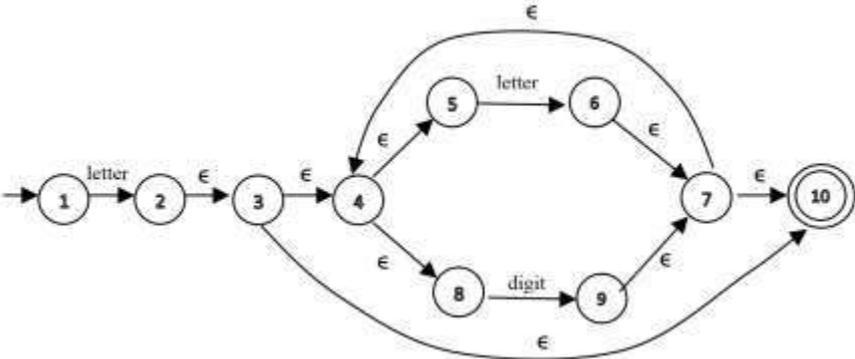
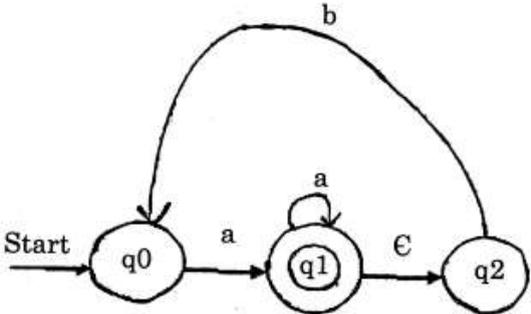
Q.No	Questions	BT Level	Competence
1.	Distinguish DFA and NFA.	BTL2	Understand
2.	Define DFA.	BTL1	Remember
3.	Write the notations of DFA.	BTL1	Remember
4.	Identify NFA- store present $a^*b c$ .	BTL2	Understand
5.	Consider the String $X=110$ and $y=0110$ . Find i) $XY$ ii) $X^2$ iii) $YX$ iv) $Y^2$	BTL1	Remember
6.	Define the following language over the input set $\Sigma=\{a,b\}$ , $L=\{a^n b^m   n,m \geq 0\}$ .	BTL2	Understand
7.	Define a non-deterministic finite automaton and the applications of automata theory.	BTL1	Remember
8.	Define a NFA which accepts the set of all strings that start with zero.	BTL1	Remember
9.	What are the applications of automata theory?	BTL1	Remember
10.	Define an identifier with a transition diagram (automata).	BTL2	Understand
11.	Define $\epsilon$ -NFA.	BTL1	Remember
12.	Define the significance of DFA.	BTL1	Remember
13.	Give the Non-deterministic automata to accept strings containing the substring 0101.	BTL2	Understand
14.	Explain that for any language L accepted by an NFA, there exists an equivalent DFA that also accepts L.	BTL2	Understand
15.	Define the term epsilon transition.	BTL1	Remember
16.	Define the extended transition function for a $\epsilon$ -NFA.	BTL1	Remember

17.	Give a FA which accepts the only input 101 over the input set: $Z=\{0,1\}$	BTL2	Understand
18.	Define a Finite automaton and give its types.	BTL1	Remember
19.	Give a DFA of strings which accepts string either 01 or 10 over $\{0, 1\}$ .	BTL2	Understand
20.	Give a Finite Automaton to check whether a given binary number is even.	BTL2	Understand
21.	Give the NFA which accepts the set of all strings that end with zero.	BTL2	Understand
22.	Explain a DFA that accepts strings over $\Sigma = \{0,1\}$ containing three consecutive zeros.	BTL2	Understand
23.	Explain an NFA that accepts all strings starting with "10".	BTL2	Understand
24.	Define alphabets and strings.	BTL1	Remember

