CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

Seventh Semester of B. Tech. (IT) Examination
Nov 2019
IT443 Language Processors

Date: 08.11.2019, Friday   **Time: 01:30 p.m. To 04:30 p.m.**   **Maximum Marks: 70**

Instructions:
1. The question paper comprises of two sections.
2. Section I and II must be attempted in separate answer sheets.
3. Make suitable assumptions and draw neat figures wherever required.
4. Use of scientific calculator is allowed.

SECTION – I

Q - 1   Answer the following questions.   **[07]**

1)   Define : (i) lexeme (ii) semantic gap   [02]

2)   Explain single pass and multi pass compiler.   [02]

3)   What is macro? Write nested macro to calculate cube of a given number.   [03]

Q - 2.a   What is language processor? Discuss language processing activities in detail.   **[04]**

Q - 2.b   Answer any TWO questions.   **[10]**

1)   Draw a deterministic finite state automaton that accepts all the strings consisting of the symbols 'a' and 'b' that have an odd number of 'a's and an even number of 'b's.

2)   Consider given DFA of which q0 is the start state and q1, q2, q4 are the accepting states. Find minimize DFA and write regular expression.
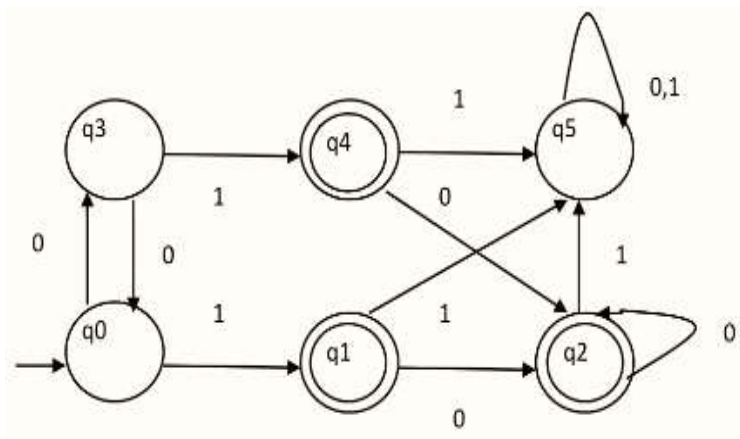
3)   Consider following C code.

```
x = x * 0;
for (i=1; i<5; i++)
{   x = x +1;
    y = 10;
}
for (i=1; i<5; i++)
{   y = y * 2;
}
```
Apply all possible code optimization techniques on above code and write final optimize code.

Q - 3   **Answer any TWO questions.**   **[14]**

1) Why lexical analyzer is separated than syntax analyzer phase? Discuss the role of lexical analyzer in compilation process with diagram.

2) What is an Assembler? Discuss the functionality of assembler. Explain two phases of assembler in detail.

3) Construct NFA-^ for following regular expression and convert it into DFA using Thompson's Construction. Draw final DFA with transition table.

<center>**(a/b)*abb**</center>

<center>**SECTION – II**</center>

Q - 4   **Do as directed.**   **[07]**

1) Identify the type of the grammar for following examples.   **[01]**

    1)   G: {S → aS | bA , A → aA | a}
    2)   G: {S → aSb | ab }

2) Write Three Address Code for following code.   **[02]**

if (a < b)
  x = y + z;
else
  p = q + r;

3) Perform the Left factoring of following grammar.   **[02]**

A → abB | aB | cdg | cdeB | cdfB

4) Eliminate the left recursion from the following grammar.   **[02]**

$S → Aa | b$

$A → Ac \ | Sd | \in$

Q - 5.a Consider following grammar.   **[04]**

S → SS+ | SS* | a

  1) Construct the parse tree and Show that the string aa+a* can be generated by the grammar.

  2) Is the grammar ambiguous for any string?

Q - 5.b **Attempt any TWO questions.**   **[10]**

1) Which are the different actions in shift-reduce parsing techniques? Describe the different types of conflicts in shift-reduce parsing.

2) Check the following grammar is LL(1) or not.

    S → iEtSS' | a
    S' → eS | ε
    E → b

3) Generate Canonical LR(0) parsing states for following grammar:

    S → aSb | ab

Q - 6.a What is symbol table? Which are different data structures to implement symbol table? **[02]**

Q - 6.b **Attempt any THREE questions.** **[12]**

1) Find out FIRST( ) and FOLLOW( ) set for following grammar.

    S → A
    A → aB / Ad
    B → b
    C → g

2) Construct a operator precedence graph and encode the following precedence table using precedence function f() and g().

	id	+	*	$
id		⋗	⋗	⋗
+	⋖	⋗	⋖	⋗
*	⋖	⋗	⋗	⋗
$	⋖	⋖	⋖	

3) Write a Syntax Directed Definitions for a desk calculator program and generate annotated parse tree for input: 3 * 4 + 5.

4) Differentiate between Canonical LR (CLR) and Look ahead LR (LALR) of Bottom-up parsing techniques.

****