INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

TUTORIAL QUESTION BANK

Course Title	**THEORY OF COMPUTATION**				
Course Code	AITB03				
Programme	B. Tech				
Semester	IV	IT			
Year	2019 – 2020				
Course type	Core				
Regulation	**IARE – R18**				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	**1**	**4**	**-**	**-**
Chief coordinate	Dr.K Srinivasa Reddy, Associate Professor				
Course Faculty	Dr.K Srinivasa Reddy, Associate Professor				

COURSE OVERVIEW:

In theoretical computer science and mathematics, the theory of computation is the branch that deals with how efficiently problems can be solved on a model of computation, using an algorithm. The field is divided into three major branches automata theory and languages, computability theory, and computational complexity theory, which are linked by the question: "What are the fundamental capabilities and limitations of computers?"

COURSE OBJECTIVES:

The course should enable the students to:

I.	Comprehend abstract, mathematical models of computation and use them to solve computational problems
II.	Interpret the relationship between formal languages in Chomsky's hierarchy and different machines.
III.	Analyze and explain the behavior of push-down automata.
IV.	Understand the limits and capacities of Turing's machines to recognize languages.

COURSE OUTCOMES (COs):

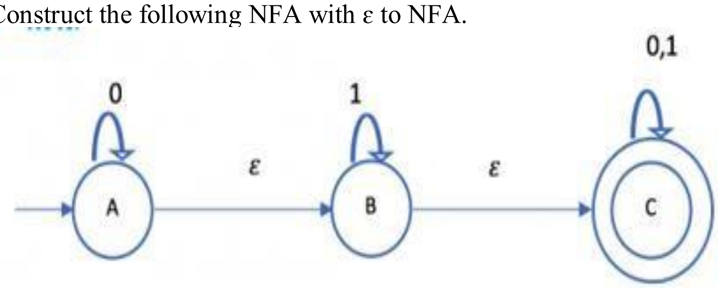
COs	Course Outcome
CO 1	Understand the functionality of deterministic finite automata and Non-deterministic finite automata
CO 2	Apply the regular languages , regular expressions to construct finite automata
CO 3	Apply the context free grammars to construct derivation trees and the accept various strings
CO 4	Compare the functionality of push down automata with deterministic finite automata
CO 5	Apply the concept of Turing machines to solve the complex functions

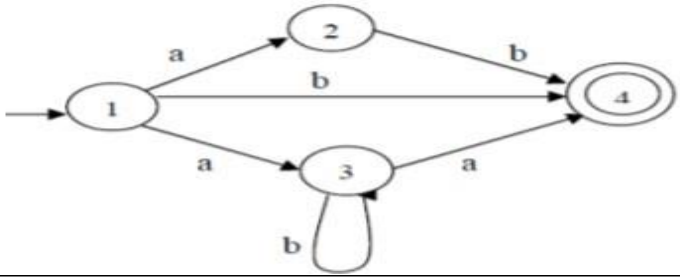
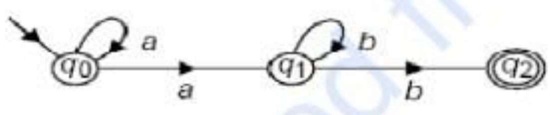
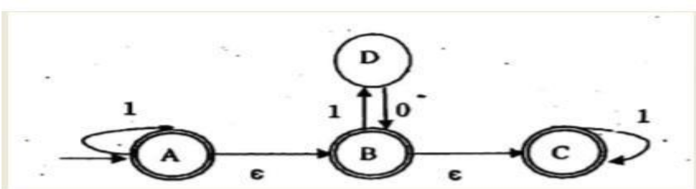
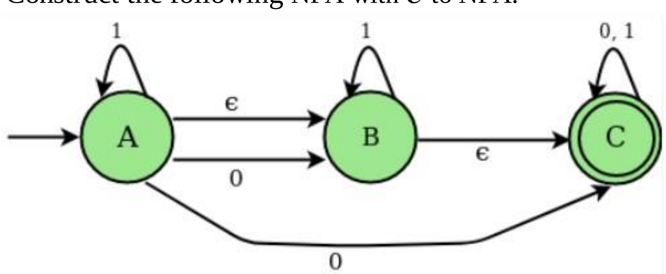
COURSE LEARNING OUTCOMES:

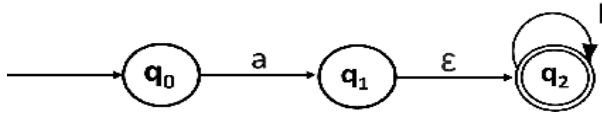
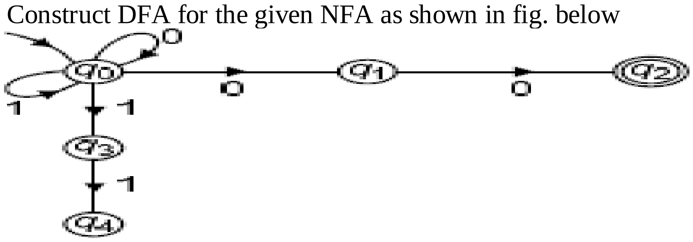
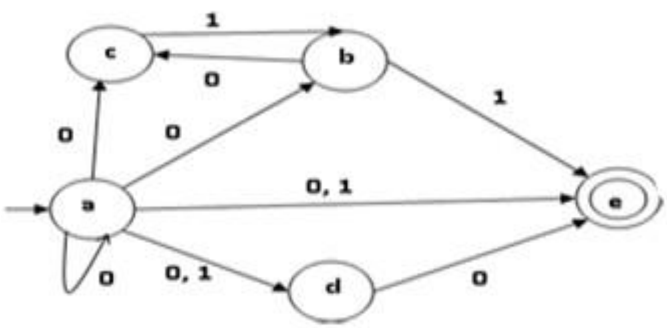
Students, who complete the course, will have demonstrated the asking to do the following:

