

JK LAKSHMIPAT UNIVERSITY, JAIPUR

INSTITUTE OF ENGINEERING AND TECHNOLOGY

End Term Examination (Regular), April-May 2016
M. Tech in Computer Science Engineering, Semester II

Subject Code: MTCS201

Roll No.....

Subject Name: Advanced Compilers

Time: 3 hours

Max. Marks: 100

Instructions to students:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever required and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics/ flow chart wherever, required are an integral part of the answer.

Q.1A Objective type multiple choice questions (10 questions @ 1 mark)

10

- i. A Lex compiler generates
 - a. Lex object code
 - b. Transition tables
 - c. C tokens
 - d. None of these
- ii. Yacc semantic action is a sequence of
 - a. Tokens
 - b. Expression
 - c. C statement
 - d. Rules
- iii. In which addressing mode the operand is given explicitly in the instruction?
 - a. Absolute mode
 - b. Immediate mode
 - c. Indirect mode
 - d. Index mode
- iv. DAG representation of a basic block allows
 - a. Automatic detection of local common sub expressions
 - b. Automatic detection of induction variables
 - c. Automatic detection of loop invariant
 - d. None of the above
- v. We have the following basic block:
$$T1 = a + b$$
$$T2 = c + d$$
$$T3 = e - T2$$
$$T4 = T1 - T3$$

Assuming that all operands are initially in memory and the final value of the computation in memory, the minimum number of MOV instructions in the code generated for this block is

 - a. 3
 - b. 2
 - c. 6
 - d. 4
- vi. Which of the following are true for an optimizing compiler:
 1. is optimized to occupy less space
 2. is optimized to take less time for execution
 3. optimized the code

(a) 1 and 2 only (b) 1 and 3 only (c) 2 and 3 only (d) 1, 2 and 3
- vii. The languages that need heap allocation in the runtime environment are
 - a. Those that use global variables
 - b. Those that use dynamic scoping
 - c. Those that use support recursion
 - d. Those that allow dynamic data structure
- viii. In later iterations, data accesses are dependent on values of data produced in earlier iterations; this dependence is called a
 - a. Data unit
 - b. Loop-carried unit



	c. Data dependence	d. Loop-carried dependence	
ix.	When instruction i and instruction j are tend to write same register or memory location, it is called		
	a. Input dependence	b. Output dependence	c. Ideal pipeline d. Digital call
x.	Which of the following is true about Just in Time (JIT) compilers		
	i. Ahead of time (AOT) delivers faster start-up time, especially in large applications where mush code executes on startup. But it requires more disk space and more memory/virtual address space to keep both the intermediate language and precompiled images. In this case, the JIT compiler has to do a lot of disk I/O actions, which are quite expensive.		
	ii. JIT can generate faster code, because it targets the current platform of execution. AOT compilation must target the lowest common denominator among all possible execution platforms.		
	iii. JIT can profile the application while it runs, and dynamically re-compile the code to deliver better performance in the hot path.		
	a. i and ii only b. ii and iii only c. i and iii only d. all of the above		
Q.2	Short answer type questions (4 questions @ 2.5 marks)		10
	i. Elaborate the concept of dependency elimination and avoidance in brief.		
	ii. Briefly describe the parallelizing code and data locality.		
	iii. How data privatization can remove anti and output dependencies?		
	iv. How the parallelism can be detected using the distance and direction vectors?		
Q.3A	Discuss the various issues which may raise during the construction of a compiler.		05
Q.3B	Elaborate the approach used in designing of a compiler.		05
Q.4A	A Lex program is given below Auxiliary definitions (none) Translation Rules a abb a*b ⁺ Implement the Lex program as DFA.		05
Q.4B	What is the main functionality of a LALR parser generator "YACC"? Show that the following grammar is not LALR(1): $S \rightarrow Aa \mid bAc \mid Bc \mid bBa$ $A \rightarrow d$ $B \rightarrow d$		05
Q.5A	Generate the low-level assembly code equivalent to the following three address code statements: A = B + C B = B - D C = C + D E = B + C		05
Q.5B	Explain the register allocation techniques with example.		05

Q.6A Explain the various steps involved in an algorithm of a simple code generator. 05

Q.6B What do you mean by Static Single Assignment (SSA)? Consider the following normal form: 05

a = 2

b = 5

x = a + b

x = x + 3

y = a + b

Obtain the optimized normal form for the above normal form using SSA.

OR

Q.6B Briefly describe the bi-directional methods of data flow analysis. Also enlist the key advantages of each method. 05

Q.7A Define the parallelizing compilers. Discuss the following w.r.t. the parallelizing compilers: 05

i. Importance of parallelizing the compilers

ii. Role of parallelizing compiler

iii. Steps to parallelizing a program

Q.7B Briefly describe the following parallelism detection techniques: 05

i. Loop Carried Dependence (LCD)

ii. Loop Independent Dependence (LID)

OR

Q.7B Define the term data partitioning. Briefly elaborate the methods of data partitioning. 05

Q.8A Construct DAG of basic blocks after converting the following code in three-address representation: 10

i = 1;

j = 2;

repeat

A[i] = j;

j = j * 2;

i = i + 1;

until (i > 10)

Q.9A i. Explain the branch and list algorithms of instruction scheduling. 05

ii. Discuss the various issues associated with basic block scheduling. 05

Q.9B i. What is machine optimization? Explain the machine independent and machine dependent optimizations. 05

ii. Explain the role of dynamic optimization in compilation. 05

OR

Q.9B Write short note on the following: (any two) (@ 5 marks each) 10

i. Data flow analysis

ii. Just in time compilers

iii. Auto scheduling compilers

