**Course Code:** 20CS0903 **R20**

# SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:PUTTUR
## (AUTONOMOUS)
**Siddharth Nagar, Narayanavanam Road – 517583**

**QUESTION BANK (DESCRIPTIVE)**

**Subject with Code: AT & CD**(20CS0903) **Course & Branch**: B.Tech – CSM,CIC

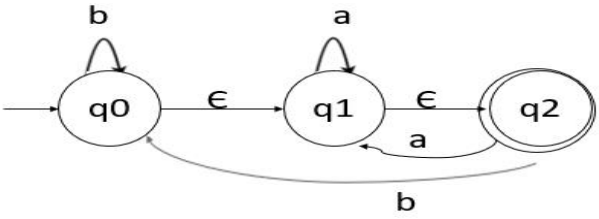
**Year &Sem:** III-B.Tech & I-Sem **Regulation:** R20

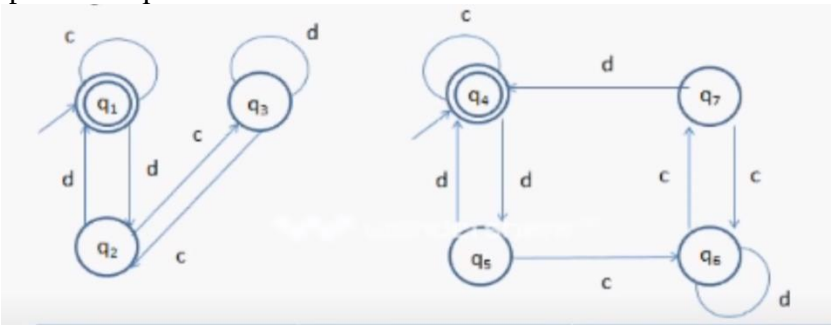
## UNIT –I
## FINITE AUTOMATA AND REGULAR LANGUAGES

| | | | | |
|---|---|---|---|---|
| 1 | a | Consider the below finite automata and check whether the strings are accepted or not<br>(i) 0001 (ii) 1010 (iii) 1001 (iv)0101 | [L1][CO1] | [8M] |
| | b | Define alphabets, strings, Languages? | [L3][CO1] | [4M] |
| 2 | a | Compare DFA and NFA | [L2][CO1] | [4M] |
| | b | Construct DFA for the given NFA | [L6][CO2] | [8M] |
| 3 | a | Write the process of conversion from NFA with ε moves to DFA. | [L4][CO3] | [4M] |
| | b | Convert the following NFA with ε moves to DFA. | [L6][CO2] | [8M] |
| 4 | a | Write the process of equivalence two FA's? | [L4][CO3] | [4M] |
| | b | Compare the equivalence two FA's or not. | [L4][CO3] | [8M] |

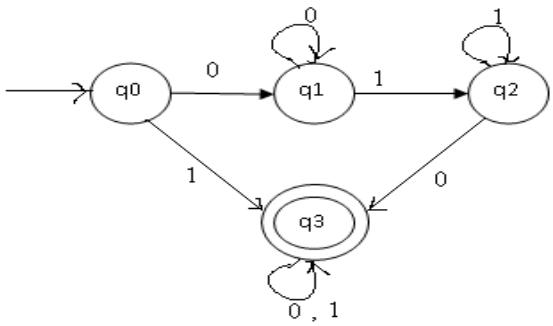
| States (Q) | Input Alphabtes 0 | 1 |
|---|---|---|
| →q0 | q1 | q3 |
| q1 | q0 | q2 |
| (q2) | q3 | q1 |
| q3 | q2 | q0 |

| | Next state 0 | 1 |
|---|---|---|
| → q0 | q0,q1 | q0 |
| q1 | q2 | q1 |
| q2 | q3 | q3 |
| (q3) | - | q2 |