**PART-B**

1.	<p>(i) Explain if L is accepted by an NFA with <math>\epsilon</math>-transition then show that L is accepted by an NFA without <math>\epsilon</math>-transition. (8)</p> <p>(ii) Construct a DFA equivalent to the NFA. <math>M = (\{p, q, r\}, \{0,1\}, \delta, p, \{q, s\})</math> Where <math>\delta</math> is defined in the following table. (8)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th><math>\delta</math></th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>p</td> <td>{q,s}</td> <td>{q}</td> </tr> <tr> <td>q</td> <td>{r}</td> <td>{q,r}</td> </tr> <tr> <td>r</td> <td>{s}</td> <td>{p}</td> </tr> <tr> <td>s</td> <td>-</td> <td>{p}</td> </tr> </tbody> </table>	$\delta$	0	1	p	{q,s}	{q}	q	{r}	{q,r}	r	{s}	{p}	s	-	{p}	BTL5	Evaluate
$\delta$	0	1																
p	{q,s}	{q}																
q	{r}	{q,r}																
r	{s}	{p}																
s	-	{p}																
2.	<p>(i) Design a DFA that recognizes the set of all strings on <math>\Sigma = \{a,b\}</math> starting with the 'prefix ab' and test using the input string. (8)</p> <p>(ii) Draw a transition diagram for a DFA that accepts the string abaa and no other strings and test using the input string. (8)</p>	BTL6	Create															
3.	Show, with a suitable construction, that for any language L accepted by an NFA, there exists an equivalent DFA that also accepts L. (16)	BTL3	Apply															
4.	Illustrate a non-deterministic finite automata accepting the set of strings in $(0+1)^*$ such that two 0's are separated by a string whose length is $4i$ , for some $i \geq 0$ . (16)	BTL3	Apply															
5.	<p>Analyze the relationship between NFA and DFA by showing that any language accepted by an NFA can also be accepted by a DFA. (16)</p>	BTL4	Analyze															

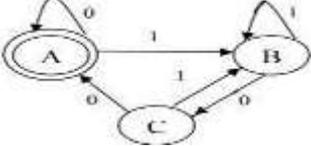
6.	<p>(i) Compose that a language L is accepted by some <math>\epsilon</math>-NFA if and only if L is accepted by some DFA. (8)</p> <p>(ii) Consider the following <math>\epsilon</math>-NFA. Compute the <math>\epsilon</math>-closure of each state and find its equivalent DFA. (8)</p> <table border="1" data-bbox="384 405 866 595"> <thead> <tr> <th><math>\delta</math></th> <th><math>\epsilon</math></th> <th>a</th> <th>b</th> <th>C</th> </tr> </thead> <tbody> <tr> <td><math>\rightarrow p</math></td> <td><math>\phi</math></td> <td>{p}</td> <td>{q}</td> <td>{r}</td> </tr> <tr> <td>q</td> <td>{p}</td> <td>{q}</td> <td>{r}</td> <td><math>\Phi</math></td> </tr> <tr> <td>*r</td> <td>{q}</td> <td>{r}</td> <td><math>\phi</math></td> <td>{p}</td> </tr> </tbody> </table>	$\delta$	$\epsilon$	a	b	C	$\rightarrow p$	$\phi$	{p}	{q}	{r}	q	{p}	{q}	{r}	$\Phi$	*r	{q}	{r}	$\phi$	{p}	BTL6	Create
$\delta$	$\epsilon$	a	b	C																			
$\rightarrow p$	$\phi$	{p}	{q}	{r}																			
q	{p}	{q}	{r}	$\Phi$																			
*r	{q}	{r}	$\phi$	{p}																			
7.	<p>(i) Demonstrate how a language L is accepted by some DFA if L is accepted by some NFA. (8)</p> <p>(ii) Convert the following NFA to its equivalent DFA (8)</p> <table border="1" data-bbox="373 741 904 954"> <thead> <tr> <th><math>\delta</math></th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td><math>\rightarrow p</math></td> <td>{p, q}</td> <td>{p}</td> </tr> <tr> <td>q</td> <td>{r}</td> <td>{r}</td> </tr> <tr> <td>r</td> <td>{s}</td> <td><math>\Phi</math></td> </tr> <tr> <td>*s</td> <td>{s}</td> <td>{s}</td> </tr> </tbody> </table>	$\delta$	0	1	$\rightarrow p$	{p, q}	{p}	q	{r}	{r}	r	{s}	$\Phi$	*s	{s}	{s}	BTL3	Apply					
$\delta$	0	1																					
$\rightarrow p$	{p, q}	{p}																					
q	{r}	{r}																					
r	{s}	$\Phi$																					
*s	{s}	{s}																					
8.	<p>(i) Design a DFA to recognize strings containing an odd number of 1's and an even number of 0's. (8)</p> <p>(ii) Design a DFA over {a, b} that accepts strings containing not more than three a's. (8)</p>	BTL6	Create																				
9.	<p>(i) Point out the steps in conversion of NFA to DFA and for the following convert NFA to a DFA: (8)</p> <table border="1" data-bbox="507 1267 767 1447"> <thead> <tr> <th><math>\delta</math></th> <th>a</th> <th>b</th> </tr> </thead> <tbody> <tr> <td>p</td> <td>{p}</td> <td>{p, q}</td> </tr> <tr> <td>q</td> <td>{r}</td> <td>{r}</td> </tr> <tr> <td>r</td> <td><math>\{\phi\}</math></td> <td><math>\{\phi\}</math></td> </tr> </tbody> </table> <p>(ii) Infer the language for the following (8)</p> 	$\delta$	a	b	p	{p}	{p, q}	q	{r}	{r}	r	$\{\phi\}$	$\{\phi\}$	BTL4	Analyze								
$\delta$	a	b																					
p	{p}	{p, q}																					
q	{r}	{r}																					
r	$\{\phi\}$	$\{\phi\}$																					
10.	<p>Design a DFA from the given NFA. (16)</p> <p><math>M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})</math> where <math>\delta</math> is given by</p> <p><math>\delta(q_0, 0) = \{q_0, q_1\}</math>, <math>\delta(q_0, 1) = \{q_1\}</math>, <math>\delta(q_1, 0) = \phi</math>, <math>\delta(q_1, 1) = \{q_0, q_1\}</math></p>	BTL6	Create																				

