| LOGO.jpg | **GEETHANJALI INSTITUTE OF SCIENCE & TECHNOLOGY**  (**AN AUTONOMOUS INSTITUTION**)  **(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)**  **(Accredited by NAAC with “A” Grade, NBA (EEE, ECE & ME) & ISO 9001:2008 Certified Institution)** |
| --- | --- |
| **QUESTIONBANK(DESCRIPTIVE)**  **Subject Name with Code:** Automata Theory & Compiler Design (23A0521T)  **Course & Branch: Year & Semester**: III-I **Regulation:** RG 23 | |

**UNIT - I**

| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| --- | --- | --- |
| **2 Marks Questions (Short)** | | |
|  | Define alphabet, string and language. | L1, CO1,2M |
|  | Distinguish between DFA and NFA | L4, CO1,2M |
|  | Define non-deterministic finite automata | L1, CO1,2M |
|  | Draw DFA that accepts strings which has a substring of 101 over an alphabet {0, 1} | L3, CO1,2M |
|  | Construct a DFA that accepts binary strings that are divisible by 3. | L3, CO1,2M |
|  | Define ε-closure | L1, CO1,2M |
|  | List Types of Grammars | L2, CO1,2M |
|  | List any four closure properties of regular languages | L1, CO1,2M |
|  | Write down the pumping lemma of regular languages. | L2, CO1,2M |
| Descriptive Questions (Long) | | |
|  | Illustrate minimization of FA using Table filling method | L3,CO1,10M |
|  | Explain DFA and NFA with differences. | L4,CO1,10M |
|  | Give the formal definition of DFA and design a DFA to accept all decimal numbers divisible by 3 on Σ = {0, 1, … . . ,9}. Show the moves of DFA for strings 369 and 964 | L3,CO1,10M |
|  | Illustrate construction of DFA to accept binary string whose decimal equivalent is divisible by 5 and ∑ = {0, 1}\*.Show the moves of DFA for 25 and 18 | L3,CO1,10M |
|  | Find the equivalent DFA for the following NFA | L3,CO1,10M |
|  | Find DFA equivalent to the following NFA - ∈ | L3,CO1,10M |
|  | State and prove Pumping Lemma for regular language and show  L = {anb2n| n > 0} is not regular language | L3,CO1,10M |
|  | Check whether the language L = {ap /where p is prime} is regular or not | L4,CO1,10M |
|  | Define Regular Expression. List and explain the closure properties of regular expression. | L2,CO1,10M |

**UNIT - II**

| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| --- | --- | --- |
| **2 Marks Questions (Short)** | | |
|  | What is ambiguity in CFG? Give an example | L1, CO2,2M |
|  | Define context free grammar. Give an example. | L1, CO2,2M |
|  | What is left factoring in CFG? Give an example | L1, CO2,2M |
|  | What is left recursion in CFG? Give an example | L1, CO2,2M |
|  | What is useless symbol in grammar? Give an example? | L2, CO2,2M |
|  | Consider the following context free grammar and construct the parse tree for the string 10011001: S → 0/1/0S0/1S1/∈. | L1, CO2,2M |
|  | Show the leftmost derivation and the corresponding parse tree for the string a+a\*a using the CFG: E → E+E / E\*E / a | L2, CO2,2M |
|  | What is instantaneous description of PDA? | L1, CO2,2M |
|  | Graphical Notation of PDA. | L1, CO2,2M |
| Descriptive Questions (Long) | | |
|  | Write about Chomsky hierarchy of languages | L2,CO2,10M |
|  | Define parse tree. Write about leftmost derivation and rightmost derivation with example | L2,CO2,10M |
|  | Explain Left recursion and left factoring with example | L2,CO2,10M |
|  | What is ambiguous CFG? Check whether the following grammar is ambiguous or not. 𝑆 → 𝐴/𝐵 𝐴 → 0𝐴/ ε 𝐵 → 1𝐵/0𝐵/ ε | L4,CO3,10M |
|  | Convert the given Grammar into Chomsky Normal form  S → aA | aBB  A → aAA | ε  B → bB | bbc  C → B | L3,CO4,10M |
|  | Convert the given grammar into GNF  S → AA | a  A→ SS | b | L3,CO4,10M |
|  | Construct Deterministic PDA to accept the following language:  L = {an C bn | n>0} Also write ID for the string = aacbb. | L3,CO4,10M |
|  | Design PDA for the language L={WWR | W∈ (0,1)+} by empty stack? | L3,CO4,10M |
|  | Construct PDA for the following CFG. Also write ID’s for  string = 00111  given CFG is:  S → A111 | S1  A → A0 | 00. | L3,CO4,10M |

