CS3300: Final Exam: Nov 19 2025 (A)

Total/Maximum marks = 100, Time: 3.00 hrs

Name: _____ Roll: _____

1. **Lexical Analysis** [6 marks] Consider the transition diagram for recognizing tokens for the following language,
   - The local variables can be a single capital letter [A-Z] only.
   - Global variables can be two capital letters only
   - The function names can only be two chars: first one is a capital letter, and second one is a digit [0-9].
   - Integer literals can be any single digit.
   - Labels can be two characters only, capital letter followed by lower case letter [a-z]

   Answer the question given below:
   Number of states = _____?,
   number of final states = _____?

2. **Parsing**
   [24 marks] Consider the grammar on the right. For the non-terminals S, E, E', T, T' and F,
   (a) number of elements in the FIRST set are __, __, __, __, __, and __, respectively. [3 marks]
   (b) number of elements in the FOLLOW set are __, __, __, __, __, __, respectively. [3 marks]
   (c) In the LL(1) predictive parsing table, the number of non-error entries = ____, number of error entries = ____ [12 + 1 marks]
   (d) If you augment the above grammar, then the number of items in the first LR(0) item (I_0) = ____.[5 marks]

$$
\begin{array}{lcl}
S & \rightarrow & E \\
E & \rightarrow & T\ E' \\
E' & \rightarrow & \texttt{OR}\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & \texttt{AND}\ F\ T' \mid \epsilon \\
F & \rightarrow & \texttt{NOT}\ F \mid \texttt{true} \mid \texttt{false} \mid (E)
\end{array}
$$

3. **Type checking**
   [6 marks] Consider the snippet of MiniJava code on the right. For the class to type check,
   T0 = _____,
   T1 = _____,
   T2 = _____,
   T3 = _____,
   T4 = _____,
   T5 = _____,
   T6 = _____,
   T7 = _____,
   T8 = _____,
   T9 = _____,
   T10= _____,
   T11= _____. [12x0.5 marks]

```
class ctr {
    public T9 s;
    public T10 c;
    public T11 t;

    public T0 ad(T1 a, T2 b) {
        c = iP(a);
        t = a + b;
        return t; }
    public T3 iP(T4 d) {
        return d > 0; }
    public T5 ct(T6 f, T7 e) {
        T8 r;
        if (f) {
            r = this.s + e;
        } else {
            r = ad(e, 1); }
        return r; }
}
```

4. **Semantic Analysis**

   Consider the grammar on the right, that accepts a subset of strings containing as and bs. Complete the partially filled SDD rules to check if the input string is a palindrome. [7 marks]
   What type of attribute is isPal? _____ [1 mark]

$S \rightarrow a\ S_1\ a$:	S.isPal = _____
$S \rightarrow a\ S_1\ b$:	S.isPal = _____
$S \rightarrow b\ S_1\ b$:	S.isPal = _____
$S \rightarrow b\ S_1\ a$:	S.isPal = _____
$S \rightarrow a$:	S.isPal = _____
$S \rightarrow b$:	S.isPal = _____
$S \rightarrow \varepsilon$:	S.isPal = _____

5. **Control Flow Analysis**
   Consider the three address code shown on the right. Assume S1, S2 etc are simple statements.
   Answer the questions for the code shown on the right [5 x 2 marks]:
   (a) Number of leaders in the code? ____.
   (b) Number of loops in the code? _____.
   (c) How many back-edges are there in the code? _____
   (d) The leader statement(s) of the natural loop-header(s) is/are _____
   (e) Number of basic blocks dominated by the block whose leader is 6? _____

```
1.  S0
2.  L1:  S1
3.  if cond1 goto L3
4.  L2:  S2
5.  if cond2 goto L4
6.  L3: S3
7.  if cond3 goto L2
8.  L4: S4
9.  if cond4 goto L1
10. S5
```

6. **Three address codes**

Consider the following C code. Fill in the blanks (on the right) with the correct 3-address code [10 marks].

```
int a[10], b, c, d;
b = 5;
c = a[b] + 3 * d;
if (c > 10) {
    a[b + 1] = c - d;
} else {
    a[b - 1] = c + d;
}
```

```
b = 5
t2 = b * _____
t3 = a[t2]
t4 = _________
c = t3 + t4
t5 = c > 10
t6 = _______
if t6 goto ______
t7 = b + 1
t8 = ______
t9 = c - d
a[t8] = t9
goto ______
L1:
t10 = _____
t11 = _______
t12 = _____
__________ = t12
L2:
```

7. **Liveness Analysis and Register Allocation**

[12 marks = 4x2 + 4] Consider the code shown on the right side that uses eight variables. For each of the following variables, state at which statement(s) the variable is live (that is, present in the IN set). Ignore the statements that only have labels.

- x _____
- y _____
- z _____
- w _____

Using Kempe's heuristic, the number of registers required to do spill-free register allocation = _____.

```
1: x = a + b
2: y = x + c
3: if y > 10 goto L1
4: z = y + d
5: goto L2
L1:
6: z = y - d
L2:
7: w = z + 1
8: return w
```

8. **Optimizations**

[12 marks] Consider the C code shown on the right side. Focus only on the body of the loop: The list of optimizations (and the number of times) that can be applied on this code are _____

```
foo(int n, int a[]){
  int i, j, k;
  int x, y, z;
  i = 0;
  j = 0;
  while (i < n) {
      x = a[i] * 4;
      y = a[i] + 4;
      z = (x + y) - 0;
      k = z * 1;
      a[j] = k;
      i = i + 1;
      if (n > z) j = j + 1;
  }
}
```

9. **Instruction Scheduling**

[10 + 1 + 1 marks] Consider the assembly code shown. Assumptions:

- Load latency: 2 cycles;
- Add, Sub, Store latency: 1 cycle;
- Multiply latency: 3 cycles;
- Instructions dependent on loads must wait until data is ready.

Give a schedule for this code, such that it leads to minimum number of pipeline stalls. (use the original instruction numbers to specify the sequence)

_____

_____.

In the original schedule, number of pipeline stalls = ____

In the best schedule, the number of pipeline stalls = ____

```
1:  load r1, 0(r2)  // Load from memory to r1
2:  load r3, 4(r2)  // Load from memory to r3
3:  add r4, r1, r3  // r4 = r1 + r3
4:  load r5, 8(r2)  // Load from memory to r5
5:  load r8, 12(r2) // Load from memory to r8
6:  mul r6, r4, r5  // r6 = r4 * r5
7:  add r7, r8, r3  // r7 = r8 + r3
8:  sub r9, r6, r1  // r9 = r6 - r1
9:  store 0(r2), r7 // Store r7 at 0(r2)
10: store 4(r2), r9 // Store r9 at 4(r2)
```