COL 380 **March 17, 2016**

Homework 2

Instructor: Subodh Sharma **Due: March 24, 23:55 hrs**

NOTE: All submissions must be made in the pdf format. Hand written assignments will not be accepted.

Problem 1: Sequential Consistency, Linearizability

- For each of the histories shown in Figs 1a and 1b, are they Sequentially consistent? Linearizable? Justify your answer. (4 marks)

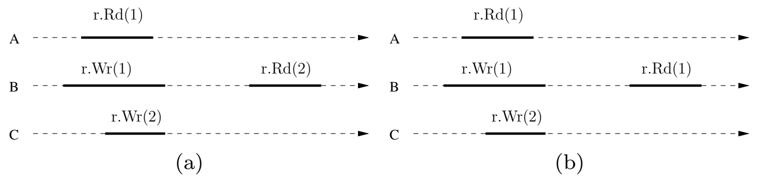

Figure 1: Traces

- Let *strict consistency* be defined in the following way: *Any read on a data item x returns a value corresponding to the result of the most recent write.* In class we had discussed the definitions of *sequential consistency* (SC) and *linearizability.* Assuming, we have a binary relation $\mathcal{W}$ that is irreflexive, antisymmetric and transitive which captures *Weaker-than* relationship among consistency models. Thus, if $(a, b) \in \mathcal{W}$ then model a is weaker than model b. Establish a $\mathcal{W}$ relationship among SC, strict consistency and linearizability. Justify your answer. (4 marks)
- A way to realize logical clocks (for establishing happens-before relation or causality among events that are necessary for SC or linearizability) is by either *Lamport clocks* or *Vector clocks.* Explain a cardinal difference between Lamport clocks and Vector clocks [Reference: Lamport clock video; Vector clock video]. Show at each event the associated vector clocks for a sequentially consistent execution history of example in Figure 1(a). (4 marks)

Problem 2: OpenMP

Consider the loop:

```
a[0] = 0;
for (i = 1; i < n; i ++)
    a[i] = a[i - 1] + i;
```

Is the loop parallelizable (with or with loop transformation)? If so, then write snippet of OpenMP code for the parallel version of this loop. (4 marks)

Problem 3: OpenMP Debug

Consider the following program. Identify correctness issues in the program and specify the solutions (SHOW ONLY RELEVANT CHANGES): the **for** loop must be parallelized on 2 threads collectively calling foo() 10 times and then each thread prints “Hello World” in an *ordered* fashion. (6 marks)

```
#pragma omp parallel
{
  omp_set_num_threads(2);
  #pragma omp parallel for
  for (int i = 0; i < 10; i++)
  {
    foo();
  }
  printf(``hello world from %d \n '', omp_get_thread_num());
}
```

Problem 4: OpenMP Performance

Find all performance issues with the program listed below and provide an alternate implementation addressing all the issues: (4 marks)

```
#pragma omp parallel for
 for ( i = 0 ; i < N ; ++i ) {
   #pragma omp critical
   {
     if (arr[i] > max) max = arr[i] ;
   }
 }
```