AITB03.01	Use the definitions and notations for sets, relations and functions in defining and study Finite Automata
AITB03.02	Remember on formal languages and Kleene"s Theorem to intend programming languages
AITB03.03	Construct deterministic and nondeterministic finite state automata (DFA and NFA) for solving simple decision problems.
AITB03.04	Perform conversions between nondeterministic finite automata, deterministic finite automata and regular expressions and finite state automata to gain Remember about formal proofs in computer science
AITB03.05	Remember on recursive definitions of regular languages, regular expressions and the use of regular expressions to represent regular languages
AITB03.06	Detailed Remember on the relationship between regular expressions and finite automata
AITB03.07	Identify that few languages are not regular by using Pumping lemma
AITB03.08	Remember on Left Linear grammar, Right Linear grammars and converting grammars into Finite Automata.
AITB03.09	Understand the fundamental role played by Context-Free Grammars (CFG) in designing formal computer languages with simple examples
AITB03.10	Remember on Context‐ Free Grammars so that able to prove properties of Context‐ Free Grammars.
AITB03.11	Identify relationship between regular languages and context-free grammars
AITB03.12	Use the pumping lemma for Context Free Languages to show that a language is not context‐free
AITB03.13	Understand the equivalence between Context-Free Grammars and Non-deterministic Pushdown Automata
AITB03.14	Understand deterministic Pushdown Automata to parse formal language strings by using (i) top down or (ii) bottom up techniques
AITB03.15	Remember on converting Context-Free Grammars into pushdown automata to identify the acceptance of a string by the Context Free Language
AITB03.16	Understand the path processing computation using Turing Machines (Deterministic and Non-Deterministic) and Church-Turing Thesis in computers.
AITB03.17	Remember on non-halting Turing Machine accepted by Recursively Enumerable Languages
AITB03.18	Understand the power of the Turing Machine, as an abstract automaton, that describes computation, effectively and efficiently
AITB03.19	Theory of Computation is important in programming language design, parsers, web-Scrappers, Natural Language Processing (NLP), and is at the heart of modern compiler architectures.
AITB03.20	Process the remember and skills for employability and to succeed in national and international level competitive exams.

TUTORIAL QUESTION BANK

MODULE-1				
FINITE AUTOMATA				
PART – A (Short Answer Questions)				
S. No	Questions	**Blooms Taxonomy Level**	**Course Outcomes**	**Course Learning Outcomes**
1	Demonstrate finite state Automata.	Understand	CO1	AITB03.01
2	Distinguish between DFA and NFA.	analyze	CO1	AITB03.03
3	Describe the concept of String.	Understand	CO1	AITB03.01
4	Describe transition function of DFA.	Understand	CO1	AITB03.01
5	Demonstrate about ε–transitions.	Understand	CO1	AITB03.03
6	Highlight the power of an alphabet (∑*).	Understand	CO1	AITB03.01
7	List out the applications of finite automata.	Remember	CO1	AITB03.01
8	Describe the concept of Null string.	Understand	CO1	AITB03.01
9	Demonstrate the Kleene Star?	Remember	CO1	AITB03.01
10	Illustrate NFA with example.	Understand	CO1	AITB03.03
11	Describe transition diagram for DFA accepting string ending with 00	Understand	CO1	AITB03.01
12	Construct DFA for a string accepting odd number of 0‟s.	apply	CO1	AITB03.03
13	Illustrate transition diagram for DFA to accept exactly one „a‟ defined. over an alphabet ∑= {a,b}..	Understand	CO1	AITB03.03
14	Construct DFA for odd number of 1‟s.	Apply	CO1	AITB03.03
15	Demonstrate ε - closure.	Understand	CO1	AITB03.03
16	Describe FSM and its structure with an example.	Understand	CO1	AITB03.01
17	State the Mathematical definition of Finite Automata.	Remember	CO1	AITB03.01
18	Construct DFA for even number of 1‟s.	Apply	CO1	AITB03.01
19	Demonstrate DFA mathematically.	Remember	CO1	AITB03.01
20	Construct DFA for the language accepting strings which contains 001 as substring.	Apply	CO1	AITB03.03
Part - B (Long Answer Questions)				
1	Construct a DFA to accept set of all strings ending with 0101.	Apply	CO1	AITB03.01
2	Evaluate the DFA with the set of strings having „aaa‟ as a substring over an alphabet ∑ ={a,b}.	Evaluate	CO1	AITB03.01
3	List out the various differences between DFA and NFA	Remember	CO1	AITB03.01
4	Describe NFA with Ɛ to NFA conversion with an example.	Understand	CO1	AITB03.01
5	Construct a DFA to accept the string a‟s and b‟s ending with abb over an alphabet ∑ ={a,b}	Apply	CO1	AITB03.03
6	Describe the properties and operations of strings and languages.	Understand	CO1	AITB03.01
7	Construct a DFA that any given decimal number is divisible by 3.	Apply	CO1	AITB03.04
8	Design DFA for the following languages shown below Σ = { a,b} a) L={w/ w is any string that doesn‟t contain exactly two a‟s} b) L={w/ w is any string that contain atmost 3a‟s}	Apply	CO1	AITB03.03
9	Construct the following NFA with ε to NFA.  	Apply	CO1	AITB03.03
10	Construct Finite Automata and draw FA for the strings over an alphabet Σ = { 0,1} (i) The string with even no of 0‟s and odd no of 1‟s	Apply	CO1	AITB03.03

