CS6235 Quz 2 Exam: Apr 17 2021

Maximum marks = 20, Time: 1.00 hr

Name: _____ Roll: _____

- **Write your roll number on every sheet of the answer book.**
- Each question is for four marks.
- There are two sections in the question paper.
  - Descriptive type: Answer any four out of the five questions.
  - True or False: Each incorrect True/False answer will lead to a deduction of 0.5 mark.
- Advise: work out each question separately and legibly.

Section 1. Descriptive type

1. Consider the following snippet of Java code.

```
class A{
 public static void main(String []a){
  S1;
  Thread t1 = new X();
  Thread t2 = new Y();
  t1.start();
  t2.start();

  t1.join();
  t2.join();
 }
}
```

```
class X extends Thread {
  public void run(){
    S2;
    S3;
  }
}
class Y extends Thread {
  public void run(){
    S4;
  }
}
```

   (a) Insert additional code in the program such that S1 and S2 may run in parallel. That is, MHP(S1, S2) = *true*. [2]

   (b) Insert additional code in the program such that S3 and S4 may NOT run in parallel. That is, MHP(S3, S4) = *false*. [2]

2. (a) Consider a compiler for Java programs. Assume that we have populated a map `MHP`, such that `MHP(S)` returns the set of statements that may run in parallel with `S`. Similarly, we have populated two maps ρ and σ, such that for any variable v, $\rho(v)$ returns the set of abstract objects v may point to, and $\sigma(O_1, f)$ returns the set of objects pointed to by field f of the abstract object O_1. Use these maps to detect if there is data race between two statements `S1` and `S2`. [2]

   (b) Say, we have populated a map `MHP`, such that `MHP(S1, S2)` returns *true*, if `S1` and `S2` may run in parallel, and *false* otherwise. Use this map to compute a map `M`, such that if `S` is a statement in the program then `M(S)` should return the set of statement that may run in parallel with `S`, [2]

3. Draw the PEG for the following Java code. Clearly state how you obtained the notify edges (if any) using the iterative analysis discussed in the class [4].

```
class Main {
 public static void main(String [] args)
 {
  Buffer buffer = new Buffer();
  Reader r1 = new Reader(buffer);
  r1.start();
  r1.join(); } }

class Writer extends Thread {
Buffer buffer;
public Writer(Buffer b){ buffer = b; }
 public void run() {
  synchronized (buffer)
  {
    buffer.write();
    buffer.notify();
  } } }
```

```
class Reader extends Thread {
Buffer buffer;
public Reader(Buffer b){ buffer = b; }
 public void run() {
  Writer w1;
  w1 = new Writer(buffer);
  w1.start();
  w1.join();
  synchronized (buffer)
  {
    buffer.wait();
    buffer.read();
  }
 }
}
```

4. Consider the problem of inter-procedural constant propagation. Write an example code that shows the imprecision resulting from using CHA for call-graph resolution. [4]

5. Consider a Java program that is written in such a way such that each local variable is written to exactly once. Does performing flow-sensitive analysis (in contrast to performing flow-insensitive analysis) improve the precision of points-to analysis for such a program? If your answer is "No", then give an argument supporting it. If your answer is "yes" - give an example supporting it. [4]

Section 2. True/False (1 mark each)

_____ The size of an PEG is $O(N)$, where N is the program size.

_____ A PSG cannot be used represent a program with recursive functions.

_____ In the set of instructions given for three-address code in the class, the unconditonal branch (`goto L`) is redundant and can be avoided.

_____ Context sensitive analysis always leads to more precise results than context insensitive analysis