Total No. of Pages 02 Roll No. 2K17/CO/045

SIXTH SEMESTER B.Tech[CO/SE]

END SEMESTER EXAMINATION *MAY-2017*

CO/SE-312: Compiler Design

Time: 3:00 Hours *Max. Marks: 70*

Note : 1) Q1 is compulsory.
2) Attempt any four questions from Q2 to Q7.

Q1. [14]

a) (0* / 1)* & (0 /1*)* are equivalent or not?
b) G= {N, { (,) } , { S-> SS, S-> (S), S-> ε },S} is a CFG to produce_____?
c) Recursive descent parser is a predictive or non-predictive parser?
d) All CFL's are regular languages or not?
e) YACC is a tool for _____?
f) _____ is a solution for common prefix problem?
g) Number of states in SLR and CLR are same or not?
h) _____ is used to identify common sub expressions?
i) Type conversion automatically done by complier is called _____?
j) Folding is the technique for _____?
k) Syntax tree is the form of _____?
l) A group of statements with single entry and single exit is called _____? (with respect to code generation)
m) Two adjacent transposed characters is an example of _____ error?
n) _____ is a hashing technique in which there is no limit on number of entries that can be made to the symbol table.

Q2. a) *Differentiate L-attributed and S-attributed definitions?* [7]
b) *Write SDT for generating three address code for an assignment statement.* [7]

Q3. a) *What is the principal criterion of choosing a hash function? Explain any three hashing functions.* [7]
b) *Explain the three storage allocation techniques?* [7]

Q4. a) *Describe machine dependent and independent phases of Compiler?* [7]
b) *Design a DFA (with input alphabet {0,1}) that accepts* [7]
i) *The set of strings where number of 0's is multiple of 3.*
ii) $L = \{(0\ 1)^i 1^{2j} / i>=1 , j >=1\}$

Q.5 Construct SLR(1) parsing table for the following grammer : [14]

S→xAy /xBy / xAz

A→aS / q

B→ q

(Also show canonical collection diagram & follow sets)

Q6. a) Discuss the desirable features of error reporter? [7]

b) Explain panic-mode error recovery strategy? [7]

Q7. a)Define a basic block? Write the steps involved in dividing the code into basic blocks. Findout the basic blocks in following piece of intermediate code. [7]

1. sum := 0
2. i := 1
3. If i> 10 goto 11
4. t_1 := 2*i
5. t_2 := a[t_1]
6. t_3 := sum+ t_2
7. sum := t_3
8. t_4 := i + 1
9. i := t_4
10. goto 3
11. t_5 := sum / i
12. Average = t_5

b) Explain any four optimizing transformations with examples? [7]