INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

INFORMATION TECHOLOGY

DEFINITIONS AND TERMINOLOGY

Course Name	:	**THEORY OF COMPUTATION**
Course Code	:	**AIT002**
Program	:	**B.Tech**
Semester	:	**IV**
Branch	:	**INFORMATION TECHNOLOGY**
Section	:	**A,B**
Academic Year	:	**2018– 2019**
Course Faculty	:	**Mr. Ch Suresh Kumar Raju, Assistant Professor, CSE** **Dr.K Srinivasa Reddy, Professor and HOD, IT**

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 TERMINOLOGYQUESTION BANK

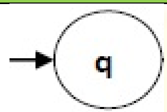
S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
UNIT - I					
1	What is proper Suffix of a String?	A group of trailing symbols other than the string is called as proper suffix. Ex: Proper Suffixes of sting abc are ε, a, bc	Remember	CLO 1	AIT002.01
2	What is power of alphabet?	Consider $\sum = \{0,1\}$ the following are the powers. 1. $\Sigma^0 = \{ \varepsilon \}$ 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
3	Define kleen closure?	The set of all strings over on alphabet. It is denoted as $\Sigma^* = \Sigma^0 U \Sigma^1 U \Sigma^2 U \Sigma^3 U\ldots$	Remember	CLO 1	AIT002.01
4	What is positive closure?	The set of all strings except ε over on alphabet. It is denoted as $\Sigma^+ = \Sigma^1 U \Sigma^2 U \Sigma^3 U\ldots$	Remember	CLO 1	AIT002.01
5	Define Language?	A language is a *set of strings*, chosen form some Σ* or A language is a subset of Σ*. A language which can be formed over 'Σ' can be Finite or Infinite. Language that contains strings over $\sum = \{a, b\}$ are $\{ \varepsilon, a, b, aa, ab, \ldots. \}$	Remember	CLO 2	AIT002.02
6	What is grammar?	It contains set of rules to generate the strings of a language.	Remember	CLO 2	AIT002.02
7	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
8	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
9	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
10	What is finite automata?	An automation the accepts regular language is called finite automata.	Remember	CLO 2	AIT002.02

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
11	Define Automata?	It is an abstract model of digital computer. It has a mechanism for reading input.	Remember	1	AIT002.01
12	Define Symbol?	Symbol is the smallest building block, which can be letter or digit or any picture. Ex: a, b, 0,1..	Remember	1	AIT002.01
13	Define Alphabet?	Alphabet is set of symbols, which are always finite. It is denoted by ∑. Ex: {0, 1}, {a, b}, {a, b, c}……	Remember	1	AIT002.01
14	What is String?	String is a finite sequence of symbols from some alphabet. String is generally denoted as w. Ex: String over ∑ = {a,b} is ababa	Remember	1	AIT002.01
15	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 ∑ = {a,b}	Remember	1	AIT002.01
16	What is substring of a string?	Any sequence of 0 or more consecutive symbols of the given string over an alphabet is called substring. Example: For string abb the substrings are ε, a, ab, abb	Remember	1	AIT002.01
17	Define Null String?	Empty string is the string with zero occurrences of symbol, represented as ε.	Remember	1	AIT002.01
18	What is Prefix of a String?	A group of leading symbols of string is called as prefix. Ex: Prefixes of sting abc are ε, a, ab, abc	Remember	1	AIT002.01
19	What is Suffix of a String?	A group of trailing symbols of string is called as suffix. Ex: Suffixes of sting abc are ε, a, bc, abc	Remember	1	AIT002.01
UNIT – II					
1	Identify the equivalent identity rule for Rε or εR	The equivalent identity rule for Rε or εR is R	Remember	CLO 5	AIT002.05
2	Identify the equivalent identity rule for ∅ L or L ∅	The equivalent identity rule for ∅ L or L ∅ is ∅	Remember	CLO 5	AIT002.05
3	Identify the equivalent identity rule for $\varepsilon+RR^*$	The equivalent identity rule for $\varepsilon+RR^*$ is R*	Remember	CLO 5	AIT002.05
4	Identify the equivalent identity rule for (PQ)*P	The equivalent identity rule for (PQ)*P is P(QP)*	Remember	CLO 5	AIT002.05

