Registration no: □□□□□□□□□□

Total Number of Pages: 02 **B.Tech**

PCCS4305

6th Semester Regular / Back Examination 2016-17

COMPILER DESIGN

BRANCH: CSE

Time: 3 Hours

Max Marks: 70

Q.CODE: Z175

Answer Question No.1 which is compulsory and any five from the rest.
The figures in the right hand margin indicate marks.

Q1 **Answer the following questions:** **(2 x 10)**

a) Give examples on a token and lexeme.

b) Differentiate between a compiler and an interpreter.

c) What is a handle used in parsing technique?

d) Differentiate between top-down and bottom up parsing.

e) Write at least four languages which are not regular languages.

f) What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon- and unit-production (i.e., of type A -> ϵ and A -> a) to parse a string with n tokens?

g) Briefly discuss about Peephole optimisation.

h) What do you mean by DAG? Give an example of its need in compilation process.

i) What are the Error-recovery actions in a lexical analyzer?

j) Define LR(k) Parsing.

Q2 **a)** Symbol table is necessary for compiler construction. Justify your statement with example. **(2)**

b) Explain ambiguity in grammars. Consider the grammar S→aSbS/bSaS/ε Show that the grammar is ambiguous by constructing two different leftmost derivations for the sentence abab. **(8)**

Q3 **a)** Construct the NFA for the regular expression $(0|1)^*011$ and convert the NFA to DFA after minimizing the number of states. **(5)**

b) Discuss the working principle of Shift-reduce parsing technique with an example. **(5)**

Q4 **a)** What is LEX? Explain, in detail, different sections of LEX program with a suitable example. **(5)**

b) What are the First and Follow sets for the following grammar? The start nonterminal is E. **(5)**

E -> A
E -> L
A -> n
A -> i
L -> (S)
S -> E , S
S -> E

Q5 **a)** Describe the relationship between a *production* and an *item* in an LR(0) grammar. How does this relate to the notion of the *stack* in an LR(0) grammar? Give an algorithm for constructing the *closure* of a set of items LR(0) with respect to a particular grammar. **(5)**

b) For the following grammar, construct the set of LR(0) states to recognize viable prefixes of this language. Then fill out an SLR parse table for this grammar and indicate whether the grammar is ambiguous. **(5)**

- S -> A $
- A -> -- A B | ++ A B | id B
- B -> -- B | ++ B |

Q6 **a)** What are the main operations for a symbol table? Discuss the data structures associated with a symbol table maintained as a list of hash tables and how the operations of a symbol table are implemented in that case. Give an example of what your symbol table would look like for a sample program. **(5)**

b) Explain briefly, precedence functions. Construct the precedence graph using the following precedence table. **(5)**

	+	*	(	)	id	$
f	2	3	0	4	4	0
g	1	3	5	0	5	0

Q7 Below is a grammar for understanding simple arithmetic expressions: **(10)**

E -> E * E | E + E | (E) | int

Assuming that precedence and associativity have been handled, what translation actions would you add to the grammar to get it to print out the input expression in a postfix notation (e.g., (3 + 5) * 4 would print out as 3 5 + 4 *). Now add actions to print out the expression in a prefix notation (* + 3 5 4).

Q8 **Write short answer on any TWO:** **(5 x 2)**

a) DAG

b) LL(1) parser

c) Handle

d) Global Data Flow