C. U. SHAH UNIVERSITY
Winter Examination-2019

Subject Name : Translator Design

Subject Code : 4TE07TDE1   **Branch: B.Tech (CE)**

Semester : 7   **Date : 20/11/2019**   **Time : 10:30 To 01:30**   **Marks : 70**

Instructions:
(1) Use of Programmable calculator & any other electronic instrument is prohibited.
(2) Instructions written on main answer book are strictly to be obeyed.
(3) Draw neat diagrams and figures (if necessary) at right places.
(4) Assume suitable data if needed.

Q-1   **Define a to g:**   **(14)**
   a) Language Processor
   b) Handle Pruning
   c) Finite Automata
   d) Parsing
   e) Ambiguous Grammar
   f) Constant Folding
   g) Cross Compiler
   Differentiate between h to n:
   h) Lexeme and Token
   i) Macro and Subroutine
   j) Linker and Loader
   k) Synthesized attribute and Inherited attribute
   l) Static memory allocation and Dynamic memory allocation
   m) Front-end and Back-end of compiler
   n) Top-down parsing and Bottom-up parsing

Attempt any four questions from Q-2 to Q-8
Q-2   **Attempt all questions**
   a) Explain different phases of compiler.   **(07)**
   b) Explain symbol table management.   **(07)**

Q-3   **Attempt all questions**
   a) What are the different kinds of error in compiler design? Explain different error   **(07)**
        recovery strategies.
   b) Write quadruples, triples and indirect triples for the following expression: **(07)**
        (a+b) * (c+d) - (a+b+c)

Q-4   **Attempt all questions**
   a) Explain left factoring with suitable example.   **(03)**
   b) Explain any two parameter passing methods.   **(04)**

c) Construct DFA for given regular expression without constructing NFA and minimize it: (a | b | c) * a (b | c) * # **(07)**

Q-5 **Attempt all questions**
- **a)** Explain peephole optimization. **(07)**
- **b)** Construct LL(1) parsing table for following grammar. Check whether the grammar is LL(1) or not. **(07)**
  A → A a B | x
  B → B C b | C y
  C → C c | ε

Q-6 **Attempt all questions**
- **a)** List and explain the major steps of relocation and linking algorithms. **(07)**
- **b)** What is operator precedence grammar? Construct the operator precedence function for the given grammar: E → E + E | E * E | E ^ E | (E) | a **(07)**

Q-7 **Attempt all questions**
- **a)** Explain various issues in the design of a code generator. **(07)**
- **b)** What is importance of intermediate code? Discuss various representations of three address code using the given expression. a = b * -c + b * -c **(07)**

Q-8 **Attempt all questions**
- **a)** What is activation record? Explain stack allocation of activation records using example. **(07)**
- **b)** Explain principle sources of code optimization. **(07)**