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
5	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|<=n and |v|>=1. (v) You achieve a contradiction to pumping lemma that there exists an 'i'	Remember	CLO 7	AIT002.07
6	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
7	What is the closure property of regular sets?	The regular sets are closed under union, concatenation and Kleene closure. r1Ur2= r1 +r2 r1.r2= r1r2 (r)*=r* The class of regular sets are closed under complementation, substitution, homomorphism and inverse homomorphism.	Remember	CLO 5	AIT002.05
8	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
9	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
10	Identify the RE for string length atleast 2	The regular expression for strings of length exactly 2 is (a+b)(a+b)(a+b)	Understand	CLO 4	AIT002.04
11	What is a regular expression?	A regular expression is a string that describes the whole set of strings according to certain syntax rules.	Remember	5	AIT002.05
12	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	5	AIT002.05
13	What is a regular language?	A language accepted by the regular expression is called as Regular Language.	Remember	5	AIT002.05
14	What is a regular set?	A regular language which is accepted by finite automata is called regular set.	Remember	5	AIT002.05
15	What is Kleen closure?	Kleen closure is defined as * that is 0 or more occurrence.	Remember	5	AIT002.05
16	What is positive closure ?	positive closure is defined as + that is 1 or more occurrence.	Remember	5	AIT002.05
17	Identify the equivalent identity rule for RR^*	The equivalent identity rule for RR^* is R*R	Remember	5	AIT002.05
18	Identify the equivalent identity rule for (R*)*	The equivalent identity rule for (R*)* is R*	Remember	5	AIT002.05

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
19	Identify the equivalent identity rule for R*R*	The equivalent identity rule for R*R*is R*	Remember	5	AIT002.05
20	Identify the equivalent identity rule for R+R	The equivalent identity rule for R+R is R(Idempotent law)	Remember	5	AIT002.05
UNIT – III					
1	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
2	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
3	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
4	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
5	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| >=n, then z=uvwxy such that : (i) |vx| >=1 (ii) |vwx| <=n and (iii) for all i>=0 uviwxiy is in L.	Understand	CLO 12	AIT002.12
6	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
7	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
8	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	CLO9	AIT002.09
9	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	Remember	CLO9	AIT002.09

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		e)For construction of compilers			
10	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	CLO10	AIT002.10
11	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	CL09	AIT002.09
12	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	CLO9	AIT002.09
13	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	CLO9	AIT002.09
14	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
15	What is CFL?	L is a context free language (CFL) if it is L(G) for some CFG G..	Remember	CLO 9	AIT002.09
16	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
17	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
18	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
19	If S->aSb | aAb , A->bAa , A->ba .Find out the CFL	S->aAb=>abab S->aSb=>a aAb b =>a a ba b b(sub S->aAb) S->aSb =>a aSb b =>a a aAb b b=>a a a ba b bb Thus L={anbmambn, where n,m>=1}	Analyze	CLO 10	AIT002.09

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
20	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
		UNIT - IV			
1	Show the notation of the input state in PDA ?		Remember	CLO 14	AIT002.14
2	What is meant by empty stack acceptability	A PDA accepts a string when, after reading the entire string, the PDA has emptied its stack. For a PDA (Q, ∑, S, δ, q_0, I, F), the language accepted by the empty stack is − L(PDA) = {w | (q_0, w, I) $\vdash^*$ (q, ε, ε), q ∈ Q}	Remember	CLO 12	AIT002.12
3	Show the notation of the output state in PDA ?		Remember	CLO 14	AIT002.14
4	Define acceptance of PDA by Final State	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. For a PDA (Q, ∑, S, δ, q_0, I, F), the language accepted by the set of final states F is − L(PDA) = {w | (q_0, w, I) ⊢* (q, ε, x), q ∈ F} for any input stack string **x**.	Remember	CLO 14	AIT002.14
5	Define acceptance of PDA by empty Stack?	In final state acceptability, a PDA accepts a string when, after reading the entire string, the PDA is in a final state. For a PDA (Q, ∑, S, δ, q_0, I, F), the language accepted by the empty stack is −	Remember	CLO 15	AIT002.15

