



# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

## COMPUTER SCIENCE AND ENGINEERING

### DEFINITIONS AND TERMINOLOGY

Course Name	:	<b>THEORY OF COMPUTATION</b>
Course Code	:	<b>AIT002</b>
Program	:	<b>B.Tech</b>
Semester	:	<b>IV</b>
Branch	:	<b>Computer Science and Engineering</b>
Section	:	<b>A, B, C, D</b>
Academic Year	:	<b>2018– 2019</b>
Course Faculty	:	<b>Ms A Jayanthi, Assistant Professor, CSE Ms E Uma Shankari, Assistant Professor, CSE Ms T Ramya Sree, Assistant Professor, CSE Mr P Anjaiah, Assistant Professor, CSE</b>

### OBJECTIVES

I	To help students to consider in depth the terminology and nomenclature used in the syllabus.
II	To focus on the meaning of new words / terminology/nomenclature

## DEFINITIONS AND TERMINOLOGY QUESTION BANK

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
<b>UNIT - I</b>					
1	Define Automata?	It is an abstract model of digital computer. It has a mechanism for reading input. Automata enable the scientists to understand how machines compute the functions and solve problems.	Remember	CLO 1	AIT002.01
2	Define Symbol?	Symbol is the smallest building block, which can be any alphabet, letter or any picture. Ex: a, b, 0,1..	Remember	CLO 1	AIT002.01
3	Define Alphabet?	Alphabet is set of symbols, which are always finite. It is denoted by $\Sigma$ . Ex: {0, 1}, {a, b}, {a, b, c}.....	Remember	CLO 1	AIT002.01
4	What is String?	String is a finite sequence of symbols from some alphabet. String is generally denoted as w. Ex: String over $\Sigma = \{a,b\}$ is ababa	Remember	CLO 1	AIT002.01
4	What is length of string?	The number of symbols in the given string is called length of the string. length of a string is denoted as  w . Example:  abb =3 over $\Sigma = \{a,b\}$	Remember	CLO 1	AIT002.01
5	What is substring of a string?	Any sequence of 0 or more consecutive symbols of the given string over an alphabets is called substring. Example: For string abb the substrings are $\epsilon$ , a, ab, abb	Remember	CLO 1	AIT002.01
6	Define Null String?	Empty string is the string with zero occurrences of symbol, represented as $\epsilon$ .	Remember	CLO 1	AIT002.01
7	What is Prefix of a String?	A group of leading symbols of string is called as prefix. Ex: Prefixes of sting abc are $\epsilon$ , a, ab, abc	Remember	CLO 1	AIT002.01
8	What is Suffix of a String?	A group of trailing symbols of string is called as suffix. Ex: Suffixes of sting abc are $\epsilon$ , a, bc, abc	Remember	CLO 1	AIT002.01
9	What is proper Prefix of a String?	A group of leading symbols other than string is called as proper prefix. Ex: Proper Prefixes of sting abc are $\epsilon$ , a, ab	Remember	CLO 1	AIT002.01
10	What is proper Suffix of a String?	A group of trailing symbols other than the string is called as proper suffix. Ex: Proper Suffixes of sting abc are $\epsilon$ , a, bc	Remember	CLO 1	AIT002.01

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
12	What is power of alphabet?	Consider $\Sigma = \{0,1\}$ the following are the powers. 1. $\Sigma^0 = \{ \epsilon \}$ 2. $\Sigma^1 = \{ 0, 1 \}$ 3. $\Sigma^2 = \{00, 01, 10, 11 \}$ $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111 \}$	Remember	CLO 1	AIT002.01
13	What is kleenclosure?	The set of all strings over on alphabet. It is denoted as $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$	Remember	CLO 2	AIT002.02
14	What is positive closure?	The set of all strings except $\epsilon$ over on alphabet. It is denoted as $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$	Remember	CLO 2	AIT002.02
15	Define Language?	A language is a <i>set of strings</i> , chosen form some $\Sigma^*$ or A language is a subset of $\Sigma^*$ . A language which can be formed over ' $\Sigma$ ' can be Finite or Infinite. Language that contains strings over $\Sigma = \{a, b\}$ are $\{ \epsilon, a, b, aa, ab, \dots \}$	Remember	CLO 2	AIT002.02
16	What is grammar?	It contains set of rules to generate the strings of a language.	Remember	CLO 2	AIT002.02
17	Define formal languages?	Formal language is set of all strings where each string restricted over particular forms of strings. String with given input.	Remember	CLO 2	AIT002.02
18	What are the types of formal languages?	There are four types 1. Regular languages 2. Context Free languages 3. Context Sensitive languages Recursively Enumerable languages	Remember	CLO 2	AIT002.02
19	What are the types of automation?	There are four types: 1. Finite Automation 2. Push Down Automation 3. Linear Bound Automation Turing Machine	Remember	CLO 2	AIT002.02
20	What is finite automata?	An automation the accepts regular language is called finite automata.	Remember	CLO 2	AIT002.02
21	What are the types of finite automata?	There are the two types: 1. Deterministic Finite Automata (DFA) 2. Non- Deterministic Finite Automata (NDFFA) Both are equivalent in power.	Remember	CLO 2	AIT002.02
22	What is Push Down automata?	An automata the accepts context free language is called Push Down automata.	Remember	CLO 2	AIT002.02
23	What are the types of Push Down automata?	There are the two types: 1. Deterministic Push Down Automata (DPDA)	Remember	CLO 2	AIT002.02

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		2. Non- Deterministic Push Down Automata (NPDA)			
24	What is Linear Bound automata?	An automata the accepts context sensitive language is called Linear Bound automata.	Remember	CLO 2	AIT002.02
25	What are the types of Linear Bound automata?	There are the two types: 1. Deterministic Linear Bound Automata (DLBA) 2. Non- Deterministic Linear Bound Automata (NLBA)	Remember	CLO 2	AIT002.02
26	What is Turing Machine?	An automata that accepts recursively enumerable language is called Turing Machine.	Remember	CLO 2	AIT002.02
27	What are the types of Turing Machine?	There are the two types: 1. Deterministic Turing Machine (DTM) 2. Non- Deterministic Turing Machine (NTM)	Remember	CLO 2	AIT002.02
28	Define Finite Automata?	A finite automata consists of finite set of states and transitions from state to state occur on input symbols chosen from an alphabet.	Remember	CLO 2	AIT002.02
29	What is Initial state?	It is the starting state from which reading of input symbols from the string starts.	Remember	CLO 2	AIT002.02
30	What is Final state?	It is the last state reached by a accepted state means accepted string should reach final state.	Remember	CLO 2	AIT002.02
31	Define Transition?	It is the change of state on input symbol.	Remember	CLO 2	AIT002.02
32	Define Transition Table?	A transition table is a tabular representation of the transition function that takes two arguments and returns a state. The column contains the state in which the automaton will be on the input which we can take as row.	Remember	CLO 2	AIT002.02
33	What is Acceptance of String?	Any string starts from initial state and ends in final state then the string is accepted.	Remember	CLO 2	AIT002.02
34	What is Rejection of String?	Any string starts from initial state and ends in non-final state then the string is rejected.	Remember	CLO 2	AIT002.02
35	What are Applications of Finite Automata?	A Finite Automata is highly useful in designing Lexical Analyzers, Text editors, Spell checkers, etc.	Remember	CLO 2	AIT002.02
36	What is Deterministic Finite Automata?	In a DFA, for a particular input character, the machine goes to one state only. A transition function is defined on every state for every input symbol. Also in DFA null (or $\epsilon$ ) move is not allowed, i.e., DFA cannot change state without any input character.	Remember	CLO 3	AIT002.03
37	What is Non Deterministic Finite Automata?	NFA is similar to DFA except following features: 1. Null (or $\epsilon$ ) move is allowed i.e., it can move forward without reading symbols. 2. Ability to transmit to any number of states for a particular input.	Understand	CLO 3	AIT002.03