	(ii) The string with odd no of 0‟s and odd no of 1‟s			
11	Construct a DFA, the language recognized by the Automaton being L={w/ w contains neither the substring ab nor ba}.Draw the transition table.	Apply	CO1	AITB03.04
12	Convert the following NFA into DFA. 	Understand	CO1	AITB03.04
13	Construct the DFA for the following language L={w/|w|mod3=0,w belongs to $(a,b)^*$} L={w/|w|mod3=1,w belongs to $(a,b)^*$}	Apply	CO1	AITB03.04
14	Design a DFA for the following language over an alphabet ∑ = { 0,1} i) The string with even no of 0‟s and even no of 1‟s ii) The string with odd no of 0‟s and even no of 1‟s	Create	CO1	AITB03.03
15	Convert the following NFA into equivalent DFA. 	Understand	CO1	AITB03.03
16	Convert the following NFA-ε to NFA. 	Analyze	CO1	AITB03.03
17	Construct the DFA for the following language L={w/n_a|w|mod3=0,w belongs to $(a,b)^*$} L={w/n_a|w|mod3=1,w belongs to $(a,b)^*$}	Apply	CO1	AITB03.03
18	Construct the following NFA with ε to NFA. 	Apply	CO1	AITB03.03
19	Construct a DFA that accepts set of strings starts with 01and ends with 01 over alphabet ∑ = {0, 1}	Apply	CO1	AITB03.03
20	Illustrate the model and behavior of finite automata with neat block diagram.	Understand	CO1	AITB03.04
Part – C (Problem Solving and Critical Thinking Questions)				
1	Design NFA for accepting any binary string that contains 11 as a substring and Convert to DFA.	Create	CO1	AITB03.04
2	Construct NFA with ε to equivalent DFA	Understand	CO1	AITB03.04

				
3	Construct a DFA that any given decimal number is divisible by 5.	Apply	CO1	AITB03.04
4	Construct the DFA for the following language L={w/|w|mod5=0,w belongs to $(a,b)^*$} L={w/|w|mod5=1,w belongs to $(a,b)^*$}			
5	Design the NFA from a given NFA with ε machine $M=(\{q_0,q_1,q_2\},\{0,1,2\}, \delta, q_0, \{q_2\})$ where δ is given by $[\delta (q_0,0)=\{q_0\}, \delta (q_0,1)=\phi, \delta (q_0,2)=\phi, \delta(q_0, \varepsilon)=\{q_1\}]$ $[\delta (q_1,0)=\phi, \delta (q_1,1)=\{q_1\} , \delta (q_1,2)=\phi ,\delta (q_1, \varepsilon)=\{q_2\}]$ $[\delta (q_2,0)=\phi, \delta (q_2,1)= \phi , \delta (q_2,2)= \{q_2\},\delta (q_2, \varepsilon)= \phi]$	Create	CO1	AITB03.03
6	Construct a NFA that strings such that the third symbol from the right end is a 0 over an alphabet ∑ ={0,1}. And Convert it into equivalent DFA.	Apply	CO1	AITB03.03
7	Construct DFA for the given NFA as shown in fig. below 	Apply	CO1	AITB03.04
8	Develop the transition diagram for the below NFA and then convert its equivalent transitition diagram for DFA. 	Apply	CO1	AITB03.03
9	Construct the DFA that will accept those words from ∑= {a, b} where the number of a's is divisible by two and the number of b's is divisible by three. Sketch the transition table of the finite automata.	Apply	CO1	AITB03.03
10	Construct the DFA that will accept those words from alphabets Σ={a, b} where the number of b"s is divisible by three. Sketch the transition table and diagram of the finite Automata.	Apply	CO1	AITB03.04

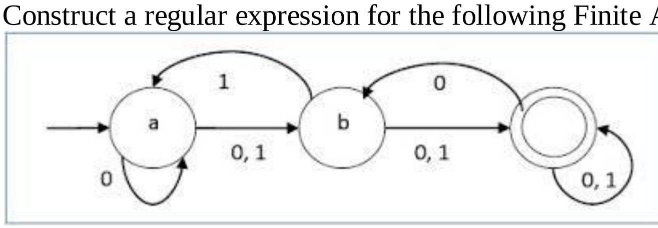
MODULE-II

REGULAR LANGUAGES

PART – A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1	Demonstrate Regular Languages.	Understand	CO2	AITB03.05
2	List out any two applications of regular expression.	Remember	CO2	AITB03.05
3	Construct Pumping Lemma for Regular Languages.	Apply	CO2	AITB03.07
4	Illustrate an example for a regular set?	Remember	CO2	AITB03.05