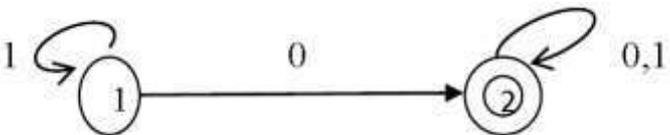
11.	Differentiate the NFA and DFA. Deduce the following $\epsilon$ -NFA to DFA. (16) <table border="1" data-bbox="531 309 994 510" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>states</th> <th><math>\epsilon</math></th> <th>a</th> <th>b</th> <th>C</th> </tr> </thead> <tbody> <tr> <td><math>\rightarrow p</math></td> <td>{q,r}</td> <td><math>\phi</math></td> <td>{q}</td> <td>{q,r}</td> </tr> <tr> <td>q</td> <td><math>\phi</math></td> <td>{p}</td> <td>{r}</td> <td>{q}</td> </tr> <tr> <td>*r</td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td>{r}</td> </tr> </tbody> </table>	states	$\epsilon$	a	b	C	$\rightarrow p$	{q,r}	$\phi$	{q}	{q,r}	q	$\phi$	{p}	{r}	{q}	*r	$\phi$	$\phi$	$\phi$	{r}	BTL4	Analyze
states	$\epsilon$	a	b	C																			
$\rightarrow p$	{q,r}	$\phi$	{q}	{q,r}																			
q	$\phi$	{p}	{r}	{q}																			
*r	$\phi$	$\phi$	$\phi$	{r}																			
12.	(i).Design the extended transition function for NFA, DFA and $\epsilon$ -NFA. (8) (ii) Consider the following $\epsilon$ -NFA for an identifier. Consider the $\epsilon$ -closure of each state and give its equivalent DFA. (8) 	BTL3	Apply																				
13.	Given $\Sigma = \{a, b\}$ Analyze and construct a DFA which recognize the language $L = \{b^m a b^n : m, n > 0\}$ . (16)	BTL6	Create																				
14.	Analyze the differences between an NFA and a DFA, and convert the given $\epsilon$ -NFA into an equivalent DFA. (16)	BTL4	Analyze																				
15.	Convert the following $\epsilon$ -NFA to DFA and analyze the difference between NFA and DFA. (16) 	BTL4	Analyze																				
16.	Solve the NFA that accepts all strings that ends in 01. Give its transition table and the extended transition function for the input string 0101. Also construct a DFA for the above NFA using subset construction method. (16)	BTL6	Create																				
17.	(i) Point out a DFA which accepts the substring 1010 and prove with the input string. (8) (ii) Analyze a DFA that accept the string $\{0, 1\}$ that always ends with 00. (8)	BTL4	Analyze																				

**UNIT II**  
**REGULAR EXPRESSION AND LANGUAGES**

Regular Expressions – FA and Regular Expressions – Proving Languages not to be regular – Closure Properties of Regular Languages–Equivalence and Minimization of Automata.

**PART-A**

Q.No	Questions	BT Level	Competence
1.	List the operators of Regular Expressions.	BTL1	Remember
2.	Distinguish between regular expression and regular.	BTL2	Understand
3.	Explain and give the regular expression for the language $L_1$ , which contains all strings over $\{0,1\}$ ending with 00.	BTL2	Understand
4.	What are the closure properties of regular languages?	BTL1	Remember
5.	Describe a finite automaton for the regular expression $0^*1^*$ .	BTL1	Remember
6.	Identify a regular expression for the set of all the strings.	BTL1	Remember
7.	Explain and give a regular expression for the set of all strings containing an odd number of 1's, e.g., R.E = $1(0+11)^*$ .	BTL2	Understand
8.	Explain the difference between the + closure and *closure.	BTL2	Understand
9.	Describe a regular expression for the set of all strings of 0's.	BTL2	Understand
10.	What is the Closure property of regular set S.?	BTL1	Remember
11.	Identify regular expression corresponding to the state diagram: 	BTL2	Understand
12.	Explain the language generated by the regular expression $(0+1)^*$ .	BTL2	Understand
13.	Name the four closure properties of RE.	BTL1	Remember
14.	Explain whether the language accepted by any NFA is different from a regular language and justify your answer.	BTL2	Understand
15.	Explain why the complement of a regular language is also regular.	BTL2	Understand
16.	Define a DFA for the regular expression $aa^*bb^*$ .	BTL1	Remember
17.	List the precedence of regular expression operators.	BTL1	Remember
18.	Explain and write a regular expression for the language over the set $\Sigma = \{a, b\}$ in which the total number of a's is divisible by 3.	BTL2	Understand
19.	Define RE.	BTL1	Remember
20.	Explain and write a regular expression to describe an identifier and a positive integer.	BTL2	Understand
21.	Express a RE for the language containing of all the strings of any number of a's and b's.	BTL2	Understand
22.	Explain arden's theorem.	BTL2	Understand
23.	Explain about the equivalence of two automata?	BTL2	Understand
24.	Define the operations on regular language.	BTL1	Remember
<b>PART-B</b>			
1.	Demonstrate how the set $L = \{ab^n/n \geq 1\}$ is not a regular. (16)	BTL3	Apply