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		However, these above features don't add any power to NFA. If we compare both in terms of power, both are equivalent.			
38	What is the purpose of epsilon?	Epsilon is used to change the state without seeing input symbol.	Remember	CLO 3	AIT002.03
39	What is the need of converting NFA to DFA?	In DFA, We can determine exact state on input symbol. So, ambiguity is not there and for string validation no backtracking is needed.	Remember	CLO 3	AIT002.03
40	What is the need for finite automata?	Finite automata are very useful for communication protocols and for matching strings against regular expressions. Finite automata are e.g. used to parse formal languages.	Remember	CLO 4	AIT002.04
41	Write the specification of DFA.	So a DFA is mathematically represented as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ 1. $Q$ - finite set of states 2. $\Sigma$ - finite set of input symbols 3. $\delta$ - transition function that takes as argument a state and a symbol and returns a state 4. $q_0$ - start state 5. $F$ - set of final or accepting states The transition function $\delta$ is a function in $Q \times \Sigma \rightarrow Q$	Remember	CLO 3	AIT002.03
42	Define transition diagram.	A diagram consisting of circles to represent states and directed line segments to represent transitions between the states. One or more actions (outputs) may be associated with each transition. The diagram represents a finite state machine.	Remember	CLO 3	AIT002.03
43	Construct the language that accepts the string of length 3 over alphabet $\Sigma = \{a,b\}$ .	$L = \{aaa, aba, aab, abb, baa, bab, bba, bbb\}$ .	Remember	CLO 3	AIT002.03
44	Define $\epsilon$ -NFA.	We extend the class of NFAs by allowing instantaneous ( $\epsilon$ ) transitions: 1. The automaton may be allowed to change its state without reading the input symbol. 2. In diagrams, such transitions are depicted by labeling the appropriate arcs with $\epsilon$ . 3. Note that this does not mean that $\epsilon$ has become an input symbol. On the contrary, we assume that the symbol $\epsilon$ does not belong to any alphabet.	Remember	CLO 3	AIT002.03
45	Define $\epsilon$ closure.	Epsilon means present state can go to other state without any input. This can happen only if the present state have epsilon transition to other state. Epsilon closure is finding all the states which can be reached from the present state on one or more epsilon transitions.	Remember	CLO 4	AIT002.04
46	Write the specification of NFA.	So a DFA is mathematically represented as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ 1. $Q$ - finite set of states 2. $\Sigma$ - finite set of input symbols 3. $\delta$ - transition function that takes as argument a state and a symbol and returns a state 4. $q_0$ - start state	Remember	CLO 4	AIT002.04

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		5. F-set of final or accepting states The transition function $\delta$ is a function in $Q \times \Sigma \rightarrow 2^Q$			
47	Write the specification of NFA- $\epsilon$ .	So a DFA is mathematically represented as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ 1. Q- finite set of states 2. $\Sigma$ - finite set of input symbols 3. $\delta$ - transition function that takes as argument a state and a symbol and returns a state 4. $q_0$ - start state 5. F-set of final or accepting states The transition function $\delta$ is a function in $Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$	Remember	CLO 4	AIT002.04
48	Construct the language that accepts the string of length 2 over alphabet $\Sigma = \{a,b\}$ .	$L = \{aa, ab, ba, bb\}$ .	Remember	CLO 2	AIT002.02
49	What is finite language?	The language which contains finite number of strings. Example: Strings of length 2 over alphabet $\Sigma = \{a,b\}$ is $L = \{aa, ab, ba, bb\}$ .	Remember	CLO 2	AIT002.02
50	What is infinite language?	The language which contains infinite number of strings. Example: Strings of length $\geq 2$ over alphabet $\Sigma = \{a,b\}$ is $L = \{aa, ab, ba, bb, aaa, aba, \dots\}$ .	Remember	CLO 2	AIT002.02
51	Construct the language that accepts the string of length 1 over alphabet $\Sigma = \{a,b\}$ .	$L = \{a,b\}$	Remember	CLO 2	AIT002.02
<b>UNIT – II</b>					
1	What is a regular expression?	A regular expression is a string that describes the whole set of strings according to certain syntax rules.	Remember	CLO 5	AIT002.05
2	Define acceptor?	An automaton that computes a Boolean function is called an acceptor. All the states of an acceptor is either accepting or rejecting the inputs given to it.	Remember	CLO 5	AIT002.05
3	What is a regular language?	A language accepted by the regular expression is called as Regular Language.	Remember	CLO 5	AIT002.05
4	What is a regular set?	A regular language which is accepted by finite automata is called regular set.	Remember	CLO 5	AIT002.05
5	What is Kleen closure?	Kleen closure is defined as $*$ that is 0 or more occurrence.	Remember	CLO 5	AIT002.05
6	What is positive closure ?	positive closure is defined as $+$ that is 1 or more occurrence.	Remember	CLO 5	AIT002.05
7	Identify the equivalent identity rule for $RR^*$	The equivalent identity rule for $RR^*$ is $R^*R$	Remember	CLO 5	AIT002.05

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
8	Identify the equivalent identity rule for $(R^*)^*$	The equivalent identity rule for $(R^*)^*$ is $R^*$	Remember	CLO 5	AIT002.05
9	Identify the equivalent identity rule for $R^*R^*$	The equivalent identity rule for $R^*R^*$ is $R^*$	Remember	CLO 5	AIT002.05
10	Identify the equivalent identity rule for $R+R$	The equivalent identity rule for $R+R$ is $R$ (Idempotent law)	Remember	CLO 5	AIT002.05
11	Identify the equivalent identity rule for $R\epsilon$ or $\epsilon R$	The equivalent identity rule for $R\epsilon$ or $\epsilon R$ is $R$	Remember	CLO 5	AIT002.05
12	Identify the equivalent identity rule for $\emptyset L$ or $L \emptyset$	The equivalent identity rule for $\emptyset L$ or $L \emptyset$ is $\emptyset$	Remember	CLO 5	AIT002.05
13	Identify the equivalent identity rule for $\epsilon+RR^*$	The equivalent identity rule for $\epsilon+RR^*$ is $R^*$	Remember	CLO 5	AIT002.05
14	Identify the equivalent identity rule for $(PQ)^*P$	The equivalent identity rule for $(PQ)^*P$ is $P(QP)^*$	Remember	CLO 5	AIT002.05
15	What are the applications of pumping lemma?	Pumping lemma is used to check if a language is regular or not. (i) Assume that the language(L) is regular. (ii) Select a constant 'n'. (iii) Select a string(z) in L, such that $ z >n$ . (iv) Split the word z into u,v and w such that $ uv \leq n$ and $ v \geq 1$ . (v) You achieve a contradiction to pumping lemma that there exists an 'i'	Remember	CLO 7	AIT002.07
16	What is Arden's Theorem?	Arden's theorem helps in checking the equivalence of two regular expressions. The regular expression R is given as : $R=Q+RP$ Which has a unique solution as $R=QP^*$ .	Remember	CLO 6	AIT002.06
17	What is the closure property of regular sets?	The regular sets are closed under union, concatenation and Kleene closure. $r_1 \cup r_2 = r_1 + r_2$ $r_1.r_2 = r_1r_2$ ( $r$ ) $^*=r^*$ The class of regular sets are closed under complementation, substitution, homomorphism and inverse homomorphism.	Remember	CLO 5	AIT002.05
18	Identify the RE for string length exactly 2?	The regular expression for strings of length exactly 2 is $(a+b)(a+b)$	Understand	CLO 4	AIT002.04
19	Identify the RE for string length exactly 3	The regular expression for strings of length exactly 3 is $(a+b)(a+b)(a+b)$	Understand	CLO 4	AIT002.04
20	Identify the RE for string length atleast 2	The regular expression for strings of length atleast 2 is $(a+b)(a+b)(a+b)^*$	Understand	CLO 4	AIT002.04