5	Construct the Regular Expression for the empty string.	Remember	CO2	A ITB03.08
6	Describe regular expression for denoting language containing empty	Understand	CO2	AITB03.05
7	Demonstrate right linear grammars.	Understand	CO2	AITB03.08
8	Construct the Regular Expression for the set of binary strings.	Apply	CO2	AITB03.05
9	Demonstrate Regular grammars.	Understand	CO2	AITB03.05
10	List out the advantages of regular expressions.	Remember	CO2	AITB03.05
11	Demonstrate Regular set?	Understand	CO2	AITB03.05
12	State regular expressions for the Set of strings over {0, 1} whose last two symbols are the same.	Remember	CO2	AITB03.05
13	Describe the regular language generated by regular expression (0+1)*001(0+1)*.	Understand	CO2	AITB03.05
14	Summarize the difference between left linear and right linear	Understand	CO2	AITB03.08
15	Describe the Regular Expression to generate at least one b over Σ={a,b}	Understand	CO2	AITB03.05
	Part - B (Long Answer Questions)			
1	Convert Regular Expression 01* + 1 to Finite Automata.	Understand	CO2	AITB03.05
2	Construct Right linear, Left linear Regular Grammars for 01*+1.	Apply	CO2	AITB03.05
3	Demonstrate Regular expression? Simplify the following Regular Expression i)ε + 1*(011)*(1*(011)*)*=(1+011)* ii)(0+11*0)+ (0+11*0)(10+10*1) * (10+10*1) *=1*0(10+10*1) *	Understand	CO2	
4	Construct Regular grammar for the given Finite Automata.(a+b)*ab*.	Apply	CO2	AITB03.08
5	Construct Regular grammar for the given Finite Automata 0*11(0+1)*	Apply	CO2	AITB03.08
6	Demonstrate Regular expression, Regular set and Finite Automata distinguish those with example representations.	Understand	CO2	AITB03.08
7	Construct the Finite Automata(NFA-ε) for given regular expression (0+1)*00(0+1)*	Apply	CO2	AITB03.07
8	Convert Regular Expression (b+aa)*a* to Finite Automata(NFA-ε).	Understand	CO2	AITB03.05
9	State Pumping Lemma for Regular Languages with a suitable example.	Remember	CO2	AITB03.07
10	Convert given Regular expression (a*b*) * to FA(NFA-ε).	Understand	CO2	AITB03.08
11	Convert the following automata into Regular expression M=({q_1,q_2,q_3},{0,1}, δ, q_1, {q_1}) where δ is given by [δ (q_1,0)={q_1}, δ (q_1,1)= {q_2}] [δ (q_2,0)= {q_3}, δ (q_2,1)={q_2}] [δ (q_3,0)= {q_1}, δ (q_3,1)= {q_2}]	Remember	CO2	AITB03.08
12	Describe Pumping lemma. Prove that the language L={yy/y belongs {0,1}*} is not regular.	Understand	CO2	AITB03.08
13	Demonstrate Regular grammar? Explain the types of regular grammar with examples.	Remember	CO2	AITB03.07
14	IIustrate the steps for conversion of regular grammar to finite automata? Construct the FA for the following grammar S→aS/bA/b A→aA/bS/a	Understand	CO2	AITB03.05
15	Convert the given Regular Expression 1(11) * to FA and convert it in to NFA.	Remember	CO2	AITB03.07
16	Prove that the following languages is not regular i) L = {$a^n b^n$/n >=1} ii) L={a^P/p is prime}	Evaluate	CO2	AITB03.08
17	Convert the following regular expression to Regular grammar (0+1)*00(0+1)*	Remember	CO2	AITB03.08
18	Construct the Left Linear Grammar for the strings start with a over an alphabet ∑ ={a,b}.	Apply	CO2	AITB03.08
19	Illustrate the steps for conversion from Finite Automata to Regular Expression with example?	Understand	CO2	AITB03.07
20	Describe Pumping lemma. Prove that the language L={yy/y belongs {0,1}*} is not regular.	Understand	CO2	AITB03.05
	Part – C (Problem Solving and Critical Thinking Questions)			

1	Convert Regular Expression (11+0)*(00+1)* to Finite Automata.	Remember	CO2	AITB03.08
2	Construct Regular Grammar for the following Expressions i) a(a + b)* ii) (aa + bb)	Understand	CO2	AITB03.08
3	DemonstratePumping Lemma for Regular Languages. Prove that the language $L = \{a^n / n \text{ is a } n^5\}$ is not regular	Understand	CO2	AITB03.08
4	Construct the DFA Transition diagram for equivalent Regular expression (ab+a) *(aa+b)	Apply	CO2	AITB03.05
5	Prove that following languages are not regular $L=\{a^nb^m \mid n, m$ and$n<m\}$ $L=\{a^n \mid n$ is a perfect square $\}$	Evaluate	CO2	AITB03.06
6	Construct the equivalent DFA for following Regular Expression (0 + 1)*(00 + 11)(0 + 1)* and Find reduced DFA.	Apply	CO2	AITB03.08
7	Construct the NFA for following Regular expression (0+1)*(01+110).	Apply	CO2	AITB03.08
8	Convert the following automata into Regular expression $M=(\{q_1,q_2,q_3\},\{0,1\}, \delta, q_1, \{q_2,q_3\})$ where δ is given by $[\delta (q_1,0)=\{q_2\}, \delta (q_1,1)= \{q_3\}]$ $[\delta (q_2,0)= \{q_1\}, \delta (q_2,1)=\{q_3\}]$ $[\delta (q_3,0)= \{q_2\}, \delta (q_3,1)= \{q_2\}]$	Remember	CO2	AITB03.05
9	Construct the following language is not regular i)L= $\{ a^n ba^n / n=0,1,2\ldots\ldots. \}$ ii) L= $\{a^nb^{2n} / n\geq 0 \}$	Apply	CO2	AITB03.06
10	Construct the Left Linear Grammar and Right Linear grammar for the automata in which strings end with 101 over an alphabet $\sum =\{0,1\}$.	Apply	CO2	AITB03.08
11	Construct a regular expression for the following Finite Automata 	Apply	CO2	AITB03.06

MODULE -III

CONTEXT FREE GRAMMARS

Part - A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	COs	Course Learning Outcomes
1	Develop a context free grammar(CFG).	Apply	CO3	AITB03.09
2	Construct the parse tree with example.	Apply	CO3	AITB03.09
3	Compare and contrast the Rightmost derivation with Left most derivation with example.	Understand	CO3	AITB03.10
4	Demonstrate a short notes about leftmost derivation with example.	Understand	CO3	AITB03.10
5	Express any two applications of Context Free Grammar.	Understand	CO3	AITB03.11
6	Narrate the left sentential form?	Understand	CO3	AITB03.12
7	Express the different ways to derive a string from a CFG.	Understand	CO3	AITB03.09
8	Demonstrate the language generated by CFG or G?	Understand	CO3	AITB03.12
9	Demonstrate the concept of parse tree?	Understand	CO3	AITB03.11
10	Demonstrate the concept of subtree.	Understand	CO3	AITB03.10
11	If S->aSb | aAb , A->bAa , A->ba. Find out the CFL	Understand	CO3	AITB03.10
12	Demonstrate the usage of normalization?	Understand	CO3	AITB03.09
13	Analyze the ambiguous grammar with example?	Analyze	CO3	AITB03.09
14	Evaluate the following grammar into regular grammar that generates the same language S-> AB	Evaluate	CO3	AITB03.09

