Course Code : CST 412 **KOLP/RW – 19 / 9155**

Seventh Semester B. E. (Computer Science and Engineering) Examination

LANGUAGE PROCESSORS

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) All questions carry marks as indicated against them.

(2) Assume suitable data and illustrate answers with neat sketches wherever necessary.

1. (a) There is a new language L which has to be made available on machine A. Give the steps along with T-diagrams to show how a compiler can be built that accepts L, produces output in A and is implemented in A. 2 (CO 1)

(b) Construct the DFA corresponding to the regular expression "(a|b)*b". 4 (CO 1)

(c) Explain the significance of symbol table in the lexical analysis phase. 2 (CO 1)

(d) Construct the transition diagram to recognize some of the given keywords of JAVA: case, catch, char, class, const, continue, final, finally, float, for, this, throw, throws, transient. 2 (CO 1)

OR

(e) Write the regular expression and construct the transition diagram for :

(i) Integer digits

(ii) Floating points numbers. 2 (CO 1)

2. (a) Consider the given grammar,

$$X \longrightarrow iXYj \mid jY$$
$$Y \longrightarrow kY \mid mX \mid Z$$
$$Z \longrightarrow Zn \mid n$$

Compute the SLR parsing table and state whether the grammar is SLR or not. 6 (CO 2)

OR

(b) Find FIRST, FOLLOW and construct the LL(1) parsing table for the given grammer :

$A \longrightarrow Bi|h$
$B \longrightarrow ejA \mid Ck \mid \varepsilon$
$C \longrightarrow m \mid n \mid Ap \mid \varepsilon$

Also show whether the string "ejhpkii" is valid or invalid. 6 (CO 2)

(c) Show whether the given grammar is CLR or not by constructing CLR parsing table :

$S \longrightarrow AA$
$A \longrightarrow aA|b$

Also construct the LALR parsing table and state whether any conflicts occur. 4 (CO 2)

3. (a) Construct the annotated parse tree and find the three - address code for the given construct.

```
for(i = 1;   i < 50;   i + 1)
  if (i < 10)   then
           a = b + c
   else
           a = c + 1
```

6 (CO 3)

OR

(b) Consider the array assignment statement, $A[i, j] = B[i, j] + C[k]$. Given the dimensions of array A and B as 10×10. The dimension C is 10 and bpw = 2. Construct the parse tree and find the three-address code. 6 (CO 3)

(c) Find the three-address code for the given Boolean expression. not(a<b and (e>f or i<j)). 4 (CO 3)

4. (a) Write the C code for merge sort. Draw the activation tree when numbers 5 8 1 9 4 2 7 3 are to be sorted. Also show the intermediate control stacks having the activation records. 3 (CO 1)

(b) Construct the LR parsing table for the given grammar and generate the error routines to handle errors.

$$A \longrightarrow A + A \mid A*A \mid a$$

Show the error recovery for the string "a+*a". 5 (CO 1)

(c) Illustrate the mark and sweep garbage collector with example. 2 (CO 1)

OR

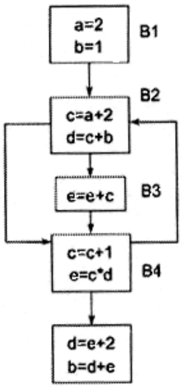
(d) Illustrate various data structures that can be used to implement symbol table. 2 (CO 1)

5. (a) Construct the DAG and eliminate the common sub-expression :

```
t1 = 4*i
n = a[t1]
t2 = 4*i
t3 = 4*j
t4 = a[t3]
a[t2] = t4
t5 = 4*j
a[t5] = n.
```

2 (CO 4)

(b) Find IN, OUT, GEN and KILL for the given CFG.



Compute the UD-chain for statement $c = a + 2$. 5 (CO 4)

(c) Explain various loop optimization techniques. 3 (CO 4)

OR

(d) Generate the program flow graph for the following :—

```
int  func(int  t)
 {
          intx,y;
           x=t;
           y=t;
          while(a>100   and   t>0)
           {
                    y=y*x+t;
                     x=x+1;
           }
 }
```

3 (CO 4)

6. (a) Consider the given three-address code :

```
t = a-b
u = c*d
v = e+u
w = t+v
```

Generate code for the following computation orders using simple code generation algorithm :

(i) t-u-v-w (ii) u-v-t-w

to prove the statement :
"Reordering the three-address statements for code generation changes the cost of computation". Comment on the cost of computation and state which algorithm can be used to find the optimal order of computation. 5 (CO 4)

(b) Generate the target code using gencode() procedure for the expression a=b - (c*d) + e/(f+g). 5 (CO 4)

OR

(c) Apply dynamic programing to generate the code. Compute the cost vector at each node of the DAG for the given three address code :

T1 = a + b

T2 = c*T1 5 (CO 4)