

Reg No.: \_\_\_\_\_

Name: \_\_\_\_\_

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

Sixth Semester B.Tech Degree Regular and Supplementary Examination July 2021

**Course Code: CS304****Course Name: COMPILER DESIGN**

Max. Marks: 100

Duration: 3 Hours

**PART A***Answer all questions, each carries 3 marks.*

Marks

- 1 Write a regular expression to denote the language of all strings of a's and b's with an even number of a's followed by an odd number of b's. (3)
- 2 Distinguish between front end and back end of a compiler. (3)
- 3 Show that the following grammar is ambiguous : (3)

$$S \rightarrow i C t S$$

$$S \rightarrow i C t S e S$$

$$S \rightarrow a$$

$$C \rightarrow b$$

- 4 Explain backtracking with an example. (3)

**PART B***Answer any two full questions, each carries 9 marks.*

- 5 a) Eliminate left recursion from the following grammar: (4)

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Ac \mid Sd \mid h$$

- b) Construct a recursive descent parser for the following grammar (5)

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow ( E ) \mid id$$

- 6 Explain in detail the various phases of a compiler with a neat diagram. Illustrate (9)

Reg No.: \_\_\_\_\_

Name: \_\_\_\_\_

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

Sixth Semester B.Tech Degree Regular and Supplementary Examination July 2021

**Course Code: CS304****Course Name: COMPILER DESIGN**

Max. Marks: 100

Duration: 3 Hours

**PART A***Answer all questions, each carries 3 marks.*

Marks

- 1 Write a regular expression to denote the language of all strings of a's and b's with an even number of a's followed by an odd number of b's. (3)
- 2 Distinguish between front end and back end of a compiler. (3)
- 3 Show that the following grammar is ambiguous : (3)

$$S \rightarrow i C t S$$

$$S \rightarrow i C t S e S$$

$$S \rightarrow a$$

$$C \rightarrow b$$

- 4 Explain backtracking with an example. (3)

**PART B***Answer any two full questions, each carries 9 marks.*

- 5 a) Eliminate left recursion from the following grammar: (4)

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Ac \mid Sd \mid h$$

- b) Construct a recursive descent parser for the following grammar (5)

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow ( E ) \mid id$$

- 6 Explain in detail the various phases of a compiler with a neat diagram. Illustrate (9)

the output of each phase for the input  $x = 2 * a + b$ , where  $a$  and  $b$  are float variables.

- 7 a) Check if following grammar is LL(1) by constructing a parse table: (5)

$$S \rightarrow ( L ) \mid a$$

$$L \rightarrow S L'$$

$$L' \rightarrow , S L' \mid \epsilon$$

Note that ‘,’ is a terminal and  $\epsilon$  is the empty string.

- b) Explain bootstrapping. (4)

### PART C

*Answer all questions, each carries 3 marks.*

- 8 Determine the FIRST and FOLLOW sets for the non-terminals in the following grammar: (3)

$$S \rightarrow AA$$

$$A \rightarrow aA \mid b$$

- 9 Define an operator grammar. Give an example (3)

- 10 Distinguish between S-attributed definitions and L-attributed definitions. (3)

- 11 What is type checking? What are its two types? (3)

### PART D

*Answer any two full questions, each carries 9 marks.*

- 12 Construct the SLR(1) parsing table for the following grammar: (9)

$$E \rightarrow T + E \mid T$$

$$T \rightarrow id$$

- 13 a) Write a syntax directed translation scheme for a simple desk calculator. (3)

- b) What sequence of moves are made by a bottom up parser on the input  $23*5+4$  (6) using the translations in part (a)?

- 14 a) Consider the following grammar: (6)

$$S \rightarrow a \mid ( T )$$

$$T \rightarrow T, S \mid S$$

For the string  $(a, (a, a))$ , indicate how a parse tree is constructed by a shift reduce parser using a rightmost derivation.

- b) Explain the bottom- up evaluation of S- attributed definitions. (3)

the output of each phase for the input  $x = 2 * a + b$ , where  $a$  and  $b$  are float variables.

- 7 a) Check if following grammar is LL(1) by constructing a parse table: (5)

$$S \rightarrow ( L ) \mid a$$

$$L \rightarrow S L'$$

$$L' \rightarrow , S L' \mid \epsilon$$

Note that ‘,’ is a terminal and  $\epsilon$  is the empty string.

- b) Explain bootstrapping. (4)

### PART C

*Answer all questions, each carries 3 marks.*

- 8 Determine the FIRST and FOLLOW sets for the non-terminals in the following grammar: (3)

$$S \rightarrow AA$$

$$A \rightarrow aA \mid b$$

- 9 Define an operator grammar. Give an example (3)

- 10 Distinguish between S-attributed definitions and L-attributed definitions. (3)

- 11 What is type checking? What are its two types? (3)

### PART D

*Answer any two full questions, each carries 9 marks.*

- 12 Construct the SLR(1) parsing table for the following grammar: (9)

$$E \rightarrow T + E \mid T$$

$$T \rightarrow id$$

- 13 a) Write a syntax directed translation scheme for a simple desk calculator. (3)

- b) What sequence of moves are made by a bottom up parser on the input  $23*5+4$  using the translations in part (a)? (6)

- 14 a) Consider the following grammar: (6)

$$S \rightarrow a \mid ( T )$$

$$T \rightarrow T, S \mid S$$

For the string  $(a, (a, a))$ , indicate how a parse tree is constructed by a shift reduce parser using a rightmost derivation.

- b) Explain the bottom- up evaluation of S- attributed definitions. (3)

**PART E**

*Answer any four full questions, each carries 10 marks.*

- 15 a) What is heap allocation strategy? (3)
- b) What is an activation record? Explain its structure with a figure. (7)
- 16 Write quadruples, triples and indirect tuples for the expression (10)  
$$(a + b) * (b + c) + (a + b + c)$$
- 17 a) Write a syntax directed translation scheme that generates three address code for Boolean expressions. (7)
- b) Distinguish between static and dynamic storage allocation. (3)
- 18 a) Write the algorithm for identifying the basic blocks from a sequence of three address code statements. (5)
- b) Construct the DAG for the following basic block (5)
- D := B \* C  
E := A + B  
B := B \* C  
A := E - D
- 19 For the following C statement, write the three-address code. (10)  
X := A - B + C - D + E - F
- Convert the three-address code into machine code.
- 20 a) How do algebraic laws help in optimizing basic blocks? (4)
- b) Write the code generation algorithm. (6)

\*\*\*\*

**PART E**

*Answer any four full questions, each carries 10 marks.*

- 15 a) What is heap allocation strategy? (3)
- b) What is an activation record? Explain its structure with a figure. (7)
- 16 Write quadruples, triples and indirect tuples for the expression (10)  
$$(a + b) * (b + c) + (a + b + c)$$
- 17 a) Write a syntax directed translation scheme that generates three address code for Boolean expressions. (7)
- b) Distinguish between static and dynamic storage allocation. (3)
- 18 a) Write the algorithm for identifying the basic blocks from a sequence of three address code statements. (5)
- b) Construct the DAG for the following basic block (5)
- D := B \* C  
E := A + B  
B := B \* C  
A := E - D
- 19 For the following C statement, write the three-address code. (10)  
X := A - B + C - D + E - F
- Convert the three-address code into machine code.
- 20 a) How do algebraic laws help in optimizing basic blocks? (4)
- b) Write the code generation algorithm. (6)

\*\*\*\*