National Institute of Technology Mizoram

End – Semester Examination, Even Semester – (2022-2023)

Principles of Compiler Design (CSL 1601)

B.Tech. 6th Semester **Full Marks: 50 marks** **Duration: 2:30 hours**

Answer all 5 (Five) Questions. All Questions carry same Marks

(5*10 = 50 Marks)

Question 1

(a) Differentiate between static and dynamic semantics of a language with examples. [2]

(b) What are the issues/challenges while designing a code generator. What are the different components of runtime storage. [2]

(c) What is compile time evaluation. What are the methods to perform compile time evaluation. [2]

(d) If you have 5 source languages and 2 target machines. How many code optimizers and code generations would you require if (i) you don't use intermediate code (ii) you use intermediate code. [2]

(e) What is a basic block. What are the two kinds of local (within a basic block) transformations/optimizations. [2]

Question 2

(a) Differentiate between L-attributed grammar and S-attributed grammar. What type of traversals can be used for evaluating: (i) synthesized attributes, (ii) inherited attributes. [3]

(b) For the given grammar below, draw the annotated parse tree for the given expression *2+3*5*. Also draw the dependency graph. [7]

Productions	*Semantic rules*
E→E_1+T	E.val := E_1.val+T.val
E→T	E.val := T.val
T→T_1*F	T.val := T_1.val * F.val
T→F	T.val := F.val
F→id	F.val := num.lexval

Question 3

(a) Differentiate between name and structural equivalence with examples. [3]

(b) Construct the syntax tree for the given arithmetic expression: [4]

(A+B)*(C−D)+((E/F)*(A+B))

(c) Write the postfix expression for the given infix expression: [3]

((H*((((A+((B+C)*D))*F)*G)*E))+J)

Question 4

(a) Generate the three-address code for the following [5]

```
while (a < c and b > d) do
        if a = 1 then c = c + 1
        else
                while a <= d
                        do a = a + b
```

(b) Write the three-address code and quadruple representation for: [5]
s = -z/a * (x + y)

Question 5

(a) Identify the basic blocks and draw the flow graph for the code below. [5]

```
→ 1) i = 1
  2) j = 1
  3) t1 = 10 * i
  4) t2 = t1 + j
  5) t3 = 8 * t2
  6) t4 = t3 - 88
  7) a[t4] = 0.0
  8) j = j + 1
  9) if j <= 10 goto (3)
  10) i = i + 1
  11) if i <= 10 goto (2)
  12) i = 1
  13) t5 = i - 1
  14) t6 = 88 * t5
  15) a[t6] = 1.0
  16) i = i + 1
  17) if i <= 10 goto (13)
```

(b) Write the DAG for the intermediate code below. Also, write the final assembly code. Assume there are only two registers R0, R1 in the target machine. [5]

```
a = b + c
b = a – d
c = b + c
d = a - d
```

National Institute of Technology Mizoram

Mid – Semester Examination, Even Semester – (2022-2023)

Principles of Compiler Design (CSL 1601)

B.Tech. 6th Semester **Full Marks: 30 marks** **Duration: 1 hour 30 mins**

Answer all 3 (Three) Questions. All Questions carry same Marks
(3 * 10 = 30 Marks)

Question 1

(a) What are the different types of languages and how are they represented? [1]

(b) Checking the types of variables, evaluation of mathematical expressions, etc. are performed in which phase of a compilation process? [1]

(c) Loop unrolling, inline functions, etc. are performed in which phase of a compilation process? [1]

(d) What is bootstrapping with respect to a compiler? [1]

(e) Differentiate between lexeme and token with an example? [1]

(f) Given that binary number strings are read from left to right and may have leading zeroes, construct a DFA (Deterministic Finite Automaton) for: [5]

- Binary number strings that represent numbers that are multiples of 4, e.g., 0, 100, 1000, 1100, and so on.
- Clearly specify the input alphabet, set of states, start state, set of final states, transition function with transition diagram

Question 2

(a) The following grammar is not suitable for a top-down predictive parser (e.g., LL(1) parser). [3+7]

i. Identify the problem (e.g., 'left recursion' or 'need left factoring') and correct it by rewriting the grammar. (A non-terminal symbol may be added in the corrected grammar)
L -> Ra | Qba
R -> abaS | cabaS
S -> bcS | ∈
Q -> bbc | bc

ii. Compute the FIRST and FOLLOW sets and also dirsymb of the 'corrected grammar' from (i). Show that your new grammar satisfies the LL(1) condition based on dirsymb.

Question 3

(a) Construct the DFA and SLR parsing table for the following grammar. [5+5]
S -> Aa | bAc | Bc | bBa
A -> d
B -> d
The Follow sets for the non-terminals are given below:
Follow(S) = {$}, Follow(A) = {a,c}, Follow(B) = {a,c}
Determine the type of conflict if there is any based on the parsing table constructed.

dr.