Exam. Code: 0918
Sub. Code: 33446

2015
B.E. (Computer Science and Engineering)
Sixth Semester
CS-604: Complier Design

Time allowed: 3 Hours **Max. Marks: 50**

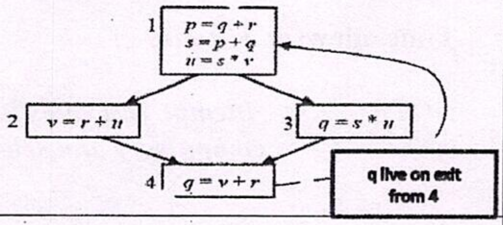
NOTE: ***Attempt five questions in all, including Question No. 1 (Section-A) which is compulsory and selecting two questions each from Section B-C.***

x-x-x

	Section-A	
Q1.	a) State merges will not produce SR conflict but can lead to RR conflict. Justify the above statement. b) Differentiate between DAG and CFG. c) What advantages and disadvantages of various data structures used for symbol table? d) Write syntax directed translation scheme for infix to postfix conversion for a suitable grammar. e) How can we use ambiguous grammar for parsing? Why such grammars are useful?	10
	Section-B	
Q2.	a) What are different phases of the compiler? Explain the phases in detail. Write down the output of each phase for the expression a:=b+c^60.	4
	b) Build the operator precedence table for given grammar. Also build function table for the same. Write advantages and disadvantages of using function tables. E -> E + T | T T -> T * F | F F -> (E) | id Also show its working for the input string: 2 + 3 * 4 – 5	6
Q3.	a) Construct DFA without constructing NFA for following regular expression. Also Find minimized DFA. a*b*a(a/b)*b*a	5
	b) What are conflicts in bottom-up parsing. What are the necessary conditions for the conflicts to appear in LR(0),SLR, CLR and LALR parsers.	5
Q4.	a) Identity if the Grammar is LL(1) S -> iEtSS' | a S' -> eS | Ɛ E -> b Additionally, You are supposed to find the reason for its LL(1) or non-LL(1) properties. Justify your answer with sample input string. What can be done to make this grammar LL(1)?	6
	b) Write a short note on error recovery techniques of different parsers.	4
	Section-C	
Q5.	a) What is DAG? Explain the value-number method for representing a node in a DAG	3
	b) E -> E + T E-> E – T E → T T -> (E) T → id T → num Write appropriate SDT to generate syntax tree for the expression: **a- 4+ c**	7

P.T.O.

Sub. Code: 33446

Q6.	a) For the expression **x= (a+b) – (e-(c+d))**, use labelling algorithm to generate code for it. Explain its working clearly depicting the output at every step. b) For the given graph, find all the live variables at each point in the program. Assume q is live on exit.  1 p=q+r s=p+q u=s*v 2 v=r+u 3 q=s*u 4 q=v+r q live on exit from 4	6 4
Q7.	a) Generate intermediate code for the following statement a< b and c< d or e<f b) Explain loop unrolling, common subexpression elimination, code motion, peephole optimization, constant folding, strength reduction.	5 5

x-x-x