2.	Analyze the regular expression “a(a+b)a” by converting it into an $\epsilon$ -NFA and determine the minimal state DFA. (16)	BTL4	Analyze
3.	Examine whether the language $L = \{0^n 1^n \mid n \geq 1\}$ is regular or not, and justify your reasoning using appropriate methods. (16)	BTL4	Analyze
4.	(i) Analyze and explain a regular expression, and derive a regular expression for the set of strings consisting of alternating 0's and 1's. (8) (ii) Analyze the regular expression $(ab+a)^*$ and construct an equivalent finite automaton, explaining the state transitions.* (8)	BTL4	Analyze
5.	(i) Analyze the closure properties of regular languages. (8) (ii) Analyze NFA with epsilon for the RE = $(a/b)^*ab$ and convert it in to DFA and further find the minimized DFA. (8)	BTL4	Analyze
6.	Show that the following languages are not regular. (i) $\{w \in \{a,b\}^* \text{ such that } w=ww^R\}$ . (8) (ii) Set of strings of 0's and 1's, beginning with a 1, whose value treated as a binary number is a prime. (8)	BTL3	Apply
7.	Verify whether $L = \{a^{2^n} \mid n \geq 1\}$ is regular or not. (16)	BTL3	Apply
8.	(i) Prove the reverse of a regular language is regular. (8) (ii) A homomorphism of regular language is regular. (8)	BTL5	Evaluate
9.	Set the algorithm for minimization of a DFA. Develop a minimized DFA for the RE $(a+b)(a+b)^*$ and trace for the string baaaab. (16)	BTL6	Create
10	i) Prove that any language accepted by a DFA can be represented by a regular expression. (8) ii) Construct a finite automata for the regular expression $10+(0+11)0^*1$ . (8)	BTL6	Create
11	Analyze the DFA Minimization algorithm with an Example (16)	BTL4	Analyze
12	Demonstrate how the set $L = \{a^n b^m \mid m, n \geq 1\}$ is not a regular. (16)	BTL5	Evaluate
13.	(i) Deduce into regular expression that denotes the language accepted by following DFA. (16) 	BTL4	Analyze
14	(i) Analyze and prove that the $L_1$ and $L_2$ are two languages then $L_1 \cup L_2$ is regular. (8) (ii) Analyze and prove that the $L_1$ and $L_2$ are two languages then $L_1 \cap L_2$ is regular (8)	BTL4	Analyze
15.	(i) Explain regular expression. (8) (ii) Explain in detail about the closure properties of regular language. (8)	BTL5	Evaluate

16.	Solve the following to a regular expression. (16)	BTL3	Apply
17.	Evaluate a minimized DFA for the RE $10+(0+11)0^*1$ (16)	BTL5	Evaluate

### UNIT III

### CONTEXT FREE GRAMMAR AND LANGUAGES

CFG - Parse Trees - Ambiguity in Grammars and Languages - Normal Forms for CFG - Definition of the Pushdown Automata – Languages of a Pushdown Automata - Equivalence of Pushdown Automata and CFG - Pumping Lemma for CFL