**UNIT - III**

| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| --- | --- | --- |
| **2 Marks Questions (Short)** | | |
|  | Explain Turing machine | L1, CO3,2M |
|  | Graphical Notation of Turing Machine. | L1, CO3,2M |
|  | What is instantaneous description of TM? | L1, CO3,2M |
|  | List the Phases of Compiler? | L2, CO4,2M |
|  | What is Compiler? | L1, CO4,2M |
|  | What is meant by sentinels? | L1, CO4,2M |
|  | Explain Interpreter | L2, CO4,2M |
|  | Explain Role of Lexical analyzer | L2, CO4,2M |
|  | Differentiate between compiler and interpreter. | L4, CO4,2M |
| Descriptive Questions (Long) | | |
|  | Design Turing machine for the language L={WWR | W∈ (a,b)+} | L3,CO3,10M |
|  | Design Turing machine for 1’s and 2’s Complement of Binary number | L3,CO3,10M |
|  | Design Turing machine for the language L={an bn cn | n>=1} | L3,CO3,10M |
|  | Explain Turing Machine with example | L2,CO3,10M |
|  | Design Turing machine for the language L={an bn | n>=1} | L3,CO3,10M |
|  | Define Compiler and Interpreter? Differentiate between compiler and interpreter. | L4,CO4,10M |
|  | Explain in detail about the phases of a compiler. | L2,CO4,10M |
|  | Explain Role of Lexical analyzer | L2,CO4,10M |
|  | Explain about input buffering. | L2,CO4,10M |

**UNIT - IV**

| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| --- | --- | --- |
| **2 Marks Questions (Short)** | | |
|  | Give the role of parser. | L1, CO5,2M |
|  | Write a short note on LR parsing. | L1, CO5,2M |
|  | List different types of Three Address Code? | L1, CO5,2M |
|  | Differentiate bottom up parsing and top down parsing | L4, CO5,2M |
|  | What is handle pruning? | L2, CO5,2M |
|  | Explain LL(1) grammar | L2, CO5,2M |
|  | Explain Topdown parsing | L2, CO5,2M |
|  | Explain Bottomup parsing | L2, CO5,2M |
|  | List any 2 instruction forms of Three Address Code | L1, CO5,2M |
| Descriptive Questions (Long) | | |
|  | Explain the recursive predictive parsing with diagram | L2,CO5,10M |
|  | Construct LALR parsing for the following grammar: S→ CC C→ cC/d | L3,CO5,10M |
|  | Construct SLR parsing table for the following grammar:  E → E + T/T  T → T \* F/F  F → (E)/a  Show the moves of the parser for parsing the string a \* a + a. | L3,CO5,10M |
|  | Explain the non-recursive predictive parsing with diagram | L2,CO5,10M |
|  | Check the following grammar is LL(1) or not?  E → E + T/T  T → T \* F/F  F → (E)/a | L3,CO5,10M |
|  | Differentiate between Top down parsing and Bottom-up parsing. | L4,CO5,10M |
|  | Explain LR Parser with diagram | L2,CO5,10M |
|  | Explain Instruction forms of Three Address Code | L2,CO5,10M |
|  | Explain types of Three Address Code | L2,CO5,10M |

**UNIT - V**

| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| --- | --- | --- |
| **2 Marks Questions (Short)** | | |
|  | Explain Dead code elimination | L2, CO6,2M |
|  | Explain Copy propagation | L2, CO6,2M |
|  | Explain Constant propagation | L2, CO6,2M |
|  | Explain Common sub expression elimination | L2, CO6,2M |
|  | List various issues in code generator | L1, CO6,2M |
|  | Explain DAG | L2, CO6,2M |
|  | Explain Strength Reduction | L2, CO6,2M |
|  | Explain machine Idioms | L2, CO6,2M |
|  | Explain Code motion | L2, CO6,2M |
| Descriptive Questions (Long) | | |
|  | Explain Peephole Optimization | L2,CO6,10M |
|  | Explain about various issues in code generator | L2,CO6,10M |
|  | Explain Function preserving transformations | L2,CO6,10M |
|  | Explain Simple Code generator | L2,CO6,10M |
|  | Explain Loop Optimization | L2,CO6,10M |
|  | Explain basic blocks with example | L2,CO6,10M |
|  | Explain flow graph with example | L2,CO6,10M |
|  | Explain DAG with example | L2,CO6,10M |
|  | Explain 1. Copy Propagation 2.Strength Reduction 3.Machine Idioms | L2,CO6,10M |

**Signature of the Staff:**

**Signature of Department Academic Committee Member 1:**

**Signature of Department Academic Committee Member 2:**

**Signature of Department Academic Committee Member 3:**