CS6235 Quz 1 Exam: Feb 16 2023

Maximum marks = 35, Time: 50 min

Name: _____ Roll: _____

- **Negative marking**. Each incorrect True/False answer will lead to a deduction of 0.5 mark.
- Advise: work out each question succinctly and legibly.
- A question with X marks will approximately take X minutes. Plan accordingly.

---

1. **Reaching definitions.**
   [Marks: 1 + 2] Recall the reaching definitions analysis discussed in the class. Does it perform a may-analysis or must analysis? What changes will you make to convert it from *may to must* (if your answer to the previous question is may-analysis) or *must to may* analysis (if your answer to the previous question is must-analysis)?

2. **Happens Before and MHP**.
   [3 marks] Give an example code to illustrate HB relation. The code satisfy the following conditions:

   - The code should include two statements `S1` and `S2`.
   - Statically HB(`S1`,`S2`) and HB(`S2`,`S1`) both hold.
   - The code should have no loops/recursion/threads.

   [2 marks] Give another example code to illustrate HB relation. The code satisfy the following conditions:

   - The code should include two statements `S1` and `S2`.
   - `S1` and `S2` are in two different functions that may run in parallel with each other.
   - HB(`S1`, `S2`) and HB(`S2`, `S1`) holds.

3. **Happens Before and MHP**.
   [5 marks] Give an example code to illustrate MHP relation. The code should satisfy the following conditions:

   - The code should include two statements `S1` and `S2`.
   - MHP(`S1`,`S2`) holds.
   - `S1` and `S2` are in the `run` methods of two different classes (`T1` and `T2`), each of which extend the Java `Thread` class.
   - No method invokes more than one method.

4. **Constant Propagation.**
   [3 marks] Consider the following C code:

```
foo() {
 ... // some code not shown.
 Sx: x = 5;
 Sy: if (y > 0) goto L1; // L1 not shown
 Sz: y = x + y;
 ...
}
```

   Give the flow functions for the three statements shown to perform flow-sensitive constant propagation.

5. **Lattices and monotonic functions**
   [3 marks] Give a function $f : L \to L$, which satisfies the following conditions:

   - L is the bit-vector lattice.
   - f is non-monotonic.

   Prove the non-monotonicity property of f.

6. **Control Flow Graphs**
   [Marks: 1.5 + 1.5] Write a Java code which can lead to a CFG with the following conditions:

   - the CFG contains at least one node having 4 successors, and
   - no node in the CFG should have more than two predecessors.

   Draw the CFG for the code shown.

7. **Points to analysis**
   [5 marks] We have studied how to perform flow-sensitive intra-procedural points-to analysis in the class. In that context, consider a statement of the form `a = b.foo(c)`. Give the transfer function to process this statement. Note: since we are doing intra-procedural analysis, we will not know the details of the function `foo`.

8. **Points to analysis**
   [3 marks] Give an example code to illustrate flow sensitive points-to analysis. The code should satisfy the following conditions:

   - The code includes a store statement.
   - We cannot perform strong update for the store statement.

Section 2. True/False (1 mark each)

_____ Each deadlocked program must have started with a data-race.

_____ A program with data-races will eventually get deadlocked.

_____ Data flow analysis cannot be used to precisely tell if any arbitrary program (that takes no user-input) will terminate.

_____ Amdahl's law says that max speedup due to parallelism does not depend on the parallel fraction of the code.

_____ Java threads share the stack space.

_____ A definition can reach itself.

_____ A Java program with no static fields and not creating any mutable object, will not have any data-race.

_____ Each definition, that is not the last statement of a function, must reach at least one more statement.

_____ A thread object in Java cannot access the static fields of other thread objects.

_____ In general, flow insensitive analysis will use less memory.