Final Assessment Test – July 2023

VIT®
Vellore Institute of Technology

Course: **BCSE307L - Compiler Design**

Class NBR(s): **0891/0893/0895/0897/0899/0901/0903/ 0905/0907/0909/0911/0913/0917/0921/0923/0925/ 0927/0929/0931/0933/0937/0939/0941/0942/1255/ 1363**

Slot: **C2+TC2**

Time: **Three Hours**

Max. Marks: **100**

KEEPING MOBILE PHONE/SMART WATCH, EVEN IN "OFF" POSITION IS TREATED AS EXAM MALPRACTICE

Answer ALL Questions

(10 X 10 = 100 Marks)

1. Convert the regular expression (a|b)*cc(d|c)* to a DFA using the direct method with syntax tree. Provide a step-by-step explanation of the process, including the construction of the syntax tree, assignment of firstpos and lastpos, computation of followpos, and the construction of the DFA states and transitions.

   Finally, test the DFA with the input strings "aabccd" and "acdd" to determine if they are accepted or rejected by the DFA.

2. a) Explain the phases of a compiler for the given expression "if (x > 5) { y = x + 2; } else { y = x - 2; }." Provide a brief description of each phase and its role in the compilation process. **[7]**

   b) Write a lex program to count the number of vowels in a given input string. **[3]**

3. Consider the following grammar:

   S -> AB
   A -> aAb | ε
   B -> bB | c

   a) Compute the First and Follow sets for each non-terminal symbol. **[3]**
   b) Construct the LL(1) parsing table for the grammar. **[5]**
   c) Use the LL(1) parsing table to parse the input string "aabc" and show the step-by-step derivation. **[2]**

4. Consider the following grammar:

   S' –> S
   S –> aAc | bCc | aCd | bAd
   A –> e
   C –> e

   Check whether the grammar is LALR(1) or not ?

   Parsing Table ?

5. a) Consider the following grammar: [5]

E -> E + T | T
T -> T * F | F
F -> (E) | id

Perform syntax-directed translation to generate three-address code for arithmetic expressions. Show the annotated parse tree and the resulting three-address code for the input string "2 + 3 * 4".

b) Consider the following expression: [5]
E = (A && B) || (C || D)

Assume short-circuit evaluation for logical AND (&&) and logical OR (||) operators. Implement a semantic analysis algorithm to evaluate the expression using short-circuit evaluation. Show the step-by-step evaluation process and the final result for the following variable assignments:

A = true
B = false
C = true
D = false

6. a) Translate the logical expression [7]
"A > B AND (C < D OR E != F)" into three-address statements using backpatching.

b) Construct the syntax tree for the expression [3]
'X * (Y - 5) + (Z / W)' using appropriate nodes and edges.

7. Consider the following code snippet:

```
#include <stdio.h>

int main() {
  int x, y, z;

  x = 10;
  y = 5;

  if (x > y) {
    z = x + y;
    printf("The sum of x and y is %d\n", z);
  } else {
    z = x - y;
    printf("The difference between x and y is %d\n", z);
  }

  return 0;
}
```

i. Identify the basic blocks in the given code.
ii. Construct the Control Flow Graph (CFG) for the code snippet, indicating the basic blocks and control flow edges.

8. a) Consider the following code snippet: [5]

```
L1: LOAD R1, [R2]
    LOAD R3, [R4]
    MUL R1, R3
    ADD R1, R5
    STORE R1, [R6]
    JMP L1
```

Perform peephole optimization on the given code snippet and provide the optimized version.

b) How can loop optimization techniques, such as loop unrolling, improve the performance of programs in compiler design? Can you provide an example and explain the process of loop unrolling in detail? [5]

9. a) What are the main challenges in designing a code generator for a compiler? How does the code generator handle issues like target architecture compatibility, code quality, and efficient register allocation? [5]

b) Consider the following code snippet: [5]

```
int main() {
    int a = 5;
    int b = 10;
    int c = a + b;
    int d = c * 2;
    int e = d / 3;
    return e;
}
```

perform register allocation and assignment for the given code snippet assuming there are 3 available registers: R1, R2, and R3. Assume that variables are assigned registers in the order they appear in the code.

10. a) What is parallelization in the context of compiler design, and what are the different types of parallelization techniques used in optimizing programs? [4]

b) Discuss the different techniques used to achieve instruction level parallelism and provide examples illustrating their impact on program execution. [6]

⇔⇔⇔