Question Paper

Exam Date & Time: 05-Jun-2018 (09:30 AM - 12:30 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

INTERNATIONAL CENTRE FOR APPLIED SCIENCES
IV SEMESTER B.S. DEGREE MAKE - UP EXAMINATION - MAY / JUNE 2018
DATE: 5 JUNE 2018
TIME 9.30AM TO 12.30 PM

Language Processors [CS 244]

Marks: 100 **Duration: 180 mins.**

Answer any FIVE FULL questions.

1)

A) Explain the input buffer schemes for scanning the source program. How the use of sentinels can improve its performance? Describe in detail with necessary algorithms. (10)

B) With an example for each, differentiate between the following terms related to Lexical Analysis: (6)
i) token
ii) pattern
iii) lexeme

C) List and Explain any FOUR compiler construction tools. (4)

2)

A) Given the regular expression xy(x|y)* (10)
i) Construct an NFA using Thompson's construction Algorithm.
ii) Convert the NFA into DFA using Subset Construction Algorithm.
iii) Show the transition table of DFA

B) What are the different error recovery strategies employed by a parser? Explain. (6)

C) Eliminate left recursion from the following grammar: (4)

S → A|B|Sc|dS
A → Bd|cA|f
B → Se|Ad|g

3)

A) Give the steps for computation of FIRST and FOLLOW for a non-terminal of a grammar. Find the FIRST and FOLLOW sets for all the non-terminals in the following grammar, clearly indicating the Rule applied for every terminal added. (8)

S → ABCd

A → e|Cf|ε

B → g|h|ε

C → p|q

B) Give the algorithm for Non Recursive Predictive Parsing. (6)

C) Write the recursive descent parser for the following grammar. The terminals are specified in boldface. (6)

S → if E then S|if E then S else S|a|b

E → x|y

4) A) Give the algorithm for construction of SLR parsing table. (6)

B) Construct the SLR parsing table for the grammar given below, showing all the intermediate steps. (10)

S → L=R

S → R

L → *R

L → id

R → L

C) Given the grammar, (4)

S → aABe   A → b   A → Abc   B → d

differentiate between the working of top down and bottom-up parser in parsing the string "abbcde".

5) A) Why does a compiler translate a source program to an intermediate code? (4)

B) Explain briefly the different ways of implementing a three address statement. Illustrate considering the following expression by first converting it into three address statements. (8)

X = a+b*c-d/(b*c)

C) Give the LR parsing algorithm along with input and output specifications. (8)

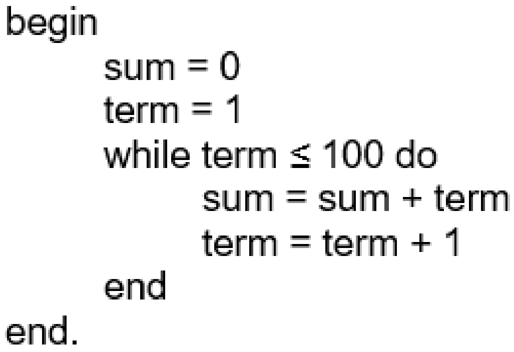
6) A) Discuss the two strategies used for dynamic storage allocation. Explain the various fields of general Activation record. (10)

B) Explain different type constructors used in type expressions, with an example for each. (10)

7) What are forward references? Explain how they are resolved in Single Pass Assembler with suitable example. (8)

A)

B) Write down the intermediate code for the following code fragment, and show the basic blocks and draw the flow graph. (12)



```
begin
        sum = 0
        term = 1
        while term ≤ 100 do
                sum = sum + term
                term = term + 1
        end
end.
```

8) A) What is DAG for expressions? Obtain DAG representation for the following basic block: (8)

```
 1) t1 = 4*i
 2) t2 = a[t1]
 3) t3 = 4*i
 4) t4 =b[t3]
 5) t5 = t2 * t4
 6) t6 = prod + t5
 7) prod = t6
 8) t7 = i + 1
 9) i = t7
10) if i<= 20 goto (1)
```

B) Give the algorithm for determining the liveness and Next-use information for each statement in a basic block. Also, compute the liveness and Next-Use of following basic block: (12)

```
x = y + z
z = x * 5
y = z - 7
x = z + y
```

-----End-----