COMPILER DESIGN
(CSEN 4111)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

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

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) YACC builds up
(a) SLR Parsing table (b) LALR Parsing table
(c) Canonical LR Parsing table (d) None of the above.

(ii) In some programming language, L denotes the set of letters and D denotes the set of digits. An identifier is permitted to be a letter followed by any number of letters or digits. The expression that defines an identifier is
(a) (L.D)* (b) (L + D)* (c) L(L.D)* (d) L(L + D)*

(iii) The language produced by the regular grammar S→ aS|bS|a|b is
(a) a*b* (b) aa*bb* (c) (a+b)* (d) (a+b)(a+b)*

(iv) Which one of the following statement is true?
(a) Canonical LR parser is more powerful than LALR parser
(b) SLR parser is more powerful than LALR
(c) LALR parser is more powerful than canonical LR parser
(d) SLR parser, canonical LR parser and LALR parser all have the same power.

(v) The number of tokens in the following C statement is
printf("i = %d, &i = %x", i, &i);
(a) 3 (b) 26 (c) 10 (d) 21

(vi) 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.

(vii) 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.

(viii) A compiler that runs on one machine and produces code for a different machine is called
(a) cross compiler (b) one pass compiler
(c) 2 pass compiler (d) none of these

(ix) Consider the SDT, A → BC {A.i = B.i} is _____
(a) S-attributed (b) L-attributed (c) both (d) none of (a), (b) & (c).

(x) Reduction in strength means
(a) replacing run time computation by compile time computation
(b) removing loop invariant computation
(c) removing common sub expression
(d) replacing a costly operation by a relatively cheaper one.

Group- B

2. (a) Construct a DFA directly (not by generating NFA) for the regular expression (a | b) * ab.
[(CO4)(Understand/HOCQ)]

(b) List the tokens (with 'type' and 'value') for:

```
int max (int x, int y)
{
        return (x > y? x : y);
}
```

[(CO2)(Understand/LOCQ)]
8 + 4 = 12

3. (a) Explain the different phases of a compiler, showing the output of each phase, using the example of the following statement:

```
for(i=0;i<10;i++)
        a=a+10;
```

[(CO2)(Understand/IOCQ)]

(b) A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2, and T_3 over the alphabet {a, b, c}.
T_1: a?(b|c)*a T_2: b?(a|c)*b T_3: c?(b|a)*c
Note that 'x?' means 0 or 1 occurrence of the symbol x. Note also that the analyzer outputs the token that matches the longest possible prefix.
If the string *bbaacabc* is processed by the analyzer, which sequence of tokens it outputs?
[(CO3)(Understand/IOCQ)]

(c) List the various error recovery strategies for a lexical analyser. [(CO4)(Remember/LOCQ)]
8 + 2 + 2 = 12

Group - C

4. (a) Consider the CFG below:

$$S \rightarrow CC$$
$$C \rightarrow cC \mid d$$

Is the grammar LL(1)as well as SLR(1)? Give reasons for your claim.
[(CO2, CO3)(Understand, Analyze/IOCQ)]

(b) An LALR(1) parser for a CFG (G)can have Shift-Reduce conflict if and only if the LR(1) parser for the same CFG (G)would suffer from Shift-Reduce conflict. Is this claim correct? Give reasons for your answer.
[(CO2, CO3)(Understand, Analyze/IOCQ)]

(c) Create a predictive parser table for the given CFG:

$$S \rightarrow FR$$
$$R \rightarrow *S \mid \in$$
$$F \rightarrow id.$$

[(CO2, C03)(Create/IOCQ)]
4 + 4 + 4 = 12

5. (a)

$E \rightarrow TE'$
$E' \rightarrow +E \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow T \mid \varepsilon$
$F \rightarrow PF'$
$F' \rightarrow *F' \mid \varepsilon$
$P \rightarrow (E) \mid a \mid b \mid \varepsilon$

Construct the FIRST and FOLLOW set for all non-terminals. Construct the predictive parsing table for the above grammar. [(CO4)(Analyze/HOCQ)]

(b) $S \rightarrow xxW$ {print"1"}
$S \rightarrow y$ {print"2"}
$W \rightarrow Sz$ {print"3"}
A shift-reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar. Using the translation scheme described by the above rule, find out the translation of xxxxyzz among the following. Explain your answer.
(i) 11231 (ii) 11233 (iii) 23131 (iv) 22331. [(CO2)(Understand/IOCQ)]
(4 + 4) + 4 = 12