**Course Code:** 20CS0903 **R20**

<table>
<tr><td>5</td><td>a</td><td>Contrast Mealy machine and Moore machine.</td><td>[L4][CO1]</td><td>[6M]</td></tr>
<tr><td></td><td>b</td><td>Convert the following Mealy machine into its equivalent Moore machine.
<table>
<tr><th rowspan="2">Present State</th><th colspan="2">I/P=0</th><th colspan="2">I/P=1</th></tr>
<tr><th>Next State</th><th>O/P</th><th>Next State</th><th>O/P</th></tr>
<tr><td>→A</td><td>C</td><td>0</td><td>B</td><td>0</td></tr>
<tr><td>B</td><td>A</td><td>1</td><td>D</td><td>0</td></tr>
<tr><td>C</td><td>B</td><td>1</td><td>A</td><td>1</td></tr>
<tr><td>D</td><td>D</td><td>1</td><td>C</td><td>0</td></tr>
</table></td><td>[L3][CO2]</td><td>[6M]</td></tr>
<tr><td>6</td><td>a</td><td>Define Melay machine and Moore machine.</td><td>[L3][CO1]</td><td>[6M]</td></tr>
<tr><td></td><td>c</td><td>Construct Mealy machine corresponding to Moore machine?
<table>
<tr><th rowspan="2">States (Q)</th><th colspan="2">Next States</th><th rowspan="2">Output</th></tr>
<tr><th>I/P=0</th><th>I/P=1</th></tr>
<tr><td>→q1</td><td>q1</td><td>q2</td><td>0</td></tr>
<tr><td>q2</td><td>q1</td><td>q3</td><td>0</td></tr>
<tr><td>q3</td><td>q1</td><td>q3</td><td>1</td></tr>
</table></td><td>[L3][CO2]</td><td>[6M]</td></tr>
<tr><td>7</td><td>a</td><td>List out the identities of Regular expression.</td><td>[L1][CO3]</td><td>[6M]</td></tr>
<tr><td></td><td>b</td><td>From the identities of RE, prove that<br>i) 10+(1010)*[^+(1010)*]=10+(1010)*<br>ii) (1+100*)+(1+100*)(0+10*)(0+10*)*=10*(0+10*)*</td><td>[L3][CO3]</td><td>[6M]</td></tr>
<tr><td>8</td><td>a</td><td>Prove R=Q+RP has unique solution, R=QP*</td><td>[L3][CO3]</td><td>[4M]</td></tr>
<tr><td></td><td>b</td><td>Construct RE from given FA by using Arden’s Theorem.

</td><td>[L6][CO3]</td><td>[8M]</td></tr>
<tr><td>9</td><td>a</td><td>State Pumping lemma for regular languages.</td><td>[L1][CO3]</td><td>[4M]</td></tr>
<tr><td></td><td>b</td><td>Prove that L = $\{a^i b^i \mid i \geq 0\}$ is not regular</td><td>[L3][CO3]</td><td>[8M]</td></tr>
<tr><td>10</td><td>a</td><td>Give the Closure properties of Regular Sets</td><td>[L1][CO2]</td><td>[6M]</td></tr>
<tr><td></td><td>b</td><td>What are the applications of Pumping Lemma?</td><td>[L1][CO3]</td><td>[6M]</td></tr>
</table>

