7. (a) Consider the following SDT. If an LR parser caries out the translations on an input string “xxyyzzwy”, what is the output? Explain.
A→xxA {print(“x”);} | yB {print(“y”);} | y {print(“z”);}
B→yzB {print(“w”);} | zwA {print(“e”);}

(b)
```
E → E1 #T      {E.value = E1.value * T.value}
| T            {E.value = T.value}
T → T1 & F     {T.value = T1.value + F.value}
|F             {T.value= F.value}
F →num         {F.value = num.value}
```
Compute E.value for the root of the parse tree for the expression: 12 # 3 & 5 # 6 & 10. Show each step of the evaluation process.

(c) Construct a DAG and write the sequence of instructions for the expression
Z = X – Y + X * Y * U – V / W + X + V

4 + 4 + 4 = 12

Group – E

8. (a) Explain the terms with example:
(i) Peephole optimization
(ii) Activation record

(b) 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.
(i) x = y ∗ z
(ii) w = x + y
(iii) y = y ∗ w

(3 + 3 = 6) +6 = 12

9. (a) Explain the following techniques of code optimization with suitable examples:
i) Copy propagation
ii) Dead code elimination
iii) Common sub-expression elimination

(b) Consider the following program code:
```
Prod = 0;
I = 1;
Do
{
      Prod=prod + a[i]*b[i];
      I = i + 1;
}while (i <= 10);
```
Construct the flow graph for the above mentioned program code.

(3 * 2 = 6) + 6=12

B.TECH/CSE /7TH SEM/ CSEN 4101/2019

COMPILER CONSTRUCTION
(CSEN 4101)

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

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and <u>any 5 (five)</u> from Group B to E, taking <u>at least one</u> 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) Which languages necessarily need heap allocation in the runtime environment?
(a) Those that support recursion
(b) Those that use dynamic scoping
(c) Those that allow dynamic data structures
(d) Those that use global variables.

(ii) Assume that SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between n_1 and n_2 is
(a) $n_1 > n_2$ (b) $n_1 = n_2$ (c) $n_1 > n_2$ (d) None of the (a),(b),(c).

(iii) The number of tokens in the following C statement is
printf(“x=%d, &x=%x”, x, &x);
(a) 13 (b) 6 (c) 10 (d) 9.

(iv) Which of the following concepts are not loop optimization techniques?
(a) Jamming (b) Unrolling
(c) Induction variable elimination (d) None of the (a),(b),(c)..

(v) Consider the grammar
S→ ABSc | Abc
BA →AB
Bb→ bb
Ab →ab
Aa →aa
Which of the following sentences can be derived by this grammar?
(a) abc (b) aab (c) abcc (d) abbc

(vi) Consider the grammar
E → E + E | id
then FOLLOW(E) will contain
(a) {$} (b) {+} (c) {$, +} (d) {$, +, id}.

(vii) Which of the following statement is false
(a) Every left recursive grammar can be LR(1)
(b) Every right recursive grammar can be LR(1)
(c) Every left recursive grammar can be LL(1)
(d) Every right recursive grammar can be LL(1).

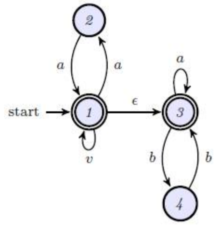
(viii) The graph that shows basic blocks and their successor relationship is called
(a) Dag (b) Flow Graph
(b) Control Graph (d) Hamilton Graph.

(ix) YACC builds up
(a) SLR Parsing table (b) LALR Parsing table
(c) Canonical LR Parsing table (d) None of the (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) What will be the regular expression of the given NFA



(b) Is look ahead essential for lexical analyser? Explain your opinion.
(c) State the differences between compiler and interpreter.

7+ (3 + 2) = 12

3. (a) Show the output of each phase of compiler on the following input.

```
A= B*C;
if (A>10)
    X=1;
else
     X=0;
```

(b) Consider the following statements and find the number of tokens with type and value as applicable:

```
int min(int x, int y)
{
    /*find max*/
    return (x < y ? x : y);
 }
```

9 + 3 = 12

Group – C

4. (a) S→SS + | SS * | a
(i) Show how the string aa + a* can be generated by this grammar.
(ii) What language does this grammar generate? Justify your answer.

(b) E→E + T T→T * F F→ (E) | J J → a | b | c
Remove left recursion from the above production rules.

(c) Show that the following grammar:
S→AaAb | BbBa
A→ε
B→ε
Is LL(1) but not SLR(1).

(2+2)+ 3+ 5= 12

5. (a) Consider the following grammar and construct the SLR parsing table
S → L=R | R
L → *R | id
R → L

(b) Then show the steps how the input string “*id” will be parsed.

8 + 4 =12

Group – D

6. (a) Define direct and indirect triples with suitable examples.

(b) Consider the following program code:

```
While (i < 10)
{
    if (i % 2 == 0)
          even = even + i;
    else
          odd = odd + i;
}
```

Generate the intermediate code for the above code segment.

4 + 8 = 12