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
21	Identify the RE for string length atmost 2	The regular expression for strings of length exactly 2 is $(a+b+ \epsilon)(a+b+ \epsilon)$	Understand	CLO 4	AIT002.04
22			Understand	CLO 4	AIT002.04
23	Identify the regular set for given regular expression $(0 + 10^*)$	Regular set is $L = \{ 0, 1, 10, 100, 1000, 10000, \dots \}$	Understand	CLO 4	AIT002.04
24	Identify the regular set for given regular expression $(0^*10^*)$	Regular set is $L = \{1, 01, 10, 010, 0010, \dots\}$	Understand	CLO 4	AIT002.04
25	Identify the regular set for given regular expression $(0 + \epsilon)(1 + \epsilon)$	Regular set is $(0 + \epsilon)(1 + \epsilon)$	Understand	CLO 4	AIT002.04
26	Identify the regular set for given regular expression $(11)^*$	Set consisting of even number of 1's including empty string, So $L = \{\epsilon, 11, 1111, 111111, \dots\}$	Understand	CLO 4	AIT002.04
27	Identify the regular set for given regular expression $(aa)^*(bb)^*b$	Set of strings consisting of even number of a's followed by odd number of b's , so $L = \{b, aab, aabbb, aabbbb, aaaab, aaaabbb, \dots\}$	Understand	CLO 4	AIT002.04
28	Identify the regular set for given regular expression $(aa + ab + ba + bb)^*$	String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so $L = \{aa, ab, ba, bb, aab, aba, \dots\}$	Understand	CLO 4	AIT002.04
29	Identify the regular set for given regular expression $(a+b)^*abb$	Set of strings of a's and b's ending with the string abb. So $L = \{abb, aabb, babb, aaabb, ababb, \dots\}$	Understand	CLO 4	AIT002.04
30	Identify the RE for strings which begin or end with either 00 or 11.	The regular expression for begins and ends with 00 or 11 is $=[(00+11)(0+1)^*] + [(0+1)^*(00+11)]$	Understand	CLO 4	AIT002.04
31	Identify the RE for strings with atleast two c's over the set $\Sigma = \{c,b\}$	The regular expression for strings with atleast two c's: $(b+c)^* c (b+c)^* c (b+c)^*$	Understand	CLO 4	AIT002.04
32	Identify the regular set for given regular expression $(0+1)^*$	Any combinations of 0's and 1's. $(0+1)^* = \{ \epsilon, 0, 1, 01, 10, 001, 101, 101001, \dots \}$	Understand	CLO 4	AIT002.04
33	Identify the regular set for given regular expression $(0+1)^+$	Any combinations of 0's and 1's. $(0+1)^+ = \{ 0, 1, 01, 10, 001, 101, 101001, \dots \}$	Understand	CLO 4	AIT002.04
34	Identify the regular expression for even length of string.	Regular expression for even length of string $R = ((a+b)(a+b))^*$	Understand	CLO 4	AIT002.04



S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
35	Identify the regular expression for odd length of string.	Regular expression for odd length of string $R = ((a+b)(a+b))^*(a+b)$	Understand	CLO 4	AIT002.04
36	Identify the regular expression for all strings beginning with '1' and ending with 'ab'.	Regular expression for all strings beginning with '1' and ending with 'ab'. $R = 11(1+a+b)^*ab$	Understand	CLO 4	AIT002.04
37	Identify the regular expression for every string will have atleast one 'a' followed by atleast one 'b'.	The regular expression for every string will have atleast one 'a' followed by atleast one 'b'. $R = a^+b^+$	Understand	CLO 4	AIT002.04
38	State the operations of regular expressions	The operations of regular expressions are Union, concatenation and kleen closure.	Remember	CLO 4	AIT002.04
39	Identify the regular expression for set of all strings over {a,b} with 3 consecutive b's.	the regular expression for set of all strings over {a,b} with 3 consecutive b's. R.E: $(a+b)^*bbb(a+b)^*$	Understand	CLO 4	AIT002.04
40	State Left distributive law in identity rules	Left distributive law is $P(Q+R) = PQ + PR$	Understand	CLO 4	AIT002.04
41	State Right distributive law in identity rules	Right distributive law is $(Q+R)P = QP + RP$	Understand	CLO 4	AIT002.04
42	Identify the properties under which regular languages are not closed.	Subset, superset, infinite union and infinite intersection.	Understand	CLO 4	AIT002.04
<b>UNIT – III</b>					
1	What are the applications of Context free languages?	Context free languages are used in: a)Defining programming languages. b)Formalizing the notion of parsing. c)Translation of programming languages. d)String processing applications.	Remember	CLO 9	AIT002.09
2	Define a context free grammar	A context free grammar (CFG) is denoted as $G=(V,T,P,S)$ where V = Finite non-empty set of variables / non-terminal symbols T = Finite set of terminal symbols P = Finite non-empty set of production rules of the form $A \rightarrow \alpha$ where $A \in V$ and $\alpha \in (V \cup T)^*$ S = Start symbol	Remember	CLO 9	AIT002.09
3	Define left-most Derivation?	In the left most derivation, the input is scanned and replaced with the production rule from left to right. So in left most derivatives read the input string from left to right.	Remember	CLO 9	AIT002.09
4	What is right-most Derivation?	In the right most derivation, the input is scanned and replaced with the production rule from right to left. So in right most derivatives read the input string from right to left.	Remember	CLO 9	AIT002.09

