



B.E DEGREE EXAMINATIONS : MAY 2015

(Regulation 2009)

Sixth Semester

COMPUTER SCIENCE AND ENGINEERING

ITY112:Principles of Compiler Design

Time: Three Hours

Maximum Marks: 100

Answer all the Questions:-

PART A (10 x 1 = 10 Marks)

- A compiler which allows only the modified section of the code to be recompiled is called as
 - Incremental Compiler
 - Reconfigurable Compiler
 - Dynamic compiler
 - Selective compiler
- Write a regular expression for denoting an identifier which starts with a letter followed by any number of letters or digits. Assume L and D are the sets of letters and digits respectively
 - $(LuD)^*$
 - $L(LuD)^*$
 - $(L . D)^*$
 - $L . (L . D)^*$
- What is meant by LL(1) grammar?
 - Scanning the input one symbol at a time from Left to right, Rightmost Derivation
 - Scanning the input one symbol at a time from right to left, Leftmost Derivation
 - Scanning the input one symbol at a time from right to left, Rightmost Derivation
 - Scanning the input one symbol at a time from Left to right, Leftmost Derivation
- Determine the FIRST (S) in the following grammar.
 $S \rightarrow aSa$
 $S \rightarrow bSb$
 $S \rightarrow \epsilon$
 - $FIRST(S) = \{ a, b, \epsilon \}$
 - $FIRST(S) = \{ \epsilon \}$
 - $FIRST(S) = \{ b, \epsilon \}$
 - $FIRST(S) = \{ a, \epsilon \}$

5. Intermediate code generation phase gets input from
 - a) Lexical Analyzer
 - b) Syntax Analyzer
 - c) Semantic Analyzer
 - d) Error Handling
6. Type checking is done during
 - a) Lexical Analysis
 - b) Syntax Analysis
 - c) Code Optimization
 - d) Syntax Directed Translation
7. The graph that shows the basic blocks and their relationship is called as
 - a) DAG
 - b) Flow graph
 - c) Control Graph
 - d) Hamiltonian Graph
8. Peephole optimization emphasizes on
 - a) Loop optimization
 - b) Local optimization
 - c) Constant folding
 - d) Data flow analysis
9. Local and Loop optimization provide motivation for
 - a) Data flow analysis
 - b) DFA and Constant Folding
 - c) Constant Folding
 - d) Peephole Optimization
10. Software that allows the computer to interact with the users, application and hardware is called
 - a) application software
 - b) system software
 - c) word processor
 - d) database software

PART B (10 x 2 = 20 Marks)

11. Define patterns, lexeme and tokens with an example.
12. State the error recovery actions.
13. Compare DFA with NFA.
14. Eliminate the left recursion for the following CFG

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Ac \mid Sd \mid \varepsilon$$
15. Construct the three address code for the expression $a < b$ or $c < d$.
16. Define back patching.
17. Construct a DAG for the expression $a := b * -c + b * -c$.
18. Develop the basic blocks for the three address code given below
 1. $prod := 0$
 2. $i = 1;$
 3. $t1 := 4 * i$
 4. $t2 := a[t1]$

5. $t3 := 4 * i$
6. $t4 := b[t3]$
7. $t5 := t2 * t4$
8. $t6 := \text{prod} + t5$
9. $\text{prod} := t6$
10. $t7 := i + 1$
11. $i := t7$
12. if $i \leq 20$ goto (3)

19. Apply any one optimization technique to optimize the code given below.

```

t1: = 4*i
t2: = a [t1]
t3: = 4*j
t4: = 4*i
t5: = n
t6: = b [t4] +t5

```

20. List the different types of loops in flow graphs.

PART C (5 x 14 = 70 Marks)

21. a) Illustrate the output of each phase of compiler for the expression $a := b * c - d$ and explain how to derive the output, where a & c are integers and b & d are real numbers.

(OR)

b) Construct an NFA for the regular expression $(a^+ / b^*) . ab$. Convert it into DFA and minimize it.

22. a) (i) Discuss about the possible conflicts that can occur in shift – reduce parsing with suitable example. (6)

(ii) Determine whether the following grammar is SLR(1) or not. Justify your answer. (8)

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow \text{id}$

$R \rightarrow L$

(OR)

- b) Propose a solution using predictive parsing technique to parse the string $id+id*id$ for the CFG given below.

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

23. a) Elaborate the different kinds of representations of intermediate code with suitable examples. Explain their implementations.

(OR)

- b) Explain the following with examples.

- a. Synthesized Attributes (4)
- b. Inherited Attributes (3)
- c. Dependency Graph (3)
- d. Evaluation order of attributes (4)

24. a) Discuss about the issues that arise during the code generation phase.

(OR)

- b) (i) Design a simple code generator to demonstrate a simple assignment statement $d := (a-b) + (a-c) + (a-c)$ and explain (8)
- (ii) Construct a DAG for the expression $((a+b)*e) + a + b + e$ and discuss the algorithm (6) used for the construction of DAG.

25. a) Explain the principal sources of optimization in detail.

(OR)

- b) Explain the storage allocation strategies used in run time environments
