COMPILER DESIGN
(INFO 3133)

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) The grammar $S \rightarrow aSa \mid bS \mid c$ is
(a) LL(1) but not LR(1) (b) LR(1)but not LR(1)
(c) Both LL(1)and LR(1) (d) Neither LL(1)nor LR(1)

(ii) The number of tokens in the following C statement
printf("i=%d, &i=%x", i, &i);
(a) 3 (b) 21 (c) 26 (d) 18.

(iii) A bottom up parser generates_____.
(a) right most derivation (b) right most derivation in reverse
(c) left most derivation (d) left most derivation in reverse

(iv) Given below are the regular expressions
(i) (a/b)* (ii) (a*b*)* (iii) (ab)*
which of them are equivalent ?
(a) (i) only (b) (i) and (ii) only
(c) (i), (ii) and (iii) only (d) (ii) and (iii) only.

(v) Which of the following is the most powerful parser?
(a) SLR (b) Canonical LR
(c) Operator Precedence (d) LALR

(vi) Consider the following grammar
$S \rightarrow FR$
$R \rightarrow S \mid \varepsilon$
$F \rightarrow id$
In the predictive parser table, M, of the grammar the entries M[S, id] and M[R, $] respectively
(a) $\{S \rightarrow FR\}$ and $\{R \rightarrow \varepsilon\}$ (b) $\{S \rightarrow FR\}$ and $\{\}$
(c) $\{S \rightarrow FR\}$ and $\{R \rightarrow *S\}$ (d) $\{F \rightarrow id\}$ and $\{R \rightarrow \varepsilon\}$.

(vii) An intermediate code form is_____.
(a) Postfix notation (b) Syntax trees
(c) Three address code (d) All of these

(viii) If a NFA has n states then the maximum number of states the equivalent DFA can have
(a) n (b) n^2
(c) 2^n (d) none of these.

(ix) The lexical analyzer takes____as input and produces a stream of_____as output.
(a) Source program, tokens (b) Token, source program
(c) Either A and B (d) None of the above.

(x) A pictorial representation of the value computed by each statement in the basic block is
(a) Tree (b) DAG
(c) Graph (d) All of these.

Group – B

2. (a) Define assembler and cross compiler.

(b) Write the differences between analysis and synthesis phase of compiler.

(c) What is regular definition? Write regular definition for the tokens if, then, else, relop and identifier.

(2 + 2) + 2 + (1 + 5) = 12

3. (a) Define Deterministic Finite Automata (DFA) and Non-deterministic Finite Automata (NFA). Write a Regular Expression (RE) for the set of strings whose 2nd bit from right end is 1 and 4th bit from right end is 0 and design its corresponding DFA.

(b) Define different types of Lexemes with example.

(2 + 2 + 2 + 3) + 3 = 12

Group – C

4. Construct an LL(1) parsing table for the following Grammar:
S→aBDh
B→cC
C→bC|ϵ
D→EF
E→ g|ϵ
F→ f|ϵ

12

5. (a) Identify the token stream for the following statement that will be the o/p of the lexical analyzer. Based on the token stream draw a parse tree.
while(M>N && M<= 2*N-5) do M:=M+N;

(b) Consider the following grammar,
G=({S,B}, {a,b,c,d},{S→cAd, A→b|a}, S), Show the steps of action taken by Predictive Parser Program(PPP) for the sentence, w=cad, The predictive parsing table is given bellow:

	a	b	c	d	$
S			S→cAd		
A	A→a	A→b			

(2 + 4) + 6 = 12

Group – D

6. Consider the following code fragment , convert it to 3-address code then represent the 3-address code in Quadruples, Triples and indirect Triples:

```
x:=y+z+k;
x:=1
  if x<10 then
     x:=y+z-k;
 else
x:= y*z*k;
```

(3 + 3 + 3 + 3) = 12

7. What do you mean by code optimization? Consider the productions and its corresponding semantic rules in the following table, derive the sentence $7*9+3_n$ (n for new line) then draw the corresponding annotated parse tree.

	production	Semantic rules
1	$L \rightarrow E_n$	Print(E.val)
2	$E \rightarrow E+T$	E.val= E.val+ T.val
3	$E \rightarrow T$	E.val= T.val
4	$T \rightarrow T*F$	T.val= T.val* F.val
5	$T \rightarrow F$	T.val=F.val
6	$F \rightarrow (E)$	F.val=E.val
7	$F \rightarrow digit$	F.val=digit.lexval

(2 + 4 + 6) = 12

Group – E

8. (a) Draw DAG then optimize for the following code fragment:
   1. P1:=4*I
   2. P2:=addr(A)-4
   3. P3:=P2[P1]
   4. P4:=4*I
   5. Addr(B)-4

6. P6:= P5[P4]
7. P7:= P3*P6
8. P8:=PROD + P7
9. PROD:=P8
10. P9:=I+1
11. I:=P9
12. if I<=20 goto (1)

(b) Explain Loop Unrolling with example ?

(6 + 3) + 3 = 12

9. Write Short Note (any three)
   (i) Yacc
   (ii) Peephole optimization
   (iii) Lex
   (iv) Reducible flow graph
   (v) Uniform Symbol Table.

(4 + 4 + 4) = 12

Department & Section	Submission Link
IT	https://classroom.google.com/c/MjQyMTE4NjcwNjM2/a/Mjc0NjY1MjkxNDI5/details