

III B.TECH II SEMESTER II MID TERM EXAMINATION

AUTOMATA THEORY AND COMPILER DESIGN

(COMMON TO IT/CSE-DS)

**PART-A 5X2=10**

1. What is difference between the decidable & undecidable problem. **(CO3,BTL1)**

2. What is the purpose of Lexical analyzer in compiler design? **(CO4, BTL1)**

3. Find First and Follow for following Grammar. **(CO4, BTL4)**

E->E+T/T

T->T\*F/F

F-> (E)/id

4. Define Syntax Directed Definition with an example**. (CO5, BTL1)**

5. Explain about Three Address code. **(CO5, BTL1)**

**PART-B 4X5=20**

**ANSWER ANY FOUR QUESTIONS FROM THE FOLLOWING**

6. What is Halting problem of Turning machine is it decidable or not? Explain.(**CO3**,**BTL2**

7. Design a Turing Machine that accepts even palindrome strings over the alphabet {a,b} (**CO3, BTL5)**

8. Explain about various phases of compiler in detail**.(CO4,BTL2)**

9. Construct Predictive parsing table for the following grammar**.(CO4,BTL3)**

E → TE′

E′ → +TE′|ε

T → FT′

T′ → \*FT′|ε

F → (E) |id and parse the string is id + id

10. Construct LALR parsing table for the given Grammar. (**CO4**, **BTL3**)

S->AA

A->aA/b

11. Construct a Quadruple, Triple and indirect triple for the statement (**CO5, BTL3**)

X= a + a\*(b-c)+(b-c)\*d.

**SCHEME OF EVALUATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **THEORY** | **MARKS** | **TOTAL** |
| 1 | **Part-A**  What is difference between the decidable & undecidable problem. | 2 | 10 |
| 2 | What is the purpose of Lexical analyzer in compiler design? | 2 |
| 3 | F First and Follow for following Grammar.  E->E+T/T  T->T\*F/F  F-> (E)/id | 2 |
| 4 | Define Syntax Directed Definition with an example**.** | 2 |
| 5 | Explain about Three Address code. | 2 |
| 6 | **Part-B**  . What is Halting problem of Turning machine is it decidable or not? Explain. | 2+3 |  |
| 7 | Design a Turing Machine that accepts even palindrome strings over the alphabet {a,b} | 5 |  |
| 8 | Explain about various phases of compiler in detail**.** | 5 | 20 |
| 9 | Construct Predictive parsing table for the following grammar**.**  E → TE′  E′ → +TE′|ε  T → FT′  T′ → \*FT′|ε  F → (E) |id and parse the string is id + id | 5 |  |
| 10 | Construct LALR parsing table for the given Grammar.  S->AA  A->aA/b | 5 |  |
| 11 | Construct a Quadruple, Triple and indirect triple for the statement  X= a + a\*(b-c)+(b-c)\*d. | 5 |  |
| **TOTAL MARKS** | | 30 | **30** |