|  |  |
| --- | --- |
| 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 (22A05601)  **Course & Branch: Year & Semester**: III-I **Regulation:** RG 22 | |

**UNIT - I**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| **2 Marks Questions (Short)** | | |
|  | Differentiate between a Moore machine and a mealy machine | L1, CO1,2M |
|  | Define alphabet, string and language. | L1, CO1,2M |
|  | Distinguish between DFA and NFA | L1, CO1,2M |
|  | Define non-deterministic finite automata | L1, CO1,2M |
|  | What are the applications of automata theory? | L2, CO1,2M |
|  | Draw DFA that accepts strings which has a substring of 101 over an alphabet {0, 1} | L2, CO1,2M |
|  | Define a DFA formally | L2, CO1,2M |
|  | Define ε-closure | L2, CO1,2M |
|  | Construct a DFA that accepts binary strings that are divisible by 3. | L2, CO1,2M |
|  | List methods for minimizing finite automata | L2, CO1,2M |
| Descriptive Questions (Long) | | |
|  | Illustrate minimization of FA using Table filling method | L2,CO1,10M |
|  | Convert the following Melay machine to Moore machine | L2,CO1,10M |
|  | |  |  | | --- | --- | | |  | | --- | | convert given Moore machine into equivalent Mealy machine  moore to melay1.JPG | | | L2,CO1,10M |
|  | Explain DFA and NFA with differences. | L3,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}\* | L2,CO1,10M |
|  | Find the equivalent DFA for the following NFA | L2,CO1,10M |
|  | Find DFA equivalent to the following NFA - ∈ | L2,CO1,10M |
|  | Convert the following NFA to DFA  DFA from NFA.JPG | L3,CO1,10M |
|  | Explain melay and moore machines with differences. | L2,CO1,10M |

**UNIT - II**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| **2 Marks Questions (Short)** | | |
|  | Define regular expression. | L1, CO2,2M |
|  | Give the regular expression for the language all string over alphabet {0,1} containing at least two consecutive 0’s. | L1, CO2,2M |
|  | Write down the pumping lemma of regular sets. | L1, CO2,2M |
|  | List any four closure properties of regular languages | L2, CO2,2M |
|  | State Arden’s theorem. | L2, CO2,2M |
|  | Differentiate between left and right linear grammars | L1, CO2,2M |
|  | Write a regular expression over the alphabet {0, 1} which starts with 0 and ends with 1. | L2, CO2,2M |
|  | Define linear grammar. | L2, CO2,2M |
|  | What are the applications of the regular expression? | L2, CO2,2M |
|  | List Types of Grammars | L2, CO2,2M |
| Descriptive Questions (Long) | | |
|  | Find the regular expression corresponding to the following DFA. | L3,CO2,10M |
|  | Define regular expression. List and explain the closure properties of regular expression. | L2,CO2,10M |
|  | Convert given finite automata to regular expression using Arden’s theorem | L3,CO2,10M |
|  | State and prove Arden’s theorem | L3,CO2,10M |
|  | Convert the following DFA to a regular expression by state elimination technique. | L2,CO2,10M |
|  | Explain how equivalence between two FA is verified with an example | L3,CO2,10M |
|  | Convert the following DFA to Regular grammar: | L2,CO2,10M |
|  | List and explain the algebraic properties of regular expressions. | L2,CO2,10M |
|  | State and prove Pumping Lemma for regular language and show  L = {anb2n| n > 0} is not regular language | L2,CO2,10M |
|  | Check whether the language L = {ap /where p is prime} is regular or not | L2,CO2,10M |