### PART-A

Q.No	Questions	BT Level	Competence
1.	Express the ways of languages accepted by PDA and define them?	BTL2	Understand
2.	Summarize PDA . Convert the following CFG to PDA $S \rightarrow aAA, A \rightarrow aS bS a$ .	BTL2	Understand
3.	Define ambiguous grammar and CFG.	BTL1	Remember
4.	Define parse tree and derivation.	BTL1	Remember
5.	Describe the context free Grammar representing the set of Palindrome over $(0+1)^*$	BTL2	Understand
6.	Distinguish Deterministic and Non deterministic PDA. Is it true that non deterministic PDA is more powerful than that of deterministic PDA? Justify your answer.	BTL2	Understand
7.	When PDA is said to be deterministic?	BTL1	Remember
8.	Explain the language $L(G)$ generated by the grammar $G$ with variables $S,A,B$ terminals $a,b$ and productions. $S \rightarrow aB, B \rightarrow b, B \rightarrow bA, A \rightarrow aB$ .	BTL2	Understand
9.	Summarize the procedure for converting CNF to GNF with an example.	BTL2	Understand
10.	Define equivalence of PDA and CFG.	BTL1	Remember
11.	Explain the languages generated by a PDA using final state of the PDA and empty stack of that PDA.	BTL2	Understand
12.	Define the rule for construction of CFG from given PDA.	BTL1	Remember
13.	Explain a CFG for the language of palindrome string over $\{a,b\}$ . Write the CFG for the language, $L=(a^n b^n   n \geq 1)$ .	BTL2	Understand
14.	Define GNF.	BTL1	Remember
15.	Define that $L=\{a^p   P \text{ is prime}\}$ is not context free.	BTL1	Remember

16.	Explain the CFG for the set of strings that contains equal number of a's and b's over $\Sigma = \{a,b\}$ .	BTL2	Understand
17.	Define the pumping Lemma for CFLs.	BTL1	Remember
18.	Explain the right most derivation (id+id*id) for using the grammar and also state whether a given grammar is ambiguous one or not. $E \rightarrow E+E/E*E/(E)/id$	BTL2	Understand
19.	Define the additional features a PDA has when compared with NFA.	BTL1	Remember
20.	Define parse tree for the grammar $S \rightarrow aS \mid aSbS \mid \epsilon$ . This grammar is ambiguous. Show that the string aab has two parse trees.	BTL1	Remember
21.	Describe the unit and null production detail.	BTL2	Understand
22.	Define the Instantaneous Description (ID) for PDA.	BTL1	Remember
23.	Define the grammar G with the following production. $S \rightarrow Aa, S \rightarrow B, B \rightarrow A, B \rightarrow bb, A \rightarrow a, A \rightarrow bc, A \rightarrow B$ Eliminate all unit production and get an equivalent grammar $G_1$ .	BTL1	Remember
24.	Explain the two different ways to define PDA acceptability.	BTL2	Understand
<b>PART-B</b>			
1.	(i) Illustrate a PDA accepting $L = \{a^n b^{3n} \mid n \geq 1\}$ by empty store. (8) (ii) Illustrate a PDA that accepts $L = \{a^n b^m c^n \mid n, m \geq 1\}$ . (8)	BTL3	Apply
2.	Design and explain the following grammar into equivalent one with no unit production and no useless symbols and convert into CNF. (16) $S \rightarrow A \mid CB$ $A \rightarrow C \mid D$ $B \rightarrow 1B \mid 1$ $C \rightarrow 0C \mid 0$ $D \rightarrow 2D \mid 2$	BTL5	Evaluate
3.	(i) Identify that deterministic PDA is less powerful than non-deterministic PDA. (8) (ii) Construct a PDA accepting $\{a^n b^m a^n \mid m, n \geq 1\}$ by empty stack. Also tell the corresponding context-free grammar accepting the same set. (8)	BTL6	Create
4.	(i) Construct a parse tree for the following grammar (8) $G = (\{S, A\}, \{a, b\}, P, S)$ where P Consists of $S \rightarrow aAS \mid b$ $A \rightarrow SbA \mid ba$ Draw the derivation tree for the string $w = abbbab$ . (ii) Let $G = (V, T, P, S)$ be a Context Free Grammar then prove that if the recursive inference procedure call tells us that terminal string $W$ is in the language of variable $A$ , then there is a parse tree with a root $A$ and yield $w$ . (8)	BTL6	Create