**Course Code:** 20CS0903 **R20**

## UNIT –II
## CONTEXT FREE GRAMMAR AND TURING MACHINE

| | | | | |
|---|---|---|---|---|
| 1 | a | Analyze and explain with example Chomsky Hierarchy of Languages | [L4][CO1] | [6M] |
| | b | Define the following terms:<br>i) Useless symbol<br>ii) Null production<br>iii) Unit productions | [L1][CO4] | [6M] |
| 2 | a | Describe what is meant by Simplifying the Grammar. | [L2][CO4] | [4M] |
| | b | Evaluate simplification of the following context free grammar.<br>S→ Aa /B<br>B→ a/bc<br>C→ a / ϵ | [L5][CO4] | [8M] |
| 3 | | Interpret simplification of the given grammar. Simplify the following CFG<br>S→ aSb S→A A→cAd A→cd | [L5][CO4] | [12M] |
| 4 | a | Remove the unit production from the grammar<br>S→AB A→E B→C C→D D→b E→a | [L3][CO4] | [6M] |
| | b | Remove ϵ productons from the grammar<br>S→ABaC A→BC B→b/ ϵ C→D/ϵ D→d | [L3][CO4] | [6M] |
| 5 | a | Write the process adapted to convert the grammar into CNF? | [L2][CO4] | [4M] |
| | b | Convert the following grammar into CNF.<br>S→ bA/aB<br>A→bAA/aS/a<br>B→aBB/bS/a. | [L3][CO4] | [8M] |
| 6 | a | State Pumping lemma for Context-free language | [L1][CO4] | [4M] |
| | b | Show that L = $\{a^n b^n c^n$ , where n>=1$\}$ is not context free. | [L3][CO4] | [8M] |
| 7 | a | State Turing machine. | [L1][CO6] | [4M] |
| | b | Construct a TM for regular Expression 01(00+11)(0+1)*1. | [L6][CO6] | [8M] |
| 8 | | Explain the various types of Turing machine. | [L2][CO6] | [12M] |
| 9 | a | Differentiate PCP and MPCP. | [L4][CO6] | [4M] |
| | b | Find the PCP solution for the following sets.<br>A: 10, 01, 0, 100, 1<br>B: 101, 100, 10, 0, 010 | [L5][CO6] | [8M] |
| 10 | a | State the formal of PDA. | [L1][CO5] | [4M] |
| | b | Construct an equivalent PDA for the following CFG.<br>S→aAB \| bBA<br>A→bS\|a<br>B→aS \| b. | [L6][CO5] | [8M] |

**Course Code:** 20CS0903 **R20**

## UNIT –III
## LEXICAL ANALYSIS AND TOP DOWN PARSING

| | | | | |
|---|---|---|---|---|
| 1 | | Explain the phases of a compiler with neat diagram. | [L2][CO2] | [12M] |
| 2 | a | Explain in detail about the role of lexical analyzer in Compiler Design. | [L2][CO1] | [6M] |
| | b | Write about input buffering? | [L3][CO1] | [6M] |
| 3 | a | Explain LEX Tool with the structure of Lex Program? | [L2][CO3] | [8M] |
| | b | Illustrate Application of compiler technology | [L3][CO1] | [4M] |
| 4 | a | State what is meant by derivation and parse tree with examples. | [L1][CO4] | [4M] |
| | b | Construct Leftmost and Rightmost derivation and derivation tree for the string 0100110<br>S→0S/1AA<br>A→0/1A/0B<br>B→1/0BB | [L6][CO4] | [8M] |
| 5 | a | Describe the procedure of eliminating Left recursion. | [L1][CO1] | [6M] |
| | b | Eliminate left recursion for the following grammar<br>E→E+T/T<br>T→T*F/F<br>F→(E)/id | [L5][CO1] | [6M] |
| 6 | a | Explain Left recursion and Left factoring. | [L2][CO1] | [6M] |
| | b | Perform left factor for the grammar A→abB/aB/cdg/cdeB/cdfB | [L3][CO4] | [6M] |
| 7 | a | Describe the role of Compiler | [L1][CO1] | [4M] |
| | b | Design the recursive decent parser for the following grammar?<br>E→E+T/T<br>T→T*F/F<br>F→(E)/id | [L6][CO3] | [8M] |
| 8 | a | Illustrate the rules to be followed in finding the FIRST and FOLLOW. | [L3][CO1] | [4M] |
| | b | Find FIRST and FOLLOW for the following grammar? E→ E+T/T T→T*F/F F→(E)/id | [L3][CO2] | [8M] |
| 9 | | Consider the grammar E→E+T/T, T→T*F/F, F→(E)\|id Design predictive parsing table and check given grammar is LL(1) Grammar or not? | [L6][CO3] | [12M] |
| 10 | | Consider the grammar E→TE$^1$<br>E$^1$→+TE$^1$\| -TE$^1$ \| ε<br>T→FT$^1$<br>T$^1$→*FT$^1$ \| / FT$^1$ \| ε<br>F→GG$^1$<br>G$^1$→^F/ ε<br>G→(E)/ id<br>Calculate FIRST and FOLLOW for the above grammar | [L4][CO2] | [12M] |