Group - D

6. (a) Define synthesized attribute and inherited attribute for syntax directed translation. [(CO4)(Remember/LOCQ)]

(b) Generate the three address code for the below program fragment:

```
sum = 0;
for( i = 1; i <= 20; i++)
        sum = sum + a[i] + b[i];
```

[(CO3, CO4)(Apply, Analyze/IOCQ)]

(c) What is the significance of flow graph? What are the contents of Activation Record? [(CO1)(Remember/LOCQ)]

2 + 6 + (2 + 2) = 12

7. (a) Translate the expression a = -(a + b) * (c + d) + (a + b + c) into
   (i) Quadruple
   (ii) Triple
   (iii) Indirect Triple. [(CO4)(Understand, Analyze/IOCQ)]

(b) Divide the following code into basic blocks and draw the flow graph.
   (i) f =1;
   (ii) i = 2;
   (iii) if (i > x) goto (viii)
   (iv) f = f * i;
   (v) t = i+1;
   (vi) i = t;
   (vii) goto (iii)
   (viii) goto calling program. [(CO3)(Understand/IOCQ)]

(2 + 2 + 2) + 6 = 12

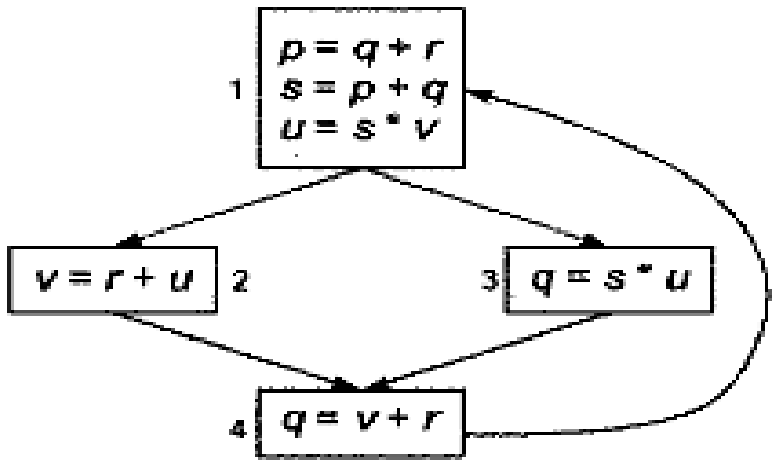
Group - E

8. (a) Consider the following C code segment for code optimization:

```
for(i  = 0; i < n; i++)
{
for(j = 0; j < n; j++)
{
        if(i % 2)
        {
                x += (4 * j + 5 * i);
                y += (7 + 4 * j);
        }
}
}
```

Show all intermediate steps of optimization and finally, present the optimized code [(CO5, CO6)(Remember, 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) ∃ 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 following flow graph:



State the variables which are live at the statements in basic block 2 and 3. [(CO5) (Evaluate/HOCQ)]

8 + 4 = 12

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

D : = B - C
E : = A - B
B : = B + C
A : = E – D. [(CO1)(Analyze/HOCQ)]

(b) Build a control-flow graph for the code given below:

(i) i = 1
(ii) j = 1
(iii) t1 = 5 * i
(iv) t2 = t1 + j
(v) t3 = 4 * t2
(vi) t4 = t3
(vii) a[t4] = -1
(viii) j = j + 1
(ix) if (j <= 5) go to iii)
(x) i = i + 1
(xi) if (i < 5) go to ii) [(CO5, CO6)(Apply, Analyze/IOCQ)]

8 + 4 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	12.5	58.33	29.16

Course Outcome (CO):

After the completion of the course students will be able to
CO1: This course will enable a student to understand the major phases of compilation including the front- and backend. They are expected to have an overview of how a real life compiler works.
CO2: After completion of this course, the students are expected to develop knowledge of Lex and YAAC tools.
CO3: The students should be able to understand various necessary tasks related to compiler construction, like token identification, grammar writing, type conversion, and storage management.
CO4: Students will learn to generate intermediate codes and actual machine codes targeting a particular architecture.
CO5: Students should acquire a detailed idea regarding optimization of generated code across various phases of the compilation process.
CO6: After completion of this course, students should be able to apply various optimization techniques for dataflow analysis.

*LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question; HOCQ: Higher Order Cognitive Question