

COMPILER DESIGN
(CSEN 4037)

Time Allotted : 2½ hrs

Full Marks : 60

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 4 (four) from Group B to E, taking one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A

1. Answer any twelve: **12 × 1 = 12**

Choose the correct alternative for the following

- (i) G is a regular grammar: $S \rightarrow aS \mid bS \mid a \mid b$. The language produced by the regular grammar G is
(a) a^*b^* (b) aa^*bb^* (c) $(a+b)^*$ (d) $(a + b)(a + b)^*$
- (ii) In some programming language, L denotes the set of letters and D denotes the set of digits. The expression that defines an identifier is
(a) $(LD)^*$ (b) $L(L + D)^*$ (c) $(L + D)^*$ (d) $L(LD)^*$
- (iii) Which Phase of the compiler is machine-dependent?
(a) Syntax Analysis (b) Semantic Analysis
(c) Code generation (d) None of these.
- (iv) The grammar: $S \rightarrow Sa1 \mid Sa2 \mid b1 \mid b2$
(a) is left recursive (b) has common left factor(s)
(c) (a) & (b) both (d) none of these
- (v) Which data structure is used to manage information about variables and their attributes?
(a) Abstract syntax tree (b) Symbol table
(c) Semantic stack (d) Parse table.
- (vi) In a bottom-up evaluation of a syntax directed definition, inherited attributes can
(a) always evaluated
(b) be evaluated only if the definition is L-attributed
(c) be evaluated only if the definition has synthesized attributes
(d) never be evaluated.
- (vii) Consider the SDT, $A \rightarrow BC \{A.i = B.i\}$ is
(a) S-attributed (b) L-attributed (c) both (d) none
- (viii) Statement starting from a leader up to the next leader is called
(a) Leader (b) DAG (c) Basic block (d) None.

- (ix) One of the purposes of using intermediate code in compilers is to
- make parsing and semantic analysis simpler
 - improve error recovery and error reporting
 - increase the chances of reusing the machine-independent code optimizer in other compilers
 - improve the register allocation.
- (x) Reduction in strength means
- replacing run time computation by compile time computation
 - removing loop invariant computation
 - removing common sub expression
 - replacing a costly operation by a relatively cheaper one.

Fill in the blanks with the correct word

- (xi) The postfix representation for the infix expression: $x + -y * (-y + z)$ is ____.
- (xii) If x is a terminal, then $FIRST\{x\}$ is ____.
- (xiii) YACC builds up ____ parsing table.
- (xiv) Consider the grammar: $S \rightarrow ABc, A \rightarrow a|\epsilon, B \rightarrow b|\epsilon$. $FOLLOW(A)$ is ____.
- (xv) In a compiler, keywords of a language are recognized during ____.

Group - B

2. (a) What will be the regular expression of a Deterministic Finite State Automata (DFA) that accepts all strings that ends with **ab**, over the I/P symbol: $\{a, b\}$.
[[CSEN4037.1](Apply/IOCQ)]
- (b) Design a DFA (M) over the I/P symbol: $\{a, b\}$ that accepts all strings that **does not end with ab**?
[[CSEN4037.1](Apply/IOCQ)]
- (c) Differentiate tokens, patterns, lexeme with example. [[CSEN4037.3, .4](Remember/LOCQ)]
4 + 4 + 4 = 12
3. (a) Give the NFA for the Regular expression: $(a | b)^* abb$, and then find the equivalent DFA for it.
[[CSEN4037.1](Apply/IOCQ)]
- (b) What are the analysis phase and synthesis phase of a compiler?
[[CSEN4037.2](Remember/LOCQ)]
- (c) Eliminate left recursion and left factor from the following grammar:
 $E \rightarrow aba | abba | Ea | EbE$
[[CSEN4037.1, .3](Apply/LOCQ)]
6 + 2 + 4 = 12

Group - C

4. (a) Consider the following grammar:
 $S \rightarrow Ab | b$
 $A \rightarrow Ac | Sd | \epsilon$
 Find $FIRST$ and $FOLLOW$ sets of each non-terminal of the grammar.
[[CSEN4037.4, .5](Apply/IOCQ)]
- (b) Construct the $LL(1)$ parsing table for the grammar.
[[CSEN4037.4, .5](Apply/IOCQ)]
6 + 6 = 12

5. (a) Consider the context free grammar (CFG) given below:

$$F \rightarrow + F \mid - F \mid \text{num}$$
 Here, F is a non-terminal (and, obviously, the start symbol). The terminals are unchanged: +, - and num. Give the LR(0) parsing table for the above CFG.
 How will the sentence “ - - 9” will be parsed? [[CSEN4037.4, .5)(Apply/IOCQ]]
- (b) Why is an LR(k+1) parser called more powerful than an LR(k) parser?
[[CSEN4037.4, .5)(Analyze/IOCQ]]
(6 + 4) + 2 = 12