	A->aAa|bAb|a|b B-> Ab|Bb|ε			
15	Simplify the CFG to reduce UNIT production.	Analyze	CO3	AITB03.11
16	Express the elimination of useless symbols in productions.	Understand	CO3	AITB03.12
17	Solve the given grammar to get the minimized CFG – S → a S/A, A ->a / B	Apply	CO3	AITB03.11
18	Express the ambiguity concept in CFG with an example	Understand	CO3	AITB03.10
19	Describe the is the use of CNF and GNF.	Understand	CO3	AITB03.11
20	Evaluate the minimization of CFG - S → aS1b S1→aS1b/ε.	Evaluate	CO3	AITB03.11
21	Express the minimized CFG - S → A,A →aA/ ε.	Understand	CO3	AITB03.12
22	Construct the minimized CFG - S →AB / a,A → a. B→b	Apply	CO3	AITB03.12
23	Covert to to the minimized CFG - S→aS/A/C A →a B→aa C→aCb.	Apply	CO3	AITB03.12
24	Convert the given grammar to CNF - S→aAbB A→aA/a B→bB/a.	Understand	CO3	AITB03.10
	Part – B (Long Answer Questions)			
1	Construct Leftmost Derivation. , Rightmost Derivation, Derivation Tree for the following grammar with respect to the string aaabbabbba. S→aB | bA A→aS| bAA|a B→ bS | aBB | b	Apply	CO3	AITB03.12
2	Design a CFG for the languages $L=\{a^i b^j \mid i<=2j\}$	Create	CO3	AITB03.12
3	Construct leftmost and rightmost derivations for the strings, if the language is given as S→AS| ε A→aa|ab|ba|bb Strings: a) aabbba b) baabab c) aaabbb	Apply	CO3	AITB03.12
4	Create the minimization of CFG - S→AbA A→Aa/ ε.	Create	CO3	AITB03.12
5	Calculate the minimization of CFG - S→aSa S →bSb S→a/b/ ε.	Understand	CO3	AITB03.10
6	Construct the minimization of CFG - S→A0/B A→0/12/B	Apply	CO3	AITB03.11
7	Convert the grammar to CNF - S→aSa/aa S→bSb/bb S →a/b.	Understand	CO3	AITB03.12

8	Write short notes on Chomsky Normal Form and Greibach Normal Form.	Understand	CO3	AITB03.10
9	What is Normalization of CFG? What is the use of Normalization? Explain different types of normal forms.	Remember	CO3	AITB03.10
8	Illustrate the construction of Greibach normal form with an example.	Understand	CO3	AITB03.10
9	Prove that the following CFG ambiguous. S→iCtS | iCtSeS | a, C→ b.	Evaluate	CO3	AITB03.12
10	Demonstrate the Pumping lemma for Context Free Languages concept with example $\{a^n b^n c^n$ where n>=0}.	Understand	CO3	AITB03.12
11	Illustrate the simplified CFG productions in S → a S1b S1 → a S1b/ Є	Understand	CO3	AITB03.11
12	Convert the following CFG into GNF. S→AA/a,A → SS/b	Remember	CO3	AITB03.11
13	Describe unit production? Explain the procedure to eliminate unit production.	Understand	CO3	AITB03.12
14	Illustrate the procedure to eliminate ϵ-productions in grammar.	Understand	CO3	AITB03.09
15	Convert the following grammar into GNF G=({A1,A2,A3},{a,b},P,A) A1→A2A3 A2→A3A1/b A3→A1A2/a	Understand	CO3	AITB03.09
16	Express simplified CFG productions from the following grammar A→aBb/bBa B→aB/bB/ϵ	Understand	CO3	AITB03.10
17.	Convert the following grammar into GNF S→ABA/AB/BA/AA/B A→aA/a, B→bB/b	Understand	CO3	AITB03.11
18	Express the minimized CFG for the following grammar S→ABCa | bD A→BC |b B→b | ε C→Đ| ε, D→d	Understand	CO3	AITB03.12
19	Covert the CFG to Greiback Normal form by taking an example	Understand	CO3	AITB03.12
20	Design the grammar G given by S->aAa A->Sb| bcc|DaA C->abb| DD E-> ac D->aDa into an equivalent grammar by removing useless symbols and useless productions from it	Create	CO3	AITB03.12
	Part – C (Problem Solving and Critical Thinking Questions)			
1	Design a grammar for valid expressions over operator - and /. The arguments of expressions are valid identifiers over symbols a,b, 0 and Derive Left Most Derivation and Right Most Derivation for string W= (a11-b0) / (b00-a01). Draw parse tree for Left Most Derivation.	Understand	CO3	AITB03.11
2	Convert the following grammar into GNF A1→A2 A3A2→A3 A1 /b A3→A1 A2 /a	Apply	CO3	AITB03.12
3	Evaluate the following grammar : S→ABC | BbB A→ aA |BaC|aaa B→ bBb| a|D C→CA|AC D→ε Eliminate ε-productions. Eliminate any unit productions in the resulting grammar. Eliminate any useless symbols in the resulting grammar. Convert the resulting grammar into Chomsky Normal Form	Evaluate	CO3	AITB03.11
4	Design a grammar for valid expressions over operator - and /. The Arguments of expressions are valid identifiers over symbols a,b, 0 and 1. Derive Left Most Derivation and Right Most Derivation for string W= (a11-b0) / (b00-a01). Draw parse tree for Left Most Derivation.	Create	CO3	AITB03.11
5	Narrate the CFG G, find a CFG G" in Chomsky Normal form generating L(G) – { Λ} S →AaA | CA | BaB A →aaBa | CDA | aa | DC B →bB | bAB | bb | aS C →*Ca* | *bC* | *D*	Understand	CO3	AITB03.11