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
5	What is CFL?	L is a context free language (CFL) if it is L(G) for some CFG G..	Remember	CLO 9	AIT002.09
6	What is Derivation?	Derivation is a sequence of production rules. It is used to get the input string through these production rules	Remember	CLO 9	AIT002.09
7	Define parse tree?	Parse tree is the graphical representation of symbol. The symbol can be terminal or non-terminal. In parsing, the string is derived using the start symbol. The root of the parse tree is that start symbol.	Remember	CLO 9	AIT002.09
8	Define subtree?	A subtree of a derivation tree is a particular vertex of the tree together with all its descendants, the edges connecting them and their labels. The label of the root may not be the start symbol of the grammar.	Remember	CLO 9	AIT002.09
9	If $S \rightarrow aSb \mid aAb$ , $A \rightarrow bAa$ , $A \rightarrow ba$ . Find out the CFL	$S \rightarrow aAb \Rightarrow abab$ $S \rightarrow aSb \Rightarrow a aAb b \Rightarrow a a ba b b$ (sub $S \rightarrow aAb$ ) $S \rightarrow aSb \Rightarrow a aSb b \Rightarrow a a aAb b b \Rightarrow a a ba b bb$ Thus $L = \{ a^n b^m a^n \mid n, m \geq 1 \}$	Analyze	CLO 10	AIT002.09
10	Define ambiguous grammar?	A grammar is said to be ambiguous if it has more than one derivation trees for a sentence or in other words if it has more than one leftmost derivation or more than one rightmost derivation	Remember	CLO 9	AIT002.09
12	Write the three ways to simplify a context free grammar?	a) By removing the useless symbols from the set of productions. b) By eliminating the empty productions. c) By eliminating the unit productions.	understand	CLO 10	AIT002.10
13	Write about normal form?	By reducing the grammar, although the grammar gets minimized but does not get standardized. This is because the RHS of productions have no specific format. In order to standardize the given grammar, we normalize it.	understand	CLO 10	AIT002.10
14	Differentiate sentences and sentential forms	A sentence is a string of terminal symbols. A sentential form is a string containing a mix of variables and terminal symbols or all variables. This is an intermediate form in doing a derivation.	understand	CLO 9	AIT002.09
15	Define Chomsky normal form?	A grammar is said to be Chomsky Normal Form (CNF), if all its productions must derive either two non-terminals or a single terminal	Remember	CLO 10	AIT002.10
16	State the pumping lemma for CFLs.	Let L be any CFL. Then there is a constant n, depending only on L, such that if z is in L and $ z  \geq n$ , then $z = uvwxy$ such that : (i) $ vx  \geq 1$ (ii) $ vwx  \leq n$ and (iii) for all $i \geq 0$ $u^i v^i w^i x^i y^i$ is in L.	analyze	CLO 12	AIT002.12

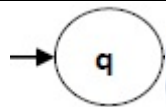

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
17	List the types of normal forms?	There are two types of normal form a)Chomsky Normal Form (CNF) b)Greibach Normal Form (GNF)	Remember	CLO 10	AIT002.10
18	What are the properties of Context free languages?	a)Context free languages are closed under union b) Context free languages are closed under concatenation c) Context free languages are closed under kleen closure d) Context free languages are closed under intersection e) Context free languages are closed under complement	Remember	CLO 9	AIT002.09
19	Define syntax tree?	Concrete syntax tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar.	Remember	CLO 9	AIT002.09
20	List the applications of CFG?	a)For defining programming languages b)For parsing the program by constructing syntax tree c)For translation of programming languages d)For describing arithmetic expression e)For construction of compilers	Remember	CLO 9	AIT002.09
21	What are decisions to be considered during derivation?	During parsing two decisions are taken. These are as follows: a) Decide the non-terminal which is to be replaced. b) Decide the production rule by which the non-terminal will be replaced.	Remember	CLO 10	AIT002.10
22	List the properties of Derivation tree?	a)The root node is always a node indicating start symbol b)The derivation is read from left to right c)The leaf nodes always terminals nodes d)The interior nodes are always non terminal nodes	Remember	CLO 10	AIT002.10
23	What is nonterminal?	Nonterminal symbols are those symbols which can be replaced. They may also be called simply syntactic variables. A formal grammar includes a start symbol, a designated member of the set of non-terminals from which all the strings in the language may be derived by successive applications of the production rules.	Remember	CLO 9	AIT002.09
24	Define Terminal?	Terminal symbols are literal symbols which may appear in the outputs of the production rules of a formal grammar and which cannot be changed using the rules of the grammar	Remember	CLO 9	AIT002.09
25	List different types of derivations?	There are two types as follows: a)Left-most Derivation b)Right-most Derivation	Remember	CLO 10	AIT002.09
26	What is root vertex?	Root vertex is always called as Non-terminal.It must be labeled by start symbol	Remember	CLO 10	AIT002.10
27	Define vertex and leaves in Derivation tree?	Vertex are labeled by a non terminal symbol Leaves are labeled by a terminal symbol or epsilon(empty symbol)	Remember	CLO 10	AIT002.10

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
28	List the approaches for representing Derivation tree?	There are two approaches a)Top-down approach b)Bottom-up approach	Remember	CLO 10	AIT002.10
29	Define Top-down approach?	This approach starts with the starting symbol S. and moves down from root node to leaf nodes using productions	Remember	CLO 10	AIT002.10
30	Define Bottom-up approach?	This approach start from leaf nodes .and it proceeds upwards to the root which is the starting symbol S	Remember	CLO 10	AIT002.10
31	What is partial Derivation tree?	A partial derivation tree is a sub tree of a derivation tree such that all of its children are in the subtree or none of them are in the subtree	Remember	CLO 10	AIT002.10
32	Define rule in Chomsky Normal form?	For converting into Chomsky normal form following rule is Non-terminal $\rightarrow$ Non-terminal.Non-terminal Non-terminal $\rightarrow$ terminal	Remember	CLO 10	AIT002.10
33	Define rule in Greibach Normal fom?	The rule for GNF as follows Nonterminal $\rightarrow$ one terminal.Any no of Non-terminals	Remember	CLO 10	AIT002.10
34	When CFG is in Chomsky normal form?	A context free grammar is in Chomsky normal form if all the production rules satisfy one of the following conditions. a)A non-terminal generating terminal b) A non-terminal generating two non- terminal c)start symbol generating epsilon	Remember	CLO 10	AIT002.10
35	What are the steps for converting CGF to CNF?	a)Eliminate start symbol from RHS b)Eliminate null, unit and useless productions c)Eliminate terminals from RHS if they exist with other terminals or non-terminals d)Eliminate RHS with more than two non-terminals.	Remember	CLO 11	AIT002.11
36	Define Context sensitive Grammar?	Context-sensitive grammar has 4-tuples. $G = \{N, \Sigma, P, S\}$ , Where N = Set of non-terminal symbols $\Sigma$ = Set of terminal symbols S = Start symbol of the production P = Finite set of productions All rules in P are of the form $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$	Remember	CLO 11	AIT002.11
37	What is Context Sensitive Language?	The language that can be defined by context-sensitive grammar is called CSL. Properties of CSL are : Union, intersection and concatenation of two context-sensitive languages is context-sensitive. Complement of a context-sensitive language is context-sensitive.	Remember	CLO 11	AIT002.11
38	What is Chomsky hierarchy?	In Chomsky hierarchy grammars are divided of four types: a)Unrestricted grammar	Remember	CLO 10	AIT002.10