5.	<p>(i) Define Non Deterministic Push Down Automata. Is it true that DPDA and NDPDA are equivalent in the sense of language acceptance is concern? Justify your answer. (8)</p> <p>(ii) Let <math>M = (\{q_0, q_1\}, \{0, 1\}, \{X, z_0\}, \delta, q_0, z_0, \Phi)</math> where <math>\delta</math> is given by:</p> $\delta(q_0, 0, z_0) = \{(q_0, Xz_0)\}$ $\delta(q_0, 0, X) = \{(q_0, XX)\}$ $\delta(q_0, 1, X) = \{(q_1, \epsilon)\}$ $\delta(q_1, 1, X) = \{(q_1, \epsilon)\}$ $\delta(q_1, \epsilon, X) = \{(q_1, \epsilon)\}$ $\delta(q_1, \epsilon, z_0) = \{(q_1, \epsilon)\}$ <p>Construct a CFG <math>G = (V, T, P, S)</math> generating <math>N(M)</math>. (8)</p>	BTL5	Remember
6.	<p>(i) Define PDA. Show an Example for a language accepted by PDA by empty stack. (8)</p> <p>(ii) Convert the grammar <math>S \rightarrow 0S1   A</math>  <math>A \rightarrow 1A0   S   \epsilon</math> into PDA that accepts the same language by the empty stack . Check whether 0101 belongs to <math>N(M)</math>. (8)</p>	BTL3	Apply
7.	<p>(i) Analyze the theorem: If L is Context free language then prove that there exists PDA M such that <math>L = N(M)</math>. (8)</p> <p>(ii) Prove that if there is PDA that accepts by the final state then there exists an equivalent PDA that accepts by Null State. (8)</p>	BTL4	Analyze
8.	<p>Solve the following grammar</p> $S \rightarrow aB   bA$ $A \rightarrow a   aS   bAA$ $B \rightarrow b   bS   aBB$ <p>for the string  “baaabbbabba”</p> <p>Give</p> <p>i) Leftmost derivation (4)</p> <p>ii) Rightmost derivation (4)</p> <p>iii) Derivation Tree (8)</p>	BTL5	Evaluate
9.	<p>Modify the following grammar G into Greibach Normal Form(GNF) (16)</p> $S \rightarrow AB$ $A \rightarrow BS   b$ $B \rightarrow SA   a$	BTL3	Apply
10.	<p>Construct a PDA that recognizes and analyzes the language <math>\{a^i b^j c^k \mid i, j, k &gt; 0 \text{ and } i=j \text{ or } i=k\}</math> and also explain about PDA acceptance</p> <p>(i) From empty Stack to final state. (8)</p> <p>(ii) From Final state to Empty Stack. (8)</p>	BTL4	Analyze
11.	<p>Suppose <math>L = L(G)</math> for some CFG <math>G = (V, T, P, S)</math>, then prove that <math>L - \{\epsilon\}</math> is <math>L(G')</math> for a CFG <math>G'</math> with no useless symbols or <math>\epsilon</math>-productions. (16)</p>	BTL5	Evaluate
12.	<p>(i) Illustrate the PDA that accept the given CFG (8)</p> $S \rightarrow xaax$ $X \rightarrow ax/bx/\epsilon$ <p>(ii) Express a PDA for the language <math>a^n b^m a^{n+m}</math> (8)</p>	BTL3	Apply

13.	(i) Illustrate a PDA for the language $\{WCWR/W \in \{0,1\}^*\}$ . (8) (ii) Illustrate a CFG for the constructed PDA. (8)	BTL3	Apply
14.	(i) Consider the grammar (8) $S \rightarrow ASB \mid \epsilon$ $A \rightarrow aAS \mid a$ $B \rightarrow SbS \mid A \mid bb$ Are there any useless symbols, $\epsilon$ -production and unit production? Eliminate if so. (ii) Define derivation tree. Explain its uses with an example (8)	BTL4	Analyze
15.	Modify the following grammar G into Greibach Normal Form (GNF) $S \rightarrow XB \mid AA$ $A \rightarrow a \mid SA$ $B \rightarrow b$ $X \rightarrow a$ (16)	BTL3	Apply
16.	(i) Solve a PDA for accepting a language $\{a^n b^{2n} \mid n \geq 1\}$ . (8) (ii) Solve a PDA for accepting a language $\{0^n 1^m 0^n \mid m, n \geq 1\}$ . (8)	BTL3	Apply
17.	Deduce PDA for the given CFG, and test whether $010^4$ is acceptable by this PDA. $S \rightarrow 0BB$ $B \rightarrow 0S \mid 1S \mid 0$ (16)	BTL5	Evaluate

**UNIT IV  
TURING MACHINES**

Turing Machines – Introduction – Formal definition of Turing machines – Instantaneous descriptions – Turing machines as Acceptors – Turing machine as Transducers computable languages and functions - Deterministic TM, Multi-track and Multitape Turing Machine- Programming Techniques for TM.

**PART-A**

Q.No	Questions	BT Level	Competence
1.	Discuss on checking off symbols.	BTL2	Understand
2.	Explain the Basic Turing Machine model.	BTL2	Understand
3.	When do you say a turing machine is an algorithm?	BTL1	Remember
4.	Define universal TM.	BTL1	Remember
5.	Express Turing machine as Transducers.	BTL2	Understand
6.	Define Turing Machine.	BTL1	Remember
7.	Discuss the applications of Turing machine.	BTL2	Understand
8.	Express Turing machines as Acceptors	BTL2	Understand
9.	What is the class of language for which the TM has both accepting and rejecting configuration? Can this be called a Context free Language? Discuss.	BTL2	Understand
10.	Define Instantaneous description of TM.	BTL1	Remember
11.	Explain the special features of TM.	BTL2	Understand
12.	Write the difference between finite automata and Turing machine.	BTL1	Remember
13.	Give a note Deterministic TM	BTL2	Understand

