Student Name: Om Swami Roll No: [illegible]

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, LUCKNOW

COMPILER DESIGN END SEMESTER EXAM

Maximum marks: 70
Date: 22-05-25

Duration: 180 mins
Instructor: Dr. Abhinesh Kaushik

Note:

- Read questions carefully. Understand a question before you start writing.
- Be clear and to the point. You will lose points for rambling and for incorrect or irrelevant statements
-

Ques 1. Here are two grammars for languages built from terminals a and b:

Grammar A:	Grammar B:
$S \rightarrow C\ a\ C\ b\ A$	$S \rightarrow A\ b$
$A \rightarrow B$	$A \rightarrow \epsilon$
$A \rightarrow S$	$A \rightarrow A\ a\ b$
$B \rightarrow \epsilon$	
$B \rightarrow b$	
$C \rightarrow \epsilon$	

In both grammars S is the start symbol.

(a) The languages L(A) and L(B) for grammars A and B are not equal. 2
   1. Give an example of a string that is in L(A) but not in L(B).
   2. Give an example of a string that is in L(B) but not in L(A).

(b) Give regular expressions describing each of L(A) and L(B). 4

(c) Compute first sets and follow sets for each grammar. 3+3

(d) State with justification whether each grammar is LL(1) and whether it is SLR(1). 4+4

Ques 2. Explain the Following terms: 12

a. Syntax Tree
b. Parse Tree
c. Handles
d. Context Free Grammar
e. Backtracking
f. Peephole optimization

Ques 3. What is meant by GoTo and Closure function in CLR parsers? Give an example and also write the algorithm for GoTo function, Closure function and CLR Parser. 2+2+2+2+2

Ques 4. 2+4

a. An identifier in a programming language consists of upto six letters and digits of which the first character must be a letter. Derive a regular expression for the identifier.

b. Build an LL(1) parsing table for the language defined by the LL(1) grammar with productions

- Program → begin *d* semi X end
- X → *d* semi X | s Y

- Y → semi s Y | ε

Ques 5. Define operator Grammar and its algorithm and explain operator precedence parsing with the help of an example 2+2+4

Ques 6.

a. Draw DAG to represent a[i]=b[i]; a[i]=&t; 4

b. Consider the intermediate code given below and draw the control flow graph:

1. i = 1
2. j = 1
3. t1 = 5 * i
4. t2 = t1 + j
5. t3 = 4 * t2
6. t4 = t3
7. a[t4] = –1
8. j = j + 1
9. if j <= 5 goto(3)
10. i = i + 1
11. if i < 5 goto(2)

The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are 4

Ques 7. 3+3

a. Consider the following Syntax Directed Translation Scheme (SDTS), with non-terminals {S, A} and terminals {a, b}}.

$$S \longrightarrow aA \quad \{ \text{print } 1 \}$$
$$S \longrightarrow a \quad \{ \text{print } 2 \}$$
$$A \longrightarrow Sb \quad \{ \text{print } 3 \}$$

Using the above SDTS, the output printed by a bottom-up parser, for the input **aab** is 2 3 1

b. Consider the translation scheme shown below

S → T R
R → + T {print ('+');} R | ε
T → num {print (num.val);}

Here num is a token that represents an integer and num.val represents the corresponding integer value. For an input string '9 + 5 + 2', this translation scheme will print.