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		b)Context Sensitive grammar c)Context free grammar d)Regular grammar			
39	Which grammar is more powerful in Chomsky hierarchy?	Context-sensitive grammars are more powerful than context-free grammars because there are some languages that can be described by CSG but not by context-free grammars and CSL are less powerful than Unrestricted grammar. Due to this context-sensitive grammars are positioned between context-free and unrestricted grammars in the Chomsky hierarchy.	understand	CLO 10	AIT002.10
40	Distinguish between Regular grammar and CFG?	a)Regular grammar is used for drawing DFA whereas CFG is used for constructing push down automata(PDA) b)Compiler cannot make use of this Regular grammar for parsing whereas Compiler make use of CFG for parsing	understand	CLO 11	AIT002.11
41	Construct the CFG for the regular expression (0+1)*	The CFG can be given by $P = \{S \rightarrow 0S \mid 1S$ $S \rightarrow \epsilon$	analyze	CLO 10	AIT002.10
42	How many cases are required to obtain CFG for unequal no of a's and b's?	a)Only a's are present and number of b's are zero b) Only b's are present and number of a's are zero c)Number of a's are atleast one more than number of b's d) Number of b's are atleast one more than number of a's	analyze	CLO 10	AIT002.10
43	Construct CFG which consist of all the strings having atleast one occurrence of 000?	Production rules are: $S \rightarrow ATA$ $A \rightarrow 0A \mid 1A \mid \epsilon$ $T \rightarrow 000$	analyze	CLO 10	AIT002.10
44	State the Pumping lemma for Regular languages?	For any regular language L, there exists an integer n, such that for all $x \in L$ with $ x  \geq n$ , there exists $u, v, w \in \Sigma^*$ , such that $x = uvw$ , and a) $ uv  \leq n$ b) $ v  \geq 1$ c) for all $i \geq 0$ : $uiv^i w \in L$	analyze	CLO 12	AIT002.12
45	Define useless productions?	The productions that can never take part in derivation of any string, are called useless productions. Similarly, a variable that can never take part in derivation of any string is called a useless variable	Remember	CLO 10	AIT002.10
46	What is Elimination of null productions?	The productions of type ' $A \rightarrow \epsilon$ ' are called $\epsilon$ productions (null productions). These productions can only be removed from those grammars that do not generate $\epsilon$ (an	Remember	CLO 10	AIT002.10

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		empty string). It is possible for a grammar to contain null productions and yet not produce an empty string.			
47	Write the steps required to remove Unit productions?	A unit production is a production $A \rightarrow B$ where both A and B are non-terminals. Unit productions are redundant and hence should be removed by using following steps a) Select a unit production $A \rightarrow B$ , such that there exist a production $B \rightarrow \alpha$ , where $\alpha$ is a terminal b) For every non-unit production, $B \rightarrow \alpha$ repeat the following step Add production $A \rightarrow \alpha$ to the grammar c) Eliminate $A \rightarrow B$ from the grammar	understand	CLO 10	AIT002.10
48	Define Derivation tree?	A derivation tree or parse tree is an ordered rooted tree that graphically represents the semantic information a string derived from a context-free grammar.	Remember	CLO 10	AIT002.10
49	List the representation technique for Derivation trees?	Root vertex – Must be labeled by the start symbol. Vertex – Labeled by a non-terminal symbol. Leaves – Labeled by a terminal symbol or $\epsilon$ .	Remember	CLO 10	AIT002.10
50	What is sentential form?	A sentential form is the start symbol S of a grammar or any string in $(V \cup T)^*$ that can be derived from S. A string of terminals and variables $\alpha$ is called a sentential form if: $S \Rightarrow \alpha$ where S is the start symbol of the grammar	Remember	CLO 10	AIT002.10
51	What are the steps for converting CGF to GNF?	a) Convert the grammar into CNF. b) Eliminate left recursion from grammar if it exists. c) Convert the production rules into GNF form.	Remember	CLO 10	AIT002.10
<b>UNIT - IV</b>					
1	What is the PDA? why it is developed ?	PDA is a tool to implement CFLs .The FA has no memory to remember the count of input symbols or to match the relationship between the different types of symbols occurring in the string, that's why PDA is developed	Remember	CLO 14	AIT002.14
2	Define Push Down Automata?	A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.  Basically a pushdown automaton is –  " <b>Finite state machine</b> " + " <b>a stack</b> "  A pushdown automaton has three components –	Remember	CLO 14	AIT002.14

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		<ul style="list-style-type: none"> <li>• an input tape,</li> <li>• a control unit, and</li> <li>• a stack with infinite size.</li> </ul>			
3	How to represent the initial store of PDA ?	A Special pushdown symbol called the initial symbol on push down store represented by $Z_0$	Remember	CLO 14	AIT002.14
4	How to represent the PDA	$PDA M = (Q, \Gamma, \Sigma, \delta, q_0, Z_0, F)$	Remember	CLO 14	AIT002.14
5	How to represent transition function in PDA	$\delta : Q \times \Sigma \times \Gamma \Rightarrow Q \times \Gamma$	Remember	CLO 14	AIT002.14
6	Name the 7 tuples of PDA	<p><math>Q</math> = Set Of All States</p> <p><math>\Gamma</math> = stack symbols</p> <p><math>\Sigma</math> = input alphabet</p> <p><math>\delta</math> = transition function</p> <p><math>q_0</math> = start state</p> <p><math>Z_0</math> = Initial stack top symbol</p> <p><math>F</math> = Final/accepting states</p>	Remember	CLO 14	AIT002.14
7	How many operations are performed on the stack ?	<p>A stack does two operations</p> <ul style="list-style-type: none"> <li>• <b>Push</b> – a new symbol is added at the top.</li> <li>• <b>Pop</b> – the top symbol is read and removed.</li> </ul>	Remember	CLO 13	AIT002.13
8	How to represent the empty stack initially?	Initially we put a special symbol ' $\$$ ' into the empty stack.	Remember	CLO 13	AIT002.13
9	Define Deterministic Context-free Language?	Deterministic CFL are subset of CFL which can be recognized by Deterministic PDA. Deterministic PDA has only one move from a given state and input symbol.	Remember	CLO 15	AIT002.15
10	Show the notation of the empty string in PDA	If $ S = 0$ , it is called an <b>empty string</b> (Denoted by $\lambda$ or $\epsilon$ )	Remember	CLO 14	AIT002.14

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
11	Show the notation of the input state in PDA ?		Remember	CLO 14	AIT002.14
12	What is meant by empty stack acceptability	<p>A PDA accepts a string when, after reading the entire string, the PDA has emptied its stack.</p> <p>For a PDA <math>(Q, \Sigma, S, \delta, q_0, I, F)</math>, the language accepted by the empty stack is –</p> $L(\text{PDA}) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$	Remember	CLO 13	AIT002.13
13	Show the notation of the output state in PDA ?		Remember	CLO 14	AIT002.14
14	Define acceptance of PDA by Final State	<p>In final state acceptability, a PDA accepts a string when, after reading the entire string, the PDA is in a final state. From the starting state, we can make moves that end up in a final state with any stack values. The stack values are irrelevant as long as we end up in a final state.</p> <p>For a PDA <math>(Q, \Sigma, S, \delta, q_0, I, F)</math>, the language accepted by the set of final states F is –</p> $L(\text{PDA}) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, x), q \in F\}$ <p>for any input stack string <math>x</math>.</p>	Remember	CLO 14	AIT002.14
15	Define acceptance of PDA by empty Stack?	<p>In final state acceptability, a PDA accepts a string when, after reading the entire string, the PDA is in a final state.</p> <p>For a PDA <math>(Q, \Sigma, S, \delta, q_0, I, F)</math>, the language accepted by the empty stack is –</p>	Remember	CLO 14	AIT002.14