**UNIT - III**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| **2 Marks Questions (Short)** | | |
|  | Explain left most and rightmost derivation for a string | L1, CO3,2M |
|  | What is ambiguity in CFG? Give an example | L1, CO3,2M |
|  | Define context free grammar. Give an example. | L1, CO3,2M |
|  | What is left factoring in CFG? Give an example | L2, CO3,2M |
|  | What are the applications of context free languages? | L2, CO3,2M |
|  | What is left recursion in CFG? Give an example | L2, CO3,2M |
|  | What is useless symbol in a grammar give an example? | L1, CO3,2M |
|  | With an example, define parse tree of a grammar | L2, CO3,2M |
|  | Consider the following context free grammar and construct the parse tree for the string 10011001: S → 0/1/0S0/1S1/∈. | L1, CO3,2M |
|  | 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, CO3,2M |
| Descriptive Questions (Long) | | |
|  | Write about Chomsky hierarchy of languages | L2,CO3,10M |
|  | Define parse tree. Write about leftmost derivation and rightmost derivation with example | L3,CO3,10M |
|  | Convert the grammar G to Chomsky Normal form: G = ({S,A,B}, {a,b}, P,S) where P consist of the productions: S->bA | aB A->bAA|aS|a B->b|bS|aBB | L2,CO3,10M |
|  | State and prove Pumping Lemma for context free language and show L= {an bn cn |n>=0} is not a context free language | L2,CO3,10M |
|  | Define context free grammar. List the closure properties of context free languages. | L3,CO3,10M |
|  | Explain Left recursion and left factoring with example | L2,CO3,10M |
|  | eliminate all 𝜀 - productions from the following grammar: S → ABCa|bD, A → BC|b, B → b|ε, C → c|ε, D → d | L2,CO3,10M |
|  | Eliminate useless production from the grammar given below. 𝑆 → 𝑎𝑆|𝐴|𝐶 𝐴 → 𝑎 𝐵 → 𝑎𝑎 𝐶 → 𝑎𝐶b. | L2,CO3,10M |
|  | Illustrate construction of parse tree, LMD, RMD for the given string w = aaabbabb CFG is as follows: S -> a S b | S1 S1 -> a S1 a| b S1 b| ε | L2,CO3,10M |
|  | What is ambiguous CFG? Check whether the following grammar is ambiguous or not. 𝑆 → 𝐴/𝐵 𝐴 → 0𝐴/𝜖 𝐵 → 1𝐵/0𝐵/ ε | L2,CO3,10M |

**UNIT - IV**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Question** | **[BT Level] [CO][ Marks]** |
| **2 Marks Questions (Short)** | | |
|  | What is Compiler? | L1, CO4,2M |
|  | Define phase and pass. | L1, CO4,2M |
|  | List phases of Compiler | L1, CO4,2M |
|  | What is a cross compiler? | L1, CO4,2M |
|  | Differentiate between compiler and interpreter. | L2, CO4,2M |
|  | Explain about sentinels | L2, CO4,2M |
|  | Explain role of input buffering | L2, CO4,2M |
|  | Explain Lex tool | L2, CO4,2M |
|  | Explain Lexeme, Token and Pattern | L2, CO4,2M |
|  | Explain Role of Lexical analyzer | L2, CO4,2M |
| Descriptive Questions (Long) | | |
|  | Explain in detail about the phases of a compiler. | L2,CO4,10M |
|  | Explain about input buffering. | L2,CO4,10M |
|  | Explain Lexical errors in detail. | L2,CO4,10M |
|  | What are a transition diagrams? Show the transition diagram for relational operators and numbers | L3,CO4,10M |
|  | Discuss in brief about LEX. | L3,CO4,10M |
|  | Explain about Design of Lexical Analyzer Generator | L2,CO4,10M |
|  | Explain Role of lexical analyzer | L2,CO4,10M |
|  | Explain Recognition of tokens. | L2,CO4,10M |
|  | What is the role of regular expression in lexical analysis? Explain with examples | L4,CO4,10M |
|  | Define Compiler and Interpreter? Differentiate between compiler and interpreter. | L2,CO4,10M |

**UNIT - V**

|  |  |  |
| --- | --- | --- |
| **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 |
|  | Differentiate between S-attribute SDDs and L-attribute SDDs. | L2, CO6,2M |
|  | List different types of Three Address Code? | L1, CO6,2M |
|  | What is handle pruning? | L2, CO5,2M |
|  | Write the different rules for computing the First (). | L1, CO5,2M |
|  | What are various parser conflicts in LR Parsers? | L2, CO5,2M |
|  | Differentiate bottom up parsing and top down parsing | L2, CO5,2M |
|  | Explain Dead code elimination | L2, CO6,2M |
|  | What is YACC stands for? What is its role? | L2, CO5,2M |
| Descriptive Questions (Long) | | |
|  | Explain the recursive predictive parsing with diagram | L3,CO5,10M |
|  | Explain about various issues in code generator | L3,CO6,10M |
|  | 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 |
|  | Write inherited attributes and synthesized attributes | L2,CO6,10M |
|  | Explain Function preserving transformations | L2,CO6,10M |
|  | Differentiate between Top down parsing and Bottom-up parsing. | L2,CO5,10M |
|  | Explain LR Parser with diagram | L2,CO5,10M |

**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:**