	D → $bD \mid \Lambda$			
6	Convert the following grammar into GNF A1→A2 A3A2→A3 A1 /b A3→A1 A2 /a	Understand	CO3	AITB03.11
7	Describe CFG and Design a CFG for the following language. L = { $0^i 1^j 0^k \mid j > i + k$ }	Understand	CO3	AITB03.11
8	Explore the following grammar : S->ABC | BbB A-> aA |BaC|aaa B-> bBb| a|D C->CA|AC D->ε Eliminate ε-productions. Eliminate any unit productions in the resulting grammar. Eliminate any useless symbols in the resulting grammar. Convert the resulting grammar into Chomsky Normal Form	Understand	CO3	AITB03.11
9	Express formally that the language { $a^n b^m \mid n \leq m \leq 2n$ } is not deterministically context-free. Unfortunately, when applying the operation pre from the chapter, we obtain { $a^n b^m \mid n < m \leq 2n$ } which still is context-free. We need a different closure property of the deterministic context-free languages to tackle this problem	Understand	CO3	AITB03.11
10	Demonstrate formally that the language $\{ w\,\mathrm{mir}(w) \mid w \in \{a, b\}^* \}$ is not deterministically context	Understand	CO3	AITB03.12
11	Show that { $a^m b^n c^p \mid m < n$ or $n < p$ } is not deterministically context	Understand	CO3	AITB03.13
12	Simplify the context free grammar for the given CFG S-> Ab |Bb A->a |aS |Baa B-> b|bS |aBB	Analyze	CO3	AITB03.11
14	Construct the CFG for the language L={$a^n b^{2n}$ where n>=1}	Understand	CO3	AITB03.12
15	Construct a CFG to generate unequal number of a"s and b"s	Understand	CO3	AITB03.13
16	Design the context free grammars in the four tuble form.(V,T,P,S) for the given languages on Σ={a,b} i) All strings having at least two a"s ii)All possible strings not containing triple b"s	Create	CO3	AITB03.11
17	Obtain the the string "aabbabba"for left most derivation and rightmost derivation using a CFG given by S->Ab|Ba A->a|aS|Baa B->b | bS |aBB	Understand	CO3	AITB03.11
18	Describe and distinguish regular grammar and context free grammar	Understand	CO3	AITB03.12
19	Develop a left linear grammar for the Language given below L={ $a^n b^m$|n>=2,m>=3}	Apply	CO3	AITB03.13
20	Develop the context free grammar with the minimal number of variables that generates the language for the below languages L={w|w=W^R}(w^R denotes the reverse of w) L={w|w=!W^R}	Apply	CO3	AITB03.11
	MODULE-IV			
	PUSH DOWN AUTMATA			
	Part - A (Short Answer Questions)			
1.	Distinguish between deterministic and nondeterministic PDA.	Analyze	CO4	AITB03.13
2.	Demonstrate the concept of PDA.	Understand	CO4	AITB03.14
3.	Demonstrate the concept of NPDA.	Understand	CO4	AITB03.14
4.	Describe the language of DPDA.	Understand	CO4	AITB03.14
5.	Convert the following PDA to CFG δ(q0,0,z0)={q0,xz0)	Understand	CO4	AITB03.15

6.	Convert the following PDA to CFG δ(q0,0,x)=(q0,xx)	Understand	CO4	AITB03.15
7.	Convert the following PDA to CFG δ(q0,1,x)=(q1,ϵ)	Understand	CO4	AITB03.15
8.	Convert the following PDA to CFG δ(q1,1,x) = (q1,ϵ)	Understand	CO4	AITB03.15
9.	List out the steps to convert CFG to PDA.	Remember	CO4	AITB03.15
10	Express the acceptance of PDF by final state.	Understand	CO4	AITB03.14
11	Express the acceptance of PDF by empty stack.	Understand	CO4	AITB03.14
12	Convert the following PDA to CFG δ(q0,b,z0)={q0,zz0)	Understand	CO4	AITB03.14
13.	construct the following PDA to CFG δ(q0,b,z) = (q0,zz)	Understand	CO4	AITB03.14
14.	Convert the following PDA to CFG δ(q0, ϵ ,z0)=(q0,ϵ)	Understand	CO4	AITB03.15
15	Describe the PDA and design PDA for L = { x ∈ { a, b}* | na(x) > nb(x) }	Understand	CO4	AITB03.15
	Part – B (Long Answer Questions)			
1.	State the NPDA(Nondeterministic PDA) and DPDA(deterministic PDA) equivalent? Illustrate with an example.	Remember	CO4	AITB03.16
2.	Construct the grammar for the following PDA. M=({q0, q1},{0,1},{X,z0},δ,q0,Z0,Φ) and where δ is 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)}.	Remember	CO4	AITB03.15
3.	Construct PDA for string of form $a^n b^{2n}$	Apply	CO4	AITB03.14
4.	Express PDA mathematically. With a neat diagram explain the working of a Turing Machine	Understand	CO4	AITB03.15
5.	Evaluate the PDA that accepts the language {a^m b^n/n>m}	Evaluate	CO4	AITB03.15
6.	Design a PDA for the following grammar S→0A, A→0AB/1,B→1	Create	CO4	AITB03.16
7.	Convert the following PDA to CFG M=({q0,q1},{a,b},{z0,za}, δ,q0,z0,Φ) δ is given by, δ(q0,a,z0)=(q0,zz) δ(q0,a,z)=(q0,zz0) δ(q0,b,z)=(q1,ϵ) δ(q1,b,z)=(q1,ϵ) δ(q1,ϵ,z0)=(q1,ϵ)	Understand	CO4	AITB03.14
8.	Demonstrate the PDA mathematically. Construct the PDA for the following language. L= {w /w of form $a^n b^n$}.	Understand	CO4	AITB03.14
9	Evaluate and express the following For the language L = { xcxr / x Є {a,b}* } design a PDA (Push Down Automata) and trace it for string “bacab “	Understand	CO4	AITB03.16
10	Demonstrate the Pushdown automaton A is specified by A = ({q0, q1}, {a, b}, {Z, X}, δ, qin, Z, ∅), where δ contains the following transitions: (q0, a, Z) ›→ (q0, λ), (q0, a, Z) ›→ (q0, XZin), (q0, a, X) ›→ (q0, XX),(q0, b, X) ›→ (q1, λ), (q1, b, X) ›→ (q1, λ),(q1, a, Z) ›→ (q0, Z). Infer a (reduced) context-free grammar G for the empty stack language of A, i.e., L(G) = Le(A).	Understand	CO4	AITB03.14
11	Construct PDA for the below grammar as shown below S->aABB |aAA A->aBB |a B->bBB | A that accepts the language generated by given grammar	Apply	CO4	AITB03.14
12	Design a PDA for the below CFG which generates the palindrome accepted by L(G) S->aSa| bSb|a|b	Create	CO4	AITB03.14
13	Design a PDA and describe a context free grammar for the language L={ $a^i b^j c^k$; i<j or j<k}	Create	CO4	AITB03.16
14	Covert the following context free grammar to push down automata S->aA|bB A-> aB|a	Understand	CO4	AITB03.14