**Course Code:** 20CS0903 **R20**

## UNIT –IV
## BOTTOM-UP PARSING AND SEMANTIC ANANLYSIS

| | | | | |
|---|---|---|---|---|
| 1 | a | Explain about handle pruning | [L2][CO1] | [6M] |
| | b | Summarize about SLR parsing | [L2][CO1] | [6M] |
| 2 | a | Describe bottom up parsing | [L1][CO2] | [4M] |
| | b | Differences between SLR,CLR, LALR parsers | [L4][CO2] | [8M] |
| 3 | | Prepare Shift Reduce Parsing for the input string using the grammar<br>S→(L)\|a<br>L→L,S\|S<br>a)(a,(a,a))<br>b)(a,a) | [L6][CO3] | [12M] |
| 4 | a | Define augmented grammar. | [L1][CO2] | [2M] |
| | b | Construct the LR(0) items for the following Grammar<br>S→L=R<br>S→R<br>L→*R<br>L→id<br>R→L | [L6][CO3] | [10M] |
| 5 | | Construct CLR Parsing table for the given grammar<br>S→CC<br>C→aC/d | [L6][CO3] | [12M] |
| 6 | | Design the LALR parser for the following Grammar<br>S → AA<br>A → aA<br>A →b | [L6][CO3] | [12M] |
| 7 | a | Define YACC parser in Syntax Analysis. | [L1][CO3] | [2M] |
| | b | Explain in detail about YACC Parser generator tool. | [L2][CO3] | [10M] |
| 8 | a | Explain syntax directed definition with simple examples | [L2][CO2] | [6M] |
| | b | Describe in detail the Translation scheme of SDD. | [L2][CO2] | [6M] |
| 9 | a | Define a syntax-directed translation and explain with example. | [L2][CO2] | [6M] |
| | b | Give the evaluation order of SDT with an example. | [L5][CO2] | [6M] |
| 10 | | Discuss Type Checking with suitable examples. | [L2][CO4] | [12M] |

**Course Code:** 20CS0903 **R20**

## UNIT –V
## CODE OPTIMIZATION AND CODE GENERATION

| | | | | |
|---|---|---|---|---|
| 1 | | Analyse different types of Intermediate Code with an example. | [L4][CO5] | [12M] |
| 2 | | Explain Representation of Three Address Codes with suitable Examples | [L2][CO5] | [12M] |
| 3 | | Produce quadruple, triples and indirect triples for following expression: (x + y) * (y + z) + (x + y + z) | [L6][CO5] | [12M] |
| 4 | a | Discuss function preserving transformations. | [L2][CO6] | [6M] |
| | b | Describe about loop optimization technique . | [L2][CO5] | [6M] |
| 5 | | Explain the peephole optimization Technique with examples. | [L2][CO5] | [12M] |
| 6 | a | Define and Show Dead-code elimination with example. | [L1][CO4] | [6M] |
| | b | List and explain the Issues in the design of a code generator | [L2][CO6] | [6M] |
| 7 | a | Analyse the different forms in target program. | [L4][CO6] | [6M] |
| | b | Explain the target machine in code generator. | [L2][CO6] | [6M] |
| 8 | a | Define flow Graph | [L1][CO4] | [2M] |
| | b | Interpret optimization techniques on Basic Blocks with simple examples? | [L3][CO5] | [10M] |
| 9 | a | Analyze Simple code generator | [L4][CO6] | [6M] |
| | b | Evaluate Register allocation and register assignment techniques. | [L5][CO6] | [6M] |
| 10 | a | Create the DAG for following statement. a+b*c+d+b*c | [L6][CO6] | [4M] |
| | b | Construct the DAG for the following basic blocks<br>1. t1:=4*i<br>2. t2:=a[t1]<br>3. t3:=4*i<br>4. t4:=b[t3]<br>5. t5:=t2*t4<br>6. t6:=prod+t5<br>7. prod:=t6<br>8. t7:=i+1<br>9. i:=t7<br>if i<=20 goto 1 | [L6][CO6] | [8M] |

**Prepared by**:
1. **B Pavan kumar**
   **Assoc. Prof./CSE**