S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		$L(PDA) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$			
16	Show the notation of the transition symbol between two states in PDA ?	$\xrightarrow{\quad}$	Remember	CLO 14	AIT002.14
17	Why input tape is used in PDA ?	Input tape is a buffer ,which is divided into cells and each cell stores a one input symbol from the give string input symbols are read by control head from left to right. \$ is placed at the end of the string.	Remember	CLO 14	AIT002.14
18	Why stack is used in the PDA ?	finite control reads the input symbols and places them in stack which is last in first out data structure	Remember	CLO 14	AIT002.14
19	Why pop operation is used in PDA ?	while reading the input symbols from input tape based on the condition top of the stack symbol is popped from stack to recognize a particular string	Remember	CLO 14	AIT002.14
20	why push operation is used in PDA ?	While reading the symbols from the input tape finite control will push them into the top of the stack.	Remember	CLO 14	AIT002.14
21	What is Non Deterministic PDA ?	Conceptually NPDA is similar to NFA. in NPDA there could be multiple transitions for a unique input. Construction of NPDA is much more simplest than PDA	Remember	CLO 14	AIT002.14
22	What is The Condition for converting CFG to PDA ?	in a production rule first symbol on the RHS is terminal symbol when we want to convert CGF to PDA	Remember	CLO 14	AIT002.14
23	Why two stacks are used in PDA ?	There are certain languages which cannot be accepted by PDA with single stack. This is a special case and when we want to perform two checks on the input strings we will use two stacks with PDA	Remember	CLO 14	AIT002.14
24	Why finite automata is less powerful than PDA	since no stack concept is used in FA, That's why PDA is more powerful than FA	Remember	CLO 14	AIT002.14
25	What is instantaneous description?	Instantaneous description of a PDA is a triple of three parameters triple(q,w,y) -q is the state - w is the remaining input string -y is the stack contents	Remember	CLO 14	AIT002.14
26	show the PDA machine configuration	A machine configuration is an element of $K \times \Sigma^* \times \Gamma^*$ . $(p,w,\gamma) = (\text{current state, unprocessed input, stack content}).$	Remember	CLO 14	AIT002.14
27	show the condition to accept the string in PDA	A string w is accepted by a PDA if and only if $(s,w, e) \vdash^* (f, e, e)$	Remember	CLO 14	AIT002.14

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
28	What are the different types of language acceptances by a PDA and define them.	For a PDA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ we define : Language accepted by final state $L(M)$ as:  $* \{ w \mid (q_0, w, Z_0) \vdash^* (p, \epsilon, \gamma) \text{ for some } p \text{ in } F \text{ and } \gamma \text{ in } \Gamma^* \}$ .  Language accepted by empty / null stack $N(M)$ is: *	Remember	CLO 14	AIT002.14
29	What is the significance of PDA?	Finite Automata is used to model regular expression and cannot be used to represent non regular languages. Thus to model a context free language, a Pushdown Automata is used.	Remember	CLO 14	AIT002.14
30	When is a string accepted by a PDA?	The input string is accepted by the PDA if: The final state is reached .The stack is empty.	Remember	CLO 14	AIT002.14
31	Give examples of languages handled by PDA.	(1) $L=\{ a^n b^n \mid n \geq 0 \}$ , here $n$ is unbounded , hence counting cannot be done by finite memory. So we require a PDA ,a machine that can count without limit.  (2) $L= \{ ww^R \mid w \in \{a,b\}^* \}$ , to handle this language we need unlimited counting capability .	Remember	CLO 14	AIT002.14
32	Is NPDA and DPDA equivalent?	The languages accepted by NPDA and DPDA are not equivalent. For example: $ww^R$ is accepted by NPDA and not by any DPDA.	Remember	CLO 14	AIT002.14
33	State the equivalence of acceptance by final state and empty stack.	If $L = L(M_2)$ for some PDA $M_2$ , then $L = N(M_1)$ for some PDA $M_1$ . If $L = N(M_1)$ for some PDA $M_1$ ,then $L = L(M_2)$ for some PDA $M_2$ .  where $L(M)$ = language accepted by PDA by reaching a final state. $N(M)$ = language accepted by PDA by empty stack.	Remember	CLO 14	AIT002.14
34	State the equivalence of PDA and CFL.	If $L$ is a context free language, then there exists a PDA $M$ such that $L=N(M)$ .If $L$ is $N(M)$ for some PDA $m$ , then $L$ is a context free language.	Remember	CLO 15	AIT002.15
35	What are the closure properties of CFL?	CFL are closed under union, concatenation and Kleene closure. CFL are closed under substitution , homomorphism.	Remember	CLO 15	AIT002.15

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		CFL are not closed under intersection , complementation.  Closure properties of CFL's are used to prove that certain languages are not context free.			
36	State the pumping lemma for CFLs.	Let L be any CFL. Then there is a constant n, depending only on L, such that if z is in L and $ z  \geq n$ , then $z=uvwxy$ such that :  (i) $ vx  \geq 1$  (ii) $ vwx  \leq n$ and  (iii) for all $i \geq 0$ $uviwx^iy$ is in L.	Remember	CLO 15	AIT002.15
37	What is the main application of pumping lemma in CFLs?	The pumping lemma can be used to prove a variety of languages are not context free . Some examples are:  $L1 = \{ a^i b^i c^i \mid i \geq 1 \}$ is not a CFL.  $L2 = \{ a^i b^i c^i d^j \mid i \geq 1 \text{ and } j \geq 1 \}$ is not a CFL.	Remember	CLO 15	AIT002.15
38	Give an example of Deterministic CFL.	The language $L = \{ a^n b^n \mid n \geq 0 \}$ is a deterministic CFL	Remember	CLO 15	AIT002.15
39	What are the properties of CFL?	Let $G=(V,T,P,S)$ be a CFG, The fanout of G , $\Phi(G)$ is largest number of symbols on the RHS of any rule in R. The height of the parse tree is the length of the longest path from the root to some leaf.	Remember	CLO 15	AIT002.15
40	What are the components of PDA ?	The PDA usually consists of four components: A control unit, A Read Unit, An input tape and A Memory unit.	Remember	CLO 14	AIT002.14
41	What is the informal definition of PDA?	A PDA is a computational machine to recognize a Context free language. Computational power of PDA is between Finite automaton and Turing machines. The PDA has a finite control , and the memory is organized as a stack.	Remember	CLO 15	AIT002.15