	B->b Verify the string aab accepted by equivalent PDA			
15	Design DPDA for L= a^nb^n where n>=1	Create	CO4	AITB03.14
16	Construct PDA accepts PDA M for the language L={ WWR|W ϵ {a,b}* }such that L=L(M)	Apply	CO4	AITB03.14
17	Design PDA M for the language L={x ϵ {a,b}* | n_a (x)>n_b(x)}	Create	CO4	AITB03.16
18	Prove that the below languages are deterministic context free languages? a) L1={0^n1^m|n=m and n>=1} b) L2 = {o^n1^m|n=2m and n>=1}	Evaluate	CO4	AITB03.14
19	Narrate deterministic context free languages and deterministic push down automata	Understand	CO4	AITB03.14
20	Construct PDA that recognizes the language L={x=x^R : x ϵ{a,b}$^+$}	Apply	CO4	AITB03.14
Part – C (Problem Solving and Critical Thinking Questions)				
1	Construct PDA for equal number of x's and y's. eg: xyyyxy	Apply	CO4	AITB03.14
2	Construct NDPDA for L = { W#W^R /W ϵ (X + Y)*}	Apply	CO4	AITB03.14
3	Convert the following PDA to CFG 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)	Understand	CO4	AITB03.15
4	Construct DPDA for L = { W#W^R /W ϵ (X + Y)*}	Apply	CO4	AITB03.15
5	Construct pushdown automata for the following languages. Acceptance either by empty stack or by finalstate. (a) $\{ a^n b^m a^n \mid m, n \in \mathbb{N}\}$ (b) $\{ a^n b^m c^m \mid m, n \in \mathbb{N} \}$ (c) aibjck | i, j, k ∈ N, i > j} (d) $\{a^i b^j c^k \mid i, j, k \in \mathbb{N}, i + j = k\}$ (e) $\{a^i b^j c^k \mid i, j, k \in \mathbb{N}, i + k = j\}$ (f) $\{ a^n b^m \mid n \leq m \leq 2n\}$ (g) PAL $= \{w \in \{a, b\}^* \mid \mathrm{mir}(w) = w\}$ (h) $\{w_1 c w_2 c \cdots c w_k c x \mid x, w_1, \ldots, w_k \in \{a, b\}^*,$ $k \in \mathbb{N}, x = \mathrm{mir}(w_j)$ for some $j\}$ (i) $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$, $\#_a(w)$ represents the number of a"s in w (j) $\{w \in \{a, b\}^* \mid \#_a(w) = 2 \cdot \#_b(w)\}$	Apply	CO4	AITB03.15
6	Construct a PDA with final state acceptance for thelanguage $B = \{ \mathrm{bin}(i)\$\mathrm{mir}(\mathrm{bin}(i + 1)) \mid i \geq 0 \} \subseteq \{0, 1, \$\}^*$ Here is bin(i) ∈ {0, 1}∗ the binary representation (without leading zero"s) of the number i. Eg. bin(11) = 1011 and mir(bin(12)) = 0011	Apply	CO4	AITB03.15
7	Construct CFG corresponding to PDA whose transition mapping is as follows. δ(S,a,X)= (s, A,X) δ(S,b,A) =(s, AA) δ(S,a,A) = (s, AA)	Apply	CO4	AITB03.14

8	Prove that given CFG with following productions S->aBc A-> abc B->aAb C->AB C->c constructs a PDA M such that the language generated by M and G are equivalent.	Evaluate	CO4	AITB03.14
9	Design a PDA for the following grammar. S->0A A->0AB B->1	Create	CO4	AITB03.15
10	Construct PDA for the following grammar S->AA | a A->SA | b	Create	CO4	AITB03.15