14.	List the Programming Techniques for TM	BTL1	Remember
15.	Draw a transition diagram for a turing machine to identify $n \bmod 2$ .	BTL1	Remember
16.	Express the techniques for TM construction.	BTL2	Understand
17.	Give short notes on two-way infinite tape TM.	BTL2	Understand
18.	Distinguish TM and PDA.	BTL2	Understand
19.	Indicate the role of checking off symbols in a Turing Machine.	BTL2	Understand
20.	Describe the basic difference between 2-way FA and TM.	BTL2	Understand
21.	Describe the language accepted by TM.	BTL1	Remember
22.	Outline the various representation of TM.	BTL1	Remember
23.	Explain the situation before and after the move caused by the transition of TM.	BTL2	Understand
24.	Give a TM for a successor function for a given unary number $f(n)=n+1$ .	BTL2	Understand

**PART-B**

1.	Illustrate the Turing machine for computing $f(m, n) = m - n$ (proper subtraction). (16)	BTL3	Apply
2.	Construct a turing machine that estimate unary multiplication (Say $000 \times 00 = 000000$ ). (16)	BTL6	Create
3.	Construct a TM to accept the language $L = \{1^n 2^n 3^n   n \geq 1\}$ . (16)	BTL6	Create
4.	Demonstrate a Turing Machine to compute, $f(m+n)=m+n$   $m, n \geq 0$ and simulate their action on the input 0100. (16)	BTL3	Apply
5.	(i) Examine the role of checking off symbols in a Turing Machine. (8) (ii) Describe a Turing Machine M to implement the function "multiplication" using the subroutine copy. (8)	BTL4	Analyze
6.	(i) Solve the turing machine to accept the language $L = \{0^n 1^n   n \geq 1\}$ . (8) (ii) Show that if a language is accepted by a multi tape turing machine, it is accepted by a single-tape TM. (8)	BTL3	Apply
7.	(i) Summarize in detail about multi head and multi tape TM with an example. (8) (ii) Construct a Turing Machine to accept palindromes of even length in an alphabet set $\Sigma = \{a, b\}$ . Trace the strings "abab" and "baab" (8)	BTL5	Evaluate
8.	(i) Explain the TM as computer of integer function with an example. (8) (ii) Design a TM to implement the function $f(x)=x+1$ . (8)	BTL5	Evaluate
9.	(i) Design a TM to accept these to all strings $\{0,1\}$ with 010 as substring. (8) (ii) Write short notes on Two-way infinite tape TM. (8)	BTL6	Create

10.	(i) Draw a Turing machine to find 1's complement of a binary number. (8) (ii) Draw a Turing machine to find 2's complement of a binary number. (8)	BTL5	Evaluate
11.	(i) Model Turing machine for computing $f(m,n)=m*n$ , $n \in \mathbb{N}$ . (8) (ii) Write notes on partial solvability. (8)	BTL5	Evaluate
12.	(i) Construct a TM to reverse the given string {abb}. (8) (ii) Explain Multitape and Multihead Turing machine with suitable example. (8)	BTL6	Create
13.	Design the various programming techniques of Turing machine construction in detail. (16)	BTL6	Create
14.	Explain a TM with no more than three states that accepts the language $a(a+b)^*$ . Assume $\Sigma = \{a,b\}$ . (16)	BTL4	Analyze
15.	Model a Turing Machine to accept palindromes of odd length in an alphabet set $\Sigma = \{a,b\}$ . Trace the strings "ababa" (16)	BTL5	Evaluate
16.	Demonstrate a TM for the language which recognizes the language $L = 01^*0$ . (16)	BTL3	Apply
17.	Compare and explain the deterministic and non-deterministic TM with an example. (16)	BTL5	Evaluate

#### UNIT V

#### COMPUTATIONAL COMPLEXITY

Undecidability- Basic definitions- Decidable and undecidable problems - Properties of Recursive and Recursively enumerable languages -- Post's Correspondence Problem-- complexity classes – introduction to NP-Hardness and NP-Completeness.