Group - D

6. (a) Construct the DAG for the following statement :

$$A = (x + y) * (x - y) + x / (x - z) + (x + y)$$
 [[CSEN4037.4)(understand/LOCQ]]
- (b) Define L-attributed definition with example. [[CSEN4037.3)(understand/LOCQ]]
- (c) A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar

$$S \rightarrow xxW \{ \text{print}("1"); \}$$

$$S \rightarrow y \{ \text{print}("2"); \}$$

$$W \rightarrow Sz \{ \text{print}("3"); \}$$

 Then what will be the translation of “xxxxyyzz”? [[CSEN4037.4, .5)(Apply/IOCQ]]
5 + 3 + 4 = 12

7. (a) Consider the following syntax-directed definition over the grammar defined below:

$$S \rightarrow A \text{ Sign } \{ S.\text{val} = A.\text{val}; A.\text{sign} = \text{Sign.sign}; \text{print}(A.\text{val}); \}$$

$$\text{Sign} \rightarrow + \{ \text{Sign.sign} = 1 \}$$

$$\text{Sign} \rightarrow - \{ \text{Sign.sign} = 0 \}$$

$$A \rightarrow n \quad \{ A.\text{val} = \text{value}(n) \}$$

$$A \rightarrow A1 , n \{ A1.\text{sign} = A.\text{sign};$$

$$\quad \text{if}(A.\text{sign} = 1) \text{ then}$$

$$\quad \quad A.\text{val} = \min (A1.\text{val}, \text{value}(n));$$

$$\quad \text{else}$$

$$\quad \quad A.\text{val} = \max(A1.\text{val}, \text{value}(n)); \}$$

 Give an attributed parse tree for the source string “5,2,3-”and evaluate the attributes in the attributed parse tree depicting the order in which the attributes need to be evaluated (if more than one order is possible indicate one.)
[[CSEN4037.3, .4)(Apply/LOCQ]]
- (b) Define direct and indirect triples with suitable examples. [[CSEN4037.3, .4)(Remember/LOCQ]]
8 + (2 + 2) = 12

Group - E

8. (a) Consider the following C code segment
- ```

i=0;
j=0;
while (i<n && j<n)
{
 x += 4*j + a[i];
 y -= a[i] - 4*j;
 i++;
 j++;
}

```

What types of optimization scopes are present in the code? Incorporate the associated optimization techniques and rewrite the code. *[[CSEN4037.4, .6)(Apply/IOCQ)]*

- (b) Translate the following code it into machine code and show the register and address descriptors while the instructions are generated. Assume that two registers are available.

```
X:=B-C
Y:=A-B
B:=B+C
A:=Y-D
```

*[[CSEN4037.4, .6)(Apply/IOCQ)]*  
**6 + 6 = 12**

9. (a) Consider the following code to find out the basic blocks and design the flow graph representing the code segment.

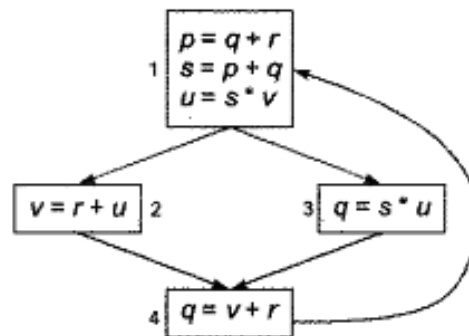
```
1. f = 1;
2. i = 2;
3. if (i > x) goto (8)
4. f = f * i;
5. t1 = i + 1;
6. i = t1;
7. goto (3)
8. goto calling program
```

*[[CSEN4037.3, .4, .6)(Apply/IOCQ)]*

- (b) A variable x is said to be live at a statement  $S_i$  in a program if the following three conditions hold simultaneously:

- (i)  $\exists$  a statement  $S_j$  that uses x
- (ii) There is a path from  $S_i$  to  $S_j$  in the flow graph of the corresponding programme
- (iii) The path has no intervening assignment to x including at  $S_i$  and  $S_j$

Consider the flow graph given below:



State the variables which are live both at the statement in basic block 2 and 3. (Do not forget to explain)

*[[CSEN4037.4, .6)(understand/IOCQ)]*

**6 + 6 = 12**

---

| Cognition Level         | LOCQ | IOCQ | HOCQ |
|-------------------------|------|------|------|
| Percentage distribution | 35   | 65   | 0    |