MODULE-V

TURING MACHINE

Part - A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	COs	Course Learning Outcomes
1	Describe the Chomsky hierarchy of languages.	Understand	CO5	AITB03.16
2	Demonstrate Context sensitive language.	Understand	CO5	AITB03.16
3	Explore Turing Machine	Understand	CO5	AITB03.16
4	Express Type 0 grammars .	Understand	CO5	AITB03.16
5	Narrate the Type 1 grammars .	Understand	CO5	AITB03.16
6	Demonstrate Type 2 grammars .	Understand	CO5	AITB03.16
7	Describe Type 3 grammars .	Remember	CO5	AITB03.16
8	List out the types of grammars.	Remember	CO5	AITB03.16
9	Describe the moves in Turing Machine.	Understand	CO5	AITB03.16
10	Demonstrate an Instantaneous Description of a Turing Machine.	Remember	CO5	AITB03.17
11	Express the Language of Turing Machine.	Remember	CO5	AITB03.17
12	List out types of TMs.	Remember	CO5	AITB03.18
13	Distinguish the difference between PDA and TM	Understand	CO5	AITB03.17
14	Describe the multi head Turing Machine.	Understand	CO5	AITB03.18
15	obtain multi dimensional Turing Machine.	Remember	CO5	AITB03.18
16	Demonstrate multiple tapes Turing Machine.	Understand	CO5	AITB03.18
17	Describe the recursive languages.	Remember	CO5	AITB03.17
18	Construct recursively enumerable languages.	Apply	CO5	AITB03.17
19	Describe two way infinite Turing Machine.	Remember	CO5	AITB03.18
20	Demonstrate the non deterministic Turing Machine.	Understand	CO5	AITB03.18
21	Construct Turing Machine for 1's complement for binary numbers.	Understand	CO5	AITB03.18
22	Distinguish Recursive languages and Recursively enumerable languages.	Understand	CO5	AITB03.19
23	Demonstrate Church's Hypothesis.	Understand	CO5	AITB03.20

Part - B (Long Answer Questions)

1	Describe short notes on Context sensitive language and linear bounded automata.	Understand	CO5	AITB03.16
2	Classify briefly about Chomsky hierarchy of languages..	Understand	CO5	AITB03.16
3	Describe a Turing Machine. With a neat diagram explain the working of a Turing Machine.	Understand	CO5	AITB03.16
4	Compare Turing Machine with other automata.	Understand	CO5	AITB03.18
5	Construct a Transition diagram for Turing Machine to accept the language L= { $w\#w^R$ | w ϵ (a + b) *}	Understand	CO5	AITB03.17
6	Express short notes on Recursive and Recursively Enumerable languages.	Understand	CO5	AITB03.17

7	Describe the properties of recursive and recursively enumerable languages.	Understand	CO5	AITB03.17
8	Develop a Turing Machine to accept strings formed with 0 and1 and having substring 000.	Understand	CO5	AITB03.16
9	Construct a Transition diagram for Turing Machine to accept the language L= { ww^R | w ϵ (a + b) *}	Apply	CO5	AITB03.16
10	Design a Transition table for TM L={$a^nb^nc^n$/n>=1}	Create	CO5	AITB03.16
11	Construct a Transition table for Turing Machine to accept the following language. L = { $0^n1^n0^n$ | n ≥1}	Apply	CO5	AITB03.16
12	Construct a Turing Machine that accepts the language L = {$1^n2^n3^n$ | n ≥1}. Give the transition diagram for the Turing Machine obtained and also show the moves made by the Turing machine for the string 111222333.	Apply	CO5	AITB03.16

13	Enumerate Linear bounded automata and explain its model?	Understand	CO5	AITB03.16
14	Demonstrate the power and limitations of Turing machine.	Understand	CO5	AITB03.18
15	Construct Transition diagram for Turing Machine - $L=\{a^nb^nc^n/n>=1\}$	Apply	CO5	AITB03.16
16	Construct a Transition diagram for Turing Machine to implement addition of two unary numbers(X+Y).	Apply	CO5	AITB03.16
17	Construct a Linear Bounded automata for a language where $L=\{a^nb^n/n>=1\}$	Apply	CO5	AITB03.16
18	Classify the types of Turing machines.	Understand	CO5	AITB03.18
19.	Describe briefly about the following a)Church's Hypothesis b)Counter machine	Understand	CO5	AITB03.16
20	Construct Transition diagram for Turing Machine that accepts the language $L = \{0^n1^n \mid n \geq 1\}$. Give the transition diagram for the Turing Machine obtained and also show the moves made by the Turing machine for the string 000111.	Apply	CO5	AITB03.16
Part – C (Problem Solving and Critical Thinking Questions)				
1	Construct a Turing Machine that accepts the language $L = \{a^{2n}b^n \mid n \geq 0\}$.Give the transition diagram for the Turing Machine obtained.	Apply	CO5	AITB03.16
2	Construct a Turing Machine that gives two's compliment for the given binary representation.	Apply	CO5	AITB03.16
3	Examine Type 3 and Type 2 grammars with example.	Analyze	CO5	AITB03.15
4	Extend the Type 1 and Type 0 grammars with example.	Understand	CO5	AITB03.17
5	Design a Turing Machine that accepts the set of all even palindromes over{0,1}	Create	CO5	AITB03.16
6	Design Turing Machine for $L=\{a^n\ b^n\ c^n \mid n \geq 1\}$	Create	CO5	AITB03.16
7	Construct Turing Machine to calculate GCD of two given numbers	Apply	CO5	AITB03.15
8	Compare and contrast the Finite state machine , PDA and Turing Machine	Understand	CO5	AITB03.17
9	Construct a Turing Machine to accept the following languages $L = \{ w^nx^ny^nz^n \mid n \geq 1\}$	Apply	CO5	AITB03.16
10	Design a Turing Machine that accepts the language denoted by regular expression (000)*	Create	CO5	AITB03.16

Prepared by:

Mr. P Anjaiah, Asstistant Professor

HOD, CSE.