B.Tech Even Semester Examination, May, 2025

Computer Science & Engineering

(6th Semester)

Course No: CSE-601

(Compiler Design)

Full Marks: 50

Pass Marks: 25

Time: 2 hours

Note:

1. Attempt any five questions.
2. Begin each answer in a new page
3. Answer parts of a question at a place
4. Assume reasonable data wherever required
5. The figure in the right margin indicates full marks for the question
6. All the mathematical symbols and abbreviations have their usual meanings.

1. What are difference between LR(0) and LR(1) items. Write the algorithm for: (i) Closure (ii) Goto (iii) Item sets of an LR(1) items. [3+2+2+3]

2. Explain the model of **non-recursive predictive parsing** and write its **parsing algorithm** [5+5]

3. Write the syntax-directed definition (SDD) for the following grammar. Also, construct the annotated parse tree for the input string 3 * 5 + 4 using **bottom-up parsing**.

   Given grammar rules:

   i. $E \rightarrow E + T$

   ii. $E \rightarrow T$

   iii. $T \rightarrow T * F$

iv. T→F

v. F→(E)

vi. F→digit [5+5]

4. What is an inherited attribute? Show that the following grammar is an inherited-attributed grammar by taking an example.

D→TL

T→int

T→real

L→L1,id

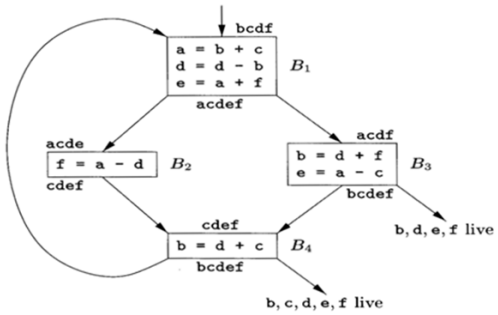
L→id. [2+8]

5. What is an activation record? Explain its structure and describe how it is used in managing function calls during program execution, providing an example. [2+4+4]

6. What is intermediate code representation? Explain three different forms of three-address code representations by taking the example

a = b * -c + b * -c . [2+8]

7. Write the algorithm for computing IN and OUT sets. Then, compute the GEN and KILL sets for the graph given in Fig-1. [5+5]



8. Given the following code segment in Fig-2, construct the flow graph. Then, perform copy propagation on the code. [5+5]

```
(1)  i := m-1
(2)  j := n
(3)  t1 := 4*n
(4)  v := a[t1]
(5)  i := i+1
(6)  t2 := 4*i
(7)  t3 := a[t2]
(8)  if t3 < v goto (5)
(9)  j := j-1
(10) t4 := 4*j
(11) t5 := a[t4]
(12) if t5 > v goto (9)
(13) if i >= j goto (23)
(14) t6 := 4*i
(15) x := a[t6]
(16) t7 := 4*i
(17) t8 := 4*j
(18) t9 := a[t8]
(19) a[t7] := t9
(20) t10 := 4*j
(21) a[t10] := x
(22) goto (5)
(23) t11 := 4*i
(24) x := a[t11]
(25) t12 := 4*i
(26) t13 := 4*n
(27) t14 := a[t13]
(28) a[t12] := t14
(29) t15 := 4*n
(30) a[t15] := x
```

******