Total No. of Pages:2 *Roll No..................*

SIXTH SEMESTER B. Tech.

END SEMESTER EXAMINATION *May-2019*

SE-306 COMPILER DESIGN

Time: 3 Hours *Max. Marks: 50*

Note: Attempt any five questions. Assume suitable missing data, if any.

Q.1 (a) What is the importance of Canonical LR (1) parser? Construct the Canonical LR parser for given grammar:

S → CC
C → C/ d

Also check the above grammar is LALR (1) or not. [7]

(b) What is the significance of Operator Precedence? Explain with suitable example. [3]

Q.2 (a) What is semantic rule? Write down semantic rule to convert infix notation to postfix notation. [3]

(b) What is intermediate code? Translate the expression (a+b)/(c+d)*(a+b/c)-d into Quadruples, triples and indirect triples. [4]

(c) Explain reducible and non-reducible flow graphs with example. [3]

Q.3 Write short note on:

(a) Array reference in arithmetic expression. [3]
(b) NFA and DFA with example [3]
(c) Loop Optimization techniques. [2]
(d) Ambiguity in Grammar and its removal. [2]

Q.4 (a) Differentiate inherited and synthesized attributes with an example. [2]

(b) Consider the following grammar

S --> E
E --> E + T
E --> T
T --> T * F
T --> F
F --> digit

Write the semantic action for above grammar and draw annotated parse tree for the input string
4 * 5 + 6. [4]

(c) Draw the LL(1) Parsing table for given grammar

$S \rightarrow iEtSS' | a$
$S' \rightarrow eS | e$
$E \rightarrow b$ [4]

Q.5 (a) Explain the error occurred in various phases of compiler. How to handle them? [5]

(b) Can we reuse the symbol table space? Explain through an example. [2]

(c) What is Activation Record? Explain its usage in stack allocation strategy. How it is different from heap allocation? [3]

Q.6 (a) What is a leader of basic block? Write and explain the algorithm used to find leaders.

Draw the flow graph for the given source code:

```
x = 20 while x < 10 Do
x = x-1;
A[x] = 10
If x = 4
then
x= x-2;
ENDIF;
ENDDO: y= x + 5
```

[5]

(c) Construct DAG for the basic block whose code is given below:

(1) $t_1 = 4*i$
(2) $t_2 = [t_1]$
(3) $t_3 = sum + t_2$
(4) $sum = t_3$
(5) $t_4 = i + 1$
(6) $i = t_4$
(7) if I <= 10 goto (3) [5]