B.Tech Third Year End Semester Examination

Department: Computer Science and Engineering

Course Name: Compilers **Code: CS 346**

Full Marks-100 **Time: 3 hours**

Answer any FIVE Questions

Make reasonable assumptions as and whenever necessary. You can answer the questions in any sequence. However, answers of all the components to any particular question should appear together.

1 (a). Consider the following grammar (*type*: Synthesized attribute, *in*: Inherited attribute):

Production	**Semantic Rules**
D→TL	L.in=T.type
T→int	T.type=integer
T→real	T.type=real
$L \rightarrow L_1$ id	L_1.in =L.in, addtype (id.entry, L.in)
L→id	addtype(id.entry, L.in)

Construct the parse tree, annotated parse tree and the dependency graph for the expression: int p q r

(b). Consider the following S-attributed definition to implement a very simple desk calculator.

Production	Semantic Rules
(1) L → E \n	val[ntop] := val[top-1]
(2) $E \rightarrow E_1 + t$	val[ntop] := val[top-2] + val[top]
(3) E → T	
(4) $T \rightarrow T_1 * F$	val[ntop] := val[top-2] * val[top]
(5) T → F	
(6) F → (E)	val[ntop] := val[top-1]
(7) F → digit	

Show the sequence of moves (i.e., the states on the stack, the values on the stack, the remaining input, productions used and the corresponding semantic rules) that result while parsing the input "(8 + 2) * 4 \n" in a bottom-up fashion. **8+12**

2. (a). Translate the following assignments into quadruples and triples.
(i). a=b[i] + c[j]

(ii). x=f(y+1) +2

(b). Distinguish between local and global optimization? What is peephole optimization? Consider the following sequence of 3-address statements.

(a). prod := 0, (b). i := 1, (c). t1 := 4 * i, (d). t2 := a[t1], (e). t3 := 4 * i , (f). if t3>1 goto (c), (g). t4 := b[t3], (h). t5 := t2 * t4, (i). t6 := prod + t5, (i). prod := t6, (k). t7 := i + 1, (l). i := t7, (m). if i <= 20 goto (e)

Determine various basic blocks in the above segment of code. Draw the control flow graph.

8+ (3+2+4+3)

3. (a). Translate the following expressions by avoiding redundant GOTO statements (or, Jump statements)

(i). If (x<100 || x>200 && x!=y) x=0

(ii). If (a==b && c==d && e==f) x=1

(b). Why instruction selection is important in code generation? Define the terms "register allocation" and "register assignment". Generate code for the following three-address statements.

(i). x=a[i], y=b[j], a[i]=y, b[j]=x (assume that a and b are arrays whose elements are 4-byte values)

(ii). s=0, i=0, L1: if i>n goto L2, s=s+1, i=i+1, goto L1, L2: (assume that n is in memory location)

(c). Write short notes on "*Loop invariant code motion*" and "*copy propagation*".

(3*2) + (4*2) + (3*2)

4. (a). Define backpatching technique and mention its advantages. Design a syntax directed translation scheme for backpatching (use **attributes**: *truelist*, *falselist*, *instr* etc. and **functions**: *makelist*, *merge*, *backpatch* etc.) the following boolean expressions during bottom-up evaluation.

$E \rightarrow E_1$ or E_2

| E_1 and E_2

| not E_1

| (E_1)

| id_1 relop id_2

| true

| false

(b). Consider the translation scheme of (b) and show the various steps of backpatch process for the expression: x<100 || x>200 && x!=y. Assume that the address of the first instruction is 100.

(c). Draw the annotated parse tree for (c) and show the values of true and false lists at the various nodes.

8+8+4

5. (a). Mention the various steps of *reaching definition algorithm* with respect to global optimization. Consider the following basic blocks:

B1= {d1: i = m – 1, d2: j = n, d3: a = u1}, **B2**={d4: i = i + 1, d5: j = j – 1}, **B3**={d6: a = u2}, **B4**={d7: i = u2}.

The corresponding CFG has edges among the following pairs: B1→B2; B2→B3, B4; B3→B4; B4→B2

Compute **GEN** and **KILL** sets for each basic block. Use this to determine **IN** and **OUT** sets.

(b). Determine the cost of the following instruction sequences.

```
LD R0, c

LD R1, i

MUL R1, 8

ST a(R1), R0
```

(5+3+10)+2

6. (a). Consider the following set of basic blocks:

B1={d1:a=1, d2: b=2}, **B2**={d3:c=a+b, d4:d=c-a}, **B3**={d5:d=b+d}, **B4**={d6:d=a+b, B7:e=e+1}, **B5**={d8:b=a+b, d9:e=c-a}, **B6**: {d10:a=b*d, d11:b=a-d}

The CFG has the following set of edges: B1→B2; B2→B3; B3→B4, B5; B4→B3; B5→B6, B2.

Compute the **DEF**, **USE**, **IN** and **OUT** sets for live variable analysis.

(b). Mention the various steps of register allocation algorithm using graph coloring. Use this algorithm to generate the target code of the following set of instructions assuming only 2 registers are available.

a := b + c, t1 := a * a , b := t1 + a , c := t1 * b, t2 := c + b , a := t2 + t2

10+3+7