CS6235 Final Exam: May 13 2021

Maximum marks = 33, Time: 2.00 hr, Open Book, Closed Neighbor

Name: ______________________________ Roll: ______________

- **Write your roll number on every sheet of the answer book.**
- The marks of each question is specified next to the question.
- There are two sections in the question paper.
  - Section 1: Descriptive type. Try to write to the point.
  - Section 2: 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. **Memory consistency model.** [5]

   (a) [2] In the following code, how is it possible have `r1 == r2 == r3 == 0`? Suggest how it can be avoided.

   `x = y = 0`

Thread 1	Thread 2
1. `r1 = x`	4. `x = 1`
2. `y = 1`	5. `r3 = y`
3. `r2 = x`	

   `r1 == r2 == r3 == 0`. Possible?

   (b) [3] Find below an execution trace. Each line described a different process. The horizontal axis represents the global time line; for instance, as per the global clock, the operation `R(x)2` in process 2 executed after `W(x)2` in process 1. There are three different options for the process 4. Using each of these options we get three different sets of traces (for example, {1-3, 4i} or {1-3, 4ii}, and {1-3, 4iii}). For each of these traces, indicate the strongest memory consistency model (among Eventual, weak, causal, PRAM, strict and sequential consistency) that satisfies the trace. If none of them satisfy a trace, then mention 'none'.

   ```
      1.   W(x)1 W(x)2 sync
      2.               R(x)2 W(x)5 sync
      3.               W(x)6 sync
     4i.                     R(x)2 R(x)1 sync
    4ii.               R(x)1 R(x)5 R(x)6
   4iii.               sync  R(x)1 R(x)5
   ```

2. **Serial program analysis.** [5]

   (a) Show an example program that shows the interdependence of call-graph construction and points-to analysis. [3]

   (b) Why the Killdall's algorithm cannot be used to identify constants on the heap? That is, to identify constant values of the fields of objects. [2]

3. **Data Race.** [5]

Consider the shown trace and answer the following questions: Does the program have a race? [1] Using CP algorithm studied in the class, show the CP ordering for the above trace and identify the race(s) in the program, if any. [4]

	T1	T2
1	`w(y)`	
2	`acq(l)`	
3	`w(x)`	
4	`rel(l)`	
5		`acq(l)`
6		`r(y)`
7		`r(x)`
8		`rel(l)`

4. **MHP Analysis** [8].
Give a scheme to perform MHP analysis for programs written in a language, where

- A program consists of a single function.
- A parallel task can be defined using the `async` keyword (like X10).
- A special statement `waitForAll` can be used by different tasks to wait for all the tasks (to execute `waitForAll` or terminate).
- Tasks cannot be nested.
- No conditional or loop statements can be written.
- All the tasks created within the function must terminate by the end of the function (you may assume an implicit `finish`).

```
S1
async{
  S21;
  waitForAll
  S22
}
async{
  S3
  }
S4
async{
  S51;
  waitForAll
  S52
}
S6
```

In this snippet:

(a) `S1` happens first and does not run in parallel with any other statement.

(b) `S21` may happen in parallel with `S3`, `S4`, `S51` and `S6`.

(c) `S22` may happen in parallel with `S52` and `S6`.

(d) `S3` may run in parallel with `S21`, `S4`, `S51` and `S6`.

(e) `S4` may run in parallel with `S21` and `S3`.

(f) `S51` may run in parallel with `S21`, `S3` and `S6`.

(g) `S52` may run in parallel with `S22` and `S6`.

5. **Parallel IR** [5].
   Draw the PST and PPG for the following X10 code [2+3].

```
int foo(int n){
  finish {
    x = n;
    async{
      y = x + 1;
      z = y + 2;
    }
    Loop {
      if (y > 3) {
        y = x + z;
        break;
      }
      async {
        z = y * z;
        y = i + z;
      }
      i++; y = y + i;
    }
  }
  return y + z;
}
```

6. **Impact of MHP Analysis.** [3].
   Consider inter-procedural flow-insensitive alias analysis of serial Java programs. Can the same analysis be used for parallel Java programs? Or it has to be extended with MHP information? If the serial analysis cannot be used as it is - shown an example supporting your argument. Otherwise, give an argument supporting your argument.

Section 2. True/False (1 mark each)

________ Each access an element in the symbol table organized as an ordered linear list requires $(n \log n)$ cost.

________ In the Liveness analysis, writing $in[n] = use[n] \cup out[n] - def[n]$ may lead to imprecision but not unsoundness.