#### PART-A

Q.No	Questions	BT Level	Competence
1.	Distinguish between PCP and MPCP? What are the concepts used in UTMs?	BTL2	Understand
2.	List out the features of universal Turing machine.	BTL1	Remember
3.	When a recursively enumerable language is said to be recursive? Discuss it.	BTL2	Understand
4.	Distinguish recursive and recursively enumerable languages	BTL2	Understand
5.	State when a problem is said to be decidable?	BTL1	Remember
6.	Define NP hard and NP completeness problem.	BTL1	Remember
7.	Define a universal language $L_u$ ?	BTL1	Remember
8.	Is it true that the language accepted by a non-deterministic Turing Machine is different from recursively enumerable language? Give your answer.	BTL2	Understand
9.	Identify the two properties of recursively Enumerable sets which are undecidable	BTL1	Remember

10.	When a problem is said to be undecidable? Give an example of undecidable problem.	BTL2	Understand
11.	What is a recursively enumerable language and recursive sets?	BTL1	Remember
12.	Define the classes of P and NP.	BTL1	Remember
13.	Is it true that complement of a recursive language is recursive? Discuss your answer.	BTL2	Understand
14.	Describe about reduction in TM.	BTL1	Remember
15.	List the properties of recursive and recursive enumerable language.	BTL1	Remember
16.	Explain on halting problem.	BTL2	Understand
17.	Give the Properties of Recursive Languages.	BTL2	Understand
18.	Explain about tractable problem.	BTL2	Understand
19.	Describe post correspondence problem.	BTL2	Understand
20.	Outline about time and space complexity of TM.	BTL1	Remember
21.	Describe the encoding of UTM.	BTL1	Remember
22.	Describe about the undecidability of PCP.	BTL1	Remember
23.	Does PCP with two lists $x = (b, a, ca, abc)$ and $y = (ca, ab, a, c)$ have a solution? Explain.	BTL2	Understand
24.	What is meant by P-Class problem?	BTL1	Remember

**PART-B**

1.	(i) Describe about the tractable and intractable problems. (8) (ii) Identify that "MPCP reduce to PCP". (8)	BTL5	Evaluate
2.	(i) Describe about Recursive and Recursive Enumerable languages with example. (8) (ii) State and describe RICE theorem. (8)	BTL5	Evaluate
3.	(i) Summarize diagonalization language. (8) Discuss the significance of universal turing machine and also construct a turing machine to add two numbers and encode it. (8)	BTL5	Evaluate

4.	Outline post correspondence problem. Let $\Sigma = \{0,1\}$ . Let A and B be the lists of three strings each, defined as	BTL4	Analyze													
	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr> <td>i</td> <td><math>w_i</math></td> <td><math>x_i</math></td> </tr> <tr> <td>1</td> <td>1</td> <td>111</td> </tr> <tr> <td>2</td> <td>10111</td> <td>10</td> </tr> <tr> <td>3</td> <td>10</td> <td>0</td> </tr> </tbody> </table>				A	B	i	$w_i$	$x_i$	1	1	111	2	10111	10	3
	A	B														
i	$w_i$	$x_i$														
1	1	111														
2	10111	10														
3	10	0														
	(i) Does the PCP have a solution? (8)															

	(ii) Prove that the universal language is recursively enumerable. (8)		
5.	(i) Illustrate computable functions with suitable example. (8) (ii) Illustrate in detail notes on Unsolvability Problems. (8)	BTL4	Analyze
6.	(i) Describe in detail notes on universal Turing machines with example. (8) (ii) Collect and write the short notes on NP-complete problems. (8)	BTL6	Create
7.	(i) Show that the diagonalization language ( $L_d$ ) is not a recursively enumerable. (8) (ii) Illustrate about unsolvability. (8)	BTL3	Apply
8.	Prove that Post Correspondence Problem is undecidable. (16)	BTL5	Evaluate
9.	(i) Explain about Universal Turing machine and show that the universal language ( $L_U$ ) is recursively enumerable but not recursive. Generalize your answer. (8) (ii) Design and explain how to measure and classify complexity. (8)	BTL6	Create
10.	Prove and explain that the halting problem is undecidable. (16)	BTL5	Evaluate
11.	(i) Show that the characteristic functions of the set of all even numbers is recursive. (8) Illustrate in detail notes on primitive recursive functions with examples. (8)	BTL3	Apply
12.	(i) Point out the Measuring and Classifying Complexity. (8) (ii) Does PCP with two lists $x=(b, b, ab^3, ba)$ and $y=(b^3, ba, a)$ have a solution. Analyze your answer. (8)	BTL4	Analyze
13.	(i) Explain in detail about time and space computing of a Turing machine. (8) (ii) Explain two languages which are not recursively enumerable. (8)	BTL5	Evaluate
14.	(i) Explain in detail Polynomial Time reduction and NP-completeness. (8) (ii) List out the short notes on NP-hard problems. (8)	BTL5	Evaluate
15.	Illustrate in detail about decidable problems. (16)	BTL3	Apply
16.	Illustrate the various complexity classes with an example. (16)	BTL3	Apply
17.	(i) Plan and explain on decidable and undecidable problems with an example. (8) (ii) Design and prove that for two recursive languages $L_1$ and $L_2$ their union and intersection is recursive. (8)	BTL6	Create