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
42	Give an example of NonDeterministic CFL	The language $L = \{ ww^R : w \in \{a,b\}^+ \}$ is a nondeterministic CFL.	Remember	CLO 15	AIT002.15
43	When DPDA becomes a PDA?	A Deterministic Push Down Automata is a Push Down Automata in which no state $p$ has two or more transitions.	Remember	CLO 14	AIT002.14
44	The transition of a Push down automaton makes is additionally dependent upon the what parameters ?	A PDA is a finite machine which has an additional stack storage. Its transitions are based not only on input and the correct state but also on the stack.	Remember	CLO 14	AIT002.14
45	With reference of a DPDA, what to perform from the start state with an empty stack?	The empty stack in the end is our requirement relative to finite state automaton.	Remember	CLO 14	AIT002.14
46	A language accepted by Deterministic Push down automata is closed under which operations	Deterministic Context free languages (one accepted by PDA by final state), are drastically different from the context free languages. For example they are closed under complementation and not union.	Remember	CLO 14	AIT002.14
47	What are different ways to represent A push down automata ?	a PDA can be represented using a transition diagram, transition table and an instantaneous description.	Remember	CLO 14	AIT002.14
48	Push down automata accepts which language?	Context free language	Remember	CLO 15	AIT002.15
49	Define deterministic CFL	A CFL that is accepted by DPDA is known as deterministic CFL	Remember	CLO 15	AIT002.15
50	What is deterministic PDA ?	A deterministic PDA is one in which every transition $\delta(q_i, a, Y)$ has at the most one implication	Remember	CLO 14	AIT002.14
<b>UNIT - V</b>					
1	Turing machine	A Turing machine consists of a tape of infinite length on which read and writes operation can be performed. The tape consists of infinite cells on which each cell either contains input symbol or a special symbol called blank. It also consists of a head pointer which points to cell currently being read and it can move in both directions.	Remember	CLO17	AIT002.17
2	recursively enumerable languages	A language is recursively enumerable if there exists a Turing machine that accepts every string of the language, and does not accept strings that are not in the language.	Remember	CLO17	AIT002.17
3	recursive languages	A language is <i>recursive</i> if there exists a Turing machine that accepts every string of the language and rejects every string that is not in the language.	Remember	CLO17	AIT002.17
4	Church's hypothesis	The Church-Turing thesis (formerly commonly known simply as Church's thesis) says that any real-world computation can be translated into an equivalent computation involving a Turing machine.	Remember	CLO17	AIT002.17

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
5	counter machine	Counter machine has the same structure as the multi-stack machine but in place of each stack is a counter. Counters hold any non-negative integer, but we can only distinguish between zero and nonzero counters. That's the move of the counter machine depends on its state, input symbol and which if any of the counters are zero.	Remember	CLO17	AIT002.17
6	linear bounded automata	A linear bounded automaton is a multi-track non-deterministic Turing machine with a tape of some bounded finite length.  Length = function (Length of the initial input string, constant c)  Here,  Memory information $\leq c \times$ Input information  The computation is restricted to the constant bounded area. The input alphabet contains two special symbols which serve as left end markers and right end markers which mean the transitions neither move to the left of the left end marker nor to the right of the right end marker of the tape.	Remember	CLO16	AIT002.16
7	context sensitive language	<b>Type-1 grammars</b> generate context-sensitive languages.	Remember	CLO16	AIT002.16
8	Chomsky hierarchy of languages	One way showing the relationship among the languages if Chomsky hierarchy.  Type 0 known as unrestricted grammar.  Type 1 known as context sensitive grammar.  Type 2 known as context free grammar.  Type 3 Regular Grammar.	Remember	CLO16	AIT002.16
9	Regular Grammar	<b>Type-3 grammars</b> generate regular languages. Type-3 grammars must have a single non-terminal on the left-hand side and a right-hand side consisting of a single terminal or single terminal followed by a single non-terminal.  The productions must be in the form $X \rightarrow a$ or $X \rightarrow aY$  where $X, Y \in N$ (Non terminal)	Remember	CLO16	AIT002.16

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		<p>and <math>a \in T</math> (Terminal)</p> <p>The rule <math>S \rightarrow \epsilon</math> is allowed if <math>S</math> does not appear on the right side of any rule.</p>			
10	Type-2 grammars	<p><b>Type-2 grammars</b> generate context-free languages.</p> <p>The productions must be in the form <math>A \rightarrow \gamma</math></p> <p>where <math>A \in N</math> (Non terminal)</p> <p>and <math>\gamma \in (T \cup N)^*</math> (String of terminals and non-terminals).</p> <p>These languages generated by these grammars are be recognized by a non-deterministic pushdown automaton.</p>	Remember	CLO19	AIT002.19
11	Context-free grammar	<p>A context-free grammar (CFG) consisting of a finite set of grammar rules is a quadruple <math>(N, T, P, S)</math> where</p> <ul style="list-style-type: none"> <li><math>N</math> is a set of non-terminal symbols.</li> <li><math>T</math> is a set of terminals where <math>N \cap T = \text{NULL}</math>.</li> <li><math>P</math> is a set of rules, <math>P: N \rightarrow (N \cup T)^*</math>, i.e., the left-hand side of the production rule <math>P</math> does not have any right context or left context.</li> </ul> <p><math>S</math> is the start symbol.</p>	Remember	CLO19	AIT002.19
12	Context-sensitive grammar	<p><b>Context sensitive grammars</b> generates context-sensitive languages.</p> <p>The productions must be in the form: <math>\alpha \rightarrow \beta</math>, with the following condition</p> <p><math> \beta  \geq  \alpha </math>, i.e. length of <math>\beta</math> is greater than length of <math>\alpha</math>.</p>	Remember	CLO19	AIT002.19
13	Unrestricted grammar	<p><b>Type-0 grammars</b> generate recursively enumerable languages. The productions have no restrictions. They are any phase structure grammar including all formal grammars.</p> <p>They generate the languages that are recognized by a Turing machine.</p>	Remember	CLO19	AIT002.19
14	Pushdown automata	<p>A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.</p>	Remember	CLO19	AIT002.19

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		<p>Basically a pushdown automaton is –</p> <p><b>"Finite state machine" + "a stack"</b></p> <p>A pushdown automaton has three components –</p> <ul style="list-style-type: none"> <li>• an input tape,</li> <li>• a control unit, and</li> <li>• a stack with infinite size.</li> </ul> <p>The stack head scans the top symbol of the stack.</p>			
15	Multi-tape Turing Machine	Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head. Each head can move independently of the other heads. Initially the input is on tape 1 and others are blank. At first, the first tape is occupied by the input and the other tapes are kept blank. Next, the machine reads consecutive symbols under its heads and the TM prints a symbol on each tape and moves its heads.	Remember	CLO19	AIT002.19
16	Turing machine with multiple tapes	Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head. Each head can move independently of the other heads. Initially the input is on tape 1 and others are blank. At first, the first tape is occupied by the input and the other tapes are kept blank. Next, the machine reads consecutive symbols under its heads and the TM prints a symbol on each tape and moves its heads.	Remember	CLO19	AIT002.19
17	Chomsky hierarchy of grammars	Type-0, Type-1, Type-2 and Type-3 grammars	Remember	CLO19	AIT002.19
18	Turing machine with two dimensional tapes	Turing machine with two dimensional tapes that have one finite control, one read-write head and one two dimensional tape. The tape has the top end and the left end but extends indefinitely to the right and down. It is divided into rows of small squares.	Remember	CLO16	AIT002.16
19	Turing machine with infinite tape	a tape has infinite length and divided into cells	Remember	CLO16	AIT002.16
20	Deterministic Turing machine	Turing machines are a model of computation. It is believed that anything that can be computed can be by a Turing Machine	Remember	CLO16	AIT002.16
21	Multi-track Turing machines	Multi-track Turing machines, a specific type of Multi-tape Turing machine, contain multiple tracks but just one tape head reads and writes on all tracks. Here, a single tape head reads n symbols from n tracks at one step. It accepts recursively enumerable languages like a normal single-track single-tape Turing Machine accepts.	Remember	CLO16	AIT002.16
22	Type 0 Grammar	Type 0 grammars generate recursively enumerable languages. The productions have no restrictions. They are any phase structure grammar including all formal grammars.	Remember	CLO19	AIT002.19
		They generate the languages that are recognized by a Turing machine.			

