B.E / B.Tech (Full-Time) Degree End Semester Examinations, Nov/Dec 2011
Anna University, Chennai

Computer Science and Engineering
Sixth Semester

CS9353 PRINCIPLES OF COMPILER DESIGN

Time 3 Hrs **Answer all Questions** **100 Marks**

PART A – (10 X 2 = 20 Marks)

1. Name few tools that could be used for the various phases of the compiler
2. Explain about input buffering.
3. Design a syntax directed translation scheme for constructing syntax tree for a postfix expression by writing appropriate grammar.
4. Write short notes on Heap management.
5. Explain the need for quadruples and indirect triples.
6. What is the need for next-use information?
7. What are the advantages of representing the input using DAG?
8. What is meant by contiguous evaluation of expressions in dynamic programming code generation algorithm? State it's advantage.
9. Define the need for optimization.
10. What type of optimization computes the available expressions?

PART B – (5 x 16 = 80)

11. i. With reference to the phases of the compiler, identify the sequence of steps that would take place for the input ans := (a + - b * 5.3) ^ 2 ^ (1 – f) [8]

    ii. Describe the storage allocation strategies used at run time for static allocation, stack allocation and heap allocation. [8]

12. a. i. Construct the predictive parsing table for the grammar

S → iEtS | iEtSeS | a

E → b

Find whether this grammar is LL(1) or not and give reason for your answer. [12]

ii. What is the need for removing left recursion from a grammar to be parsed using predictive parsing? [4]

(OR)

b. Construct SLR parsing table for the following context free grammar [16]

E → E + T | T

T → T * F | F

F → (E) | id

Check whether the string id + (id*id) is valid using the parse table

13. a. i. Write the translation scheme for translating assignment statements having scalar variables and array references to three-address statements. [8]

ii. Explain the various types of 3- address code. [8]

(OR)

b. Write the production rules needed for recognizing the a program block along with the translations using back patching for each rule if the program block consists of ' 'while' and 'if - else' statements [16]

14. a. i. Discuss any four important issues in the code generation phase. [6]

ii. Apply the code generation algorithm for the following basic block and find the object code produced for a machine with exactly two registers namely R1 and R2. The next use information, address descriptors, register descriptors should be taken into account while looking for registers to carry out the computation. Initially the register descriptors for all registers are empty and it is not required to keep any variable live on exit from the basic block.

a = b * c; c = a + d; d = d - c; c = a + d; d = c + d;

(OR)

b. i. Construct a DAG for the following basic block and find the optimal ordering of instructions for code generation which requires minimal number of registers. [8]

$t_1 = a + b$; $t_2 = t_1 - c$; $t_3 = d + e$; $t_4 = t_2 * t_3$; $t_5 = t_4 - e$; $t_6 = t_1 + t_4$; $t_7 = t_6 * t_5$;

ii. Construct Dynamic programming based approach to generate code, for the following expression, a: = b+c*d-e/f, using only two registers [8]

15. a. i. Define peephole and explain all the transformations carried out in peephole optimization. [10]

ii. Explain the sources of optimization in detail. [6]

(OR)

b. i. Write the data flow equations for basic blocks and derive the data flow equations for the most familiar three program control constructs [10]

ii. Compute the definitions in and out of each basic block from the flowgraph given here. [6]