COLLEGE OF ENGINEERING AND MANAGEMENT PUNNAPRA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMPILER DESIGN

MODULE V & VI

OLD UNIVERSITY QUESTIONS

APRIL 2018

1. Write syntax directed definitions to construct syntax tree and three address code for assignment statements.
2. Explain quadruples and triples with an example each.
3. Construct the syntax tree and then draw the DAG for the statement e := (a*b) + (c-d) * (a*b)
4. Explain static allocation and heap allocation strategies.
5. With an example each explain the following loop optimization techniques: (i) Code motion (ii) Induction variable elimination and (iii) strength reduction
6. Explain any two issues in the design of a code generator.
7. Explain the optimization of basic blocks.
8. Write the Code Generation Algorithm and explain the *getreg* function.
9. Generate a code sequence for the assignment d=(a-b)+(a-c)+(a-c)

MAY 2019

10. Explain storage organization and storage allocation strategies
11. Explain intermediate code generation of an assignment statement
12. Explain quadruples, triples and dags with an example each
13. Explain the principal sources of optimization
14. Explain optimization of basic blocks
15. With suitable examples explain loop optimization
16. Explain issues in design of a code generator
17. Explain simple code generation algorithm

DECEMBER 2019

18. Explain how DAGs help in intermediate code generation?
19. Explain the code generation algorithm. Illustrate with an example
20. Define the following and show an example for each.
    i). Three-address code iii). Triples ii). Quadruples iv). Indirect triples
21. State the issues in design of a code generator
22. Explain different stack allocation strategies with suitable examples.

23. Explain different code optimization techniques available in local and global optimizations?
24. How is storage organization and management done during runtime?
25. How the optimization of basic blocks is done by a compiler?
26. Write the algorithm for partitioning a sequence of three-address instructions into basic blocks
27. Construct the DAG and three address code for the expression a+a*(b-c)+(b-c)*d