

R18

Code No: 156AH

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech III Year II Semester Examinations, March - 2024

COMPILER DESIGN

(Common to CSE, ECM, CSE(DS))

Time: 3 Hours

Max. Marks: 75

Note: i) Question paper consists of Part A, Part B.

ii) Part A is compulsory, which carries 25 marks. In Part A, Answer all questions.

iii) In Part B, Answer any one question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART - A

(25 Marks)

- 1.a) What do you understand by backtracking? [2]
- b) How does the lexical analyzer handle whitespace and comments in source code? [3]
- c) How can parser generators help in automating the generation of parsers for programming languages? [2]
- d) What are the advantages and disadvantages of top-down parsing? [3]
- e) How are switch statements represented in intermediate code? [2]
- f) What are the different evaluation orders for Syntax-Directed Definitions? [3]
- g) Define basic blocks and flow graphs. [2]
- h) How does the run-time environment provide access to nonlocal data on the stack? [3]
- i) What do you understand by partial-redundancy elimination? [2]
- j) List the applications of data-flow analysis in compiler optimization. [3]

PART - B

(50 Marks)

- 2.a) Suppose a source program contains the assignment statement
 $position = initial + rate * 60$
Show the translation of this assignment statement in each phase of the compiler.
- b) Draw the transition diagrams for unsigned numbers and whitespace. [5+5]

OR

- 3.a) Write a Lex program that copies a file, replacing each non-empty sequence of white space with a single blank.
- b) Suppose we have two tokens, the keyword if and identifiers, which are strings of letters other than if. Show (i) the NFA for these tokens and (ii) the DFA for these tokens. [5+5]
- 4.a) What language is generated by the following grammars? In each case justify your answer.
(i) $S \rightarrow 0 S 1 \mid 0 1$ (ii) $S \rightarrow + S S \mid - S S \mid a$
- b) Construct recursive-descent parsers for the following grammar. [5+5]
 $S \rightarrow S (S) S \mid \epsilon$

OR

QA QA QA QA QA QA QA G

5. Show that the following grammar is LL(1) but not SLR(1).

$S \rightarrow A a A b \mid B b B a$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

QA QA QA QA QA QA QA [10] QA G

6.a) Construct a syntax-directed translation scheme to translate postfix arithmetic expressions into equivalent prefix arithmetic expressions.

b) Give an SDD to differentiate expressions such as $x * (3 * x + x * x)$ involving the operators $+$ and $*$, the variable x , and constants. Assume that no simplification occurs, so that, for example, $3 * x$ will be translated into $3 * 1 + 0 * x$. [5+5]

QA QA QA QA QA QA QA OR QA QA QA QA G

7.a) Show how to transform a three-address code sequence into one in which each defined variable gets a unique variable name.

b) Write the unification algorithm of type checking. [5+5]

8.a) Explain the static and dynamic storage allocation.

b) Write Cheney's copying collector algorithm in the context of trace-based collection. [5+5]

QA QA QA QA QA QA QA OR QA QA QA QA G

9.a) Discuss the design goals for garbage collectors.

b) Generate code for the following three-address sequence assuming that p and q are in memory locations.

$y = *q$

$q = q + 4$

$*p = y$

$p = p + 4$

QA QA QA QA QA QA QA [5+5] QA G

10.a) Discuss the lazy-code-motion problem.

b) Show that every depth-first presentation for every nonreducible flow graph has a retreating edge that is not a back edge. [5+5]

OR

11. Explain the following terms with suitable examples. (a) Copy motion (b) Induction variables and reduction in strength. [5+5]

QA QA QA QA QA QA QA QA QA QA G

---ooOoo---

QA QA QA QA QA QA QA QA QA QA G

QA QA QA QA QA QA QA QA QA QA G