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		L(PDA) = {w | (q_0, w, I) ⊢* (q, ε, ε), q ∈ Q}			
6	Show the notation of the transition symbol between two states in PDA ?	$\longrightarrow$	Remember	CLO 14	AIT002.14
7	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
8	***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
9	***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
10	***Define NPDA***	A nondeterministic pushdown automaton (npda) is basically an nfa with a stack added to it.	Remember	CLO 14	AIT002.14
11	***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
12	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 − "Finite state machine" + "a stack" A pushdown automaton has three components − • an input tape, • a control unit, and • a stack with infinite size.	Remember	CLO 14	AIT002.14
13	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
14	How to represent the PDA?	PDA M = (Q, ,Γ,∑, δ,q_0,Z_0,F)	Remember	CLO 14	AIT002.14
15	How to represent transition function in PDA?	δ : Q x ∑ x Γ=> Q x Γ	Remember	CLO 14	AIT002.14

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
16	Name the 7 tuples of PDA	Q= Set Of All States Γ = stack symbols $\sum$ = input alphabet δ = transition function q_0 =start state Z_0 = Initial stack top symbol F = Final/accepting states	Remember	CLO 14	AIT002.14
17	How many operations are performed on the stack ?	A stack does two operations • Push − a new symbol is added at the top. • Pop − the top symbol is read and removed.	Remember	CLO 14	AIT002.14
18	How to represent the empty stack initially?	Initially we put a special symbol **'$'** into the empty stack.	Remember	CLO 14	AIT002.14
19	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 14	AIT002.14
20	Show the notation of the empty string in PDA	If $\|S\|= 0$, it is called an empty string (Denoted by λ or ε)	Remember	CLO 14	AIT002.14
UNIT - V					
1	Context-sensitive grammar	Context sensitive grammars generates context-sensitive languages. The productions must be in the form: $\alpha \rightarrow \beta$, with the following condition $\|\beta\| >=\|\alpha\|$,i.e. length of β is greater than length of α.	Remember	CLO19	AIT002.19
2	Unrestricted grammar	Type-0 grammars generate recursively enumerable languages. The productions have no restrictions. They are any phase structure grammar including all formal grammars. They generate the languages that are recognized by a Turing machine.	Remember	CLO19	AIT002.19

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
3	Pushdown 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 – "Finite state machine" + "a stack" A pushdown automaton has three components − • an input tape, • a control unit, and • a stack with infinite size. The stack head scans the top symbol of the stack.	Remember	CLO19	AIT002.19
4	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
5	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
6	Chomsky hierarchy of grammars	Type-0, Type-1, Type-2 and Type-3 grammars	Remember	CLO19	AIT002.19
7	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	CLO19	AIT002.19
8	***Turing machine with infinite tape***	a tape has infinite length and divided into cells	Remember	CLO19	AIT002.19
9	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	CLO19	AIT002.19
10	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	Remember	CLO19	AIT002.19

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		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.			
11	Define 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
12	Define 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	CLO16	AIT002.16
13	Define recursive languages?	A language is *recursive* if there exists a Turing machine that accepts every string of the language and rejects every string that is not in the language.	Remember	CLO16	AIT002.16
14	Define 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	CLO16	AIT002.16
15	Define 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	CLO16	AIT002.16
16	Define 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
17	Define context sensitive language?	Type-1 grammars generate context-sensitive languages.	Remember	CLO16	AIT002.16
18	Define Chomsky hierarchy of languages?	One way showing the relationship among the languages if Chomsky hierarchy. Type 0 known as unrestricted grammar.	Remember	CLO16	AIT002.16

S No	QUESTION	ANSWER	Blooms Level	CLO	CLO Code
		Type 1 known as context sensitive grammar. Type 2 known as context free grammar. Type 3 Regular Grammar.			
19	Define Regular Grammar?	Type-3 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. The productions must be in the form $X \to a$ or $X \to aY$ where $X, Y \in N$ (Non terminal) and $a \in T$ (Terminal) The rule $S \to \varepsilon$ is allowed if S does not appear on the right side of any rule.	Remember	CLO16	AIT002.16
20	Define Type-2 grammars ?	Type-2 grammars generate context-free languages. The productions must be in the form $A \to \gamma$ 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 be recognized by a non-deterministic pushdown automaton.	Remember	CLO16	AIT002.16

Signature of the Faculty **Signature of the HOD**