USN | | | | | | | | | | |

06CS63

Sixth Semester B.E. Degree Examination, December 2011
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain the different phases of compilation. **(10 Marks)**

b. Explain input buffering strategy, used in lexical analysis phase. **(10 Marks)**

2 a. Write the transition diagram for an unsigned number. **(04 Marks)**

b. Show that the following grammar is ambiguous: $E \rightarrow E + E \mid E * E \mid (E) \mid id$. Write an unambiguous grammar for the same. **(06 Marks)**

c. Write a recursive descent parser for the grammar : $S \rightarrow cAd$ $A \rightarrow ab/a$ and for the input 'cad' trace the parser. **(10 Marks)**

3 a. Construct the predictive parse table for the following grammar :

$S \rightarrow a \mid \uparrow \mid (T)$

$T \rightarrow T, S \mid S$ **(10 Marks)**

b. Explain the working of a shift reduce parser. **(05 Marks)**

c. Explain handle pruning. Explain the same for the grammar $E \rightarrow E + E \mid E * E \mid (CE) \mid id$ and the input string is id1 + id2 * id3. **(05 Marks)**

4 a. Consider the following grammar :

$S \rightarrow AS \mid b$

$A \rightarrow SA \mid a$

Construct the SLR parse table for the grammar. Show the actions of the parser, for the input string "abab". **(10 Marks)**

b. Construct the CLR parse table for the following grammar :

$S \rightarrow CC$

$C \rightarrow cC \mid d$ **(10 Marks)**

PART – B

5 a. Define the following with examples :
Synthesized attribute
Inherited attribute
S – attributed definitions
L – attributed definitions. **(08 Marks)**

b. Explain the parser stack implementation of post fix SDT, with an example. **(08 Marks)**

c. For the SDD shown below :

Production	Semantic rules
$L \rightarrow E_n$	L.val = E.val
$E \rightarrow E_1 + T$	E.val = E1.val + T.val
$E \rightarrow T$	E.val = T.val
$T \rightarrow T1 * F$	T.val = T1.val * F.val
$T \rightarrow F$	T.val = F.val
$F \rightarrow (E)$	F.val = E.val
$F \rightarrow$ digit	F.val = digit.lexval

construct the annotated parse tree for 3 * 5 + 4n. **(04 Marks)**

6 a. Explain the following, with an example :
   i) Quadraples
   ii) Triples
   iii) Indirect triples. **(09 Marks)**

b. Write an algorithm for the unification of a pair of nodes in a type graph. **(06 Marks)**

c. Explain syntax directed translation of switch statements. **(05 Marks)**

7 a. What is an activation record? Explain its possible structure. **(08 Marks)**

b. Explain the design goals for garbage collector. **(06 Marks)**

c. Explain the desirable properties of memory manager. **(06 Marks)**

8 a. What is next use information? Write an algorithm to determine the liveness and next use info for each statement in a basic block. Apply the same for the following basic block :
   3. T1 = Add(A) – 4
   4. T2 = 4 * i
   5. T3 = T1 [T2]
   6. Sum = Sum + T3
   7. I = I + 1
   8. If I ≤ 20 go to 3 **(12 Marks)**

b. Generate the intermediate code for the statement: sum = A [i, j] + B [i, j]. Construct DAG and simplify the code. **(08 Marks)**

* * * * *