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
23	Type 1 Grammar	Type 1 grammars generate context-sensitive languages.  The productions must be in the form: $\alpha \rightarrow \beta$ , with the following condition $ \beta  \geq  \alpha $ , i.e. length of $\beta$ is greater than length of $\alpha$ .	Remember	CLO19	AIT002.19
24	Type 2 Grammar	<b>Type-2 grammars</b> generate context-free languages.  The productions must be in the form $A \rightarrow \gamma$  where $A \in N$ (Non terminal)  and $\gamma \in (T \cup N)^*$ (String of terminals and non-terminals).  These languages generated by these grammars are recognized by a non-deterministic pushdown automaton.	Remember	CLO19	AIT002.19
25	Type 3 Grammar	<b>Type-3 grammars</b> generate regular languages. Type-3 grammars must have a single non-terminal on the left-hand side and a right-hand side consisting of a single terminal or single terminal followed by a single non-terminal.  The productions must be in the form $X \rightarrow a$ or $X \rightarrow aY$	Remember	CLO19	AIT002.19
26	Multiple Tracks	It is a tape, which is divided into multiple tracks, on the first track the input is placed is surrounded by # and \$, using which we can perform arithmetic operations	Remember	CLO18	AIT002.18
27	Author of Turing Machine	Alan Turing OBE FRS was an English mathematician, computer scientist, logician, cryptanalyst, philosopher and theoretical  Born: 23 June 1912, Maida Vale  Died: 7 June 1954, Wilmslow, United Kingdom  Award: Smith's Prize  Education: Princeton University (1936–1938)	Remember	CLO18	AIT002.18
28	Left linear Grammar	In a <i>left-linear grammar</i> , all productions have one of the two forms:  $V \rightarrow VT^*$	Remember	CLO18	AIT002.18



S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		<p>or</p> $V \rightarrow T^*$ <p>That is, the left-hand side must consist of a single variable, and the right-hand side consists of an optional single variable followed by any number of terminals.</p>			
29	Right linear Grammar	<p>A strictly right-linear grammar is a context free grammar <math>G</math> where each rule has one of the following forms:</p> $A \rightarrow xB \text{ for } x \in \Sigma \quad A \rightarrow \epsilon$ <p>Any derivation of a word <math>w</math> from a start symbol <math>S</math> in a strictly right-linear grammar has the form</p> $S \Rightarrow x_1 A_1 \Rightarrow x_1 x_2 A_2 \Rightarrow \dots \Rightarrow x_1 \dots x_n A_n \Rightarrow x_1 \dots x_n$ <p>where <math>w = x_1 \dots x_n</math>. Note that some of the <math>x_i</math> may be <math>\epsilon</math>.</p>	Remember	CLO18	AIT002.18
30	What is the grammar accepted by Linear-bounded automaton?	Context-sensitive grammar	Remember	CLO15	AIT002.15
31	Regular Grammar	<p>Regular grammars generate regular languages. Type-3 grammars must have a single non-terminal on the left-hand side and a right-hand side consisting of a single terminal or single terminal followed by a single non-terminal.</p> <p>The productions must be in the form <math>X \rightarrow a</math> or <math>X \rightarrow aY</math></p>	Remember	CLO18	AIT002.18
32	Which operation a Turing machine cannot perform ?	insertion operation cannot perform	Remember	CLO16	AIT002.16
33	what is the Turing machine that is able to simulate other Turing machines:	A more mathematically oriented definition with the same universal nature was introduced by church and turing together called the Church-Turing thesis(formal theory of computation).	Remember	CLO17	AIT002.17
34	what is the value of n if Turing machine is defined using n-tuples	<p>The 7-tuple definition of turing machine: <math>(Q, S, G, d, q_0, B, F)</math>  where <math>Q</math>= The finite set of states of finite control  <math>S</math>= The finite set of input symbols  <math>G</math>= The complete set of tape symbols  <math>d</math>= The transition function  <math>q_0</math>= The start state, a member of <math>Q</math>, in which the finite control is found initially.  <math>B</math>= The blank symbol  <math>F</math>= The set of final or accepting states, a subset of <math>Q</math>.</p>	Remember	CLO16	AIT002.16

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
35	What is RASP ?	RASP or Random access stored program is an abstract machine that has instances like modern stored computers.	Remember	CLO16	AIT002.16
36	When two machines are said to be equivalence?	two machines be P and Q. The state in which P can simulate Q and Q can simulate P is called equivalent if P can simulate Q and Q can simulate P	Remember	CLO19	AIT002.19
37	What are games fall under the category of Turing-complete?	Many games fall under the category of Turing complete  one of them is Conway's Game of Life	Remember	CLO16	AIT002.16
38	What are the languages comes under a Non-Turing Complete language?	There exists some computational languages which are not turing complete. Regular language which is accepted by finite automata. Other examples are pixel shader languages embedded in Direct3D and OpenGL extensions.	Remember	CLO16	AIT002.16
39	What is the grammar accepted by Turing machine ?	unrestricted grammar	Remember	CLO16	AIT002.16
40	What is the grammar accepted by Linear-bounded automaton?	Context-sensitive grammar	Remember	CLO15	AIT002.15
41	How many types of grammars are there as per Chomsky hierarchy?	four types of grammars	Remember	CLO16	AIT002.16
42	Variations of Turing machines	Turing machines with two-way infinite tapes Multiple Turing machines Multihued Turing machines Nondeterministic Turing machines Turing machines with two- dimensional tapes	Remember	CLO16	AIT002.16
43	Define two-dimensional tape?	the Two –Dimensional tape is in tabular format in which the head moves one square up ,down, left or right	Remember	CLO18	AIT002.18
44	Counter Machine	there are two ways to represent counter machine  i) it is similar to the multitask Turing machine but here in place of each stack there is a counter  the counter machine is similar to restricted multitask machine	Remember	CLO18	AIT002.18
45	Turing machine with multiple heads	the Turing machine with multiple heads can have n heads ,but in any state, only one head can move	Remember	CLO18	AIT002.18
46	Non-Deterministic Turing machine	It is similar to NFA. For any state and any input symbol it can take any action from set rather than a definite predetermined action	Remember	CLO16	AIT002.16
47	Limitations of TM	It cannot decide whether two CFG are equivalent  it will not solve halting problem	Remember	CLO18	AIT002.18
48	Two way infinite tape	a tape has infinite length, the tape head can move either in forward and backward direction	Remember	CLO18	AIT002.18

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
49	Computable Languages	The Computable Languages can perform computable functions such as addition, subtraction, division, power function , square function, logarithmic functions and many more.	Remember	CLO17	AIT002.17
50	Subroutine	subroutine is nothing but a method/ function using which we can construct a TM	Remember	CLO17	AIT002.17
51	Turing machine	A Turing machine consists of a tape of infinite length on which read and writes operation can be performed. The tape consists of infinite cells on which each cell either contains input symbol ora special symbol called blank. It also consists of a head pointer which points to cell currently being read and it can move in both directions.	Remember	CLO16	AIT002.16

**Signature of the Faculty**

**Signature of the HOD**

