Assignment 2

ECSE 420: Parallel Computing

Due: November 9, 2017

Submission instructions: Students are to work in groups of two on the assignment. Students must also submit a pdf file that contains the following information: student's names, student ids, instructions on how to run each file and the associated question solved. Students are also expected to submit a zip file containing their source code. This code must compile and run without error. Code must be well formatted, commented, and follow the google java style guide. For more information, see the ECSE420AssignmentSubmissionInstructions.pdf in the Assignment section on myCourses.

Questions

1. (36 marks) This question will examine the concept of Locks.
   1.1. Implement the Filter lock described in Chapter 2 of the course text.
   1.2. Does the Filter lock allow some threads to overtake others an arbitrary number of times? Explain.
   1.3. Implement Lamport's Bakery lock also described in Chapter 2.
   1.4. Does the Bakery lock allow some threads to overtake others an arbitrary number of times? Explain
   1.5. Propose a test that verifies if a lock works, i.e., if it provides mutual exclusion.
   1.6. Provide an implementation for the proposed test and verify if the implemented locks do provide mutual exclusion.

2. (6 marks) Does Peterson's two-thread mutual exclusion algorithm work if the shared atomic flag registers are replaced by regular registers?

3. (12 marks) Programmers at the Flaky Computer Corporation designed the protocol shown in Fig. 1 to achieve n-thread mutual exclusion. For each question, either sketch a proof, or display an execution where it fails.
   3.1. Does this protocol satisfy mutual exclusion? (Hint: Start the proof by assuming that two threads A and B are in the critical section at the same time.)
   3.2. Is this protocol deadlock-free? (Hint: show an execution that generates a dead-lock)
   3.3. Is this protocol starvation-free?

```
1  class Flaky implements Lock {
2    private int turn;
3    private boolean busy = false;
4    public void lock() {
5      int me = ThreadID.get();
6      do {
7        do {
8          turn = me;
9        } while (busy);
10       busy = true;
11     } while (turn != me);
12   }
13   public void unlock() {
14     busy = false;
15   }
16 }
```

Fig.1 Flaky Lock used in exercise 2

4. (12 marks) For each of the histories shown in Figs. 2 and 3, are they sequentially consistent? Linearizable? Justify your answer.

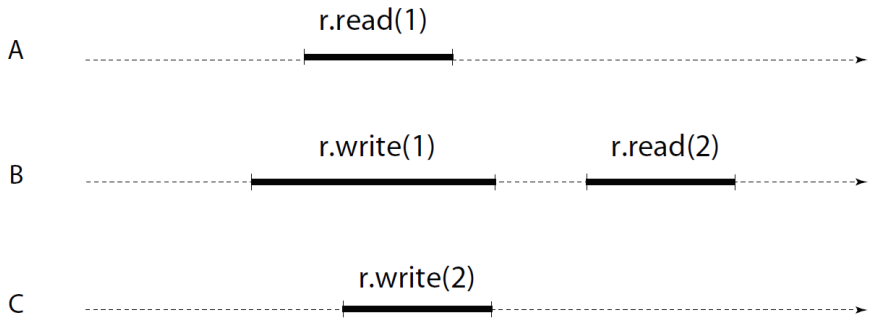

Fig. 2 History (a)

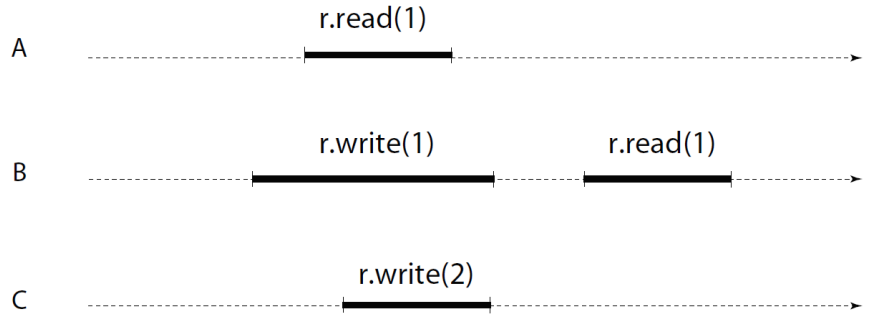

Fig. 3 History (b)

5. (6 marks) Consider the class shown in Fig. 4. According to what you have been told about the Java memory model, will the reader method ever divide by zero?

```
1  class VolatileExample {
2    int x = 0;
3    volatile boolean v = false;
4    public void writer() {
5      x = 42;
6      v = true;
7    }
8    public void reader() {
9      if (v == true) {
10       int y = 100/x;
11     }
12   }
13 }
```

Fig. 4 Volatile example

6. (12 marks) Consider the safe Boolean MRSW construction shown in Fig. 5 True or false:
   6.1. If we replace the safe Boolean SRSW register array with an array of safe M-valued SRSW registers, then the construction yields a safe M-valued MRSW register. Justify your answer.
   6.2. If we replace the safe Boolean SRSW register array with an array of regular Boolean SRSW registers, then the construction yields a regular Boolean MRSW register. Justify your answer.

```
1  public class SafeBooleanMRSWRegister implements Register<Boolean> {
2    boolean[] s_table; // array of safe SRSW registers
3    public SafeBooleanMRSWRegister(int capacity) {
4      s_table = new boolean[capacity];
5    }
6    public Boolean read() {
7      return s_table[ThreadID.get()];
8    }
9    public void write(Boolean x) {
10     for (int i = 0; i < s_table.length; i++)
11       s_table[i] = x;
12   }
13 }
```

Fig. 5 The SafeBoolMRSWRegister class: a safe Boolean MRSW register

7. (6 marks) Show that if binary consensus using atomic registers is impossible for two threads, then it is also impossible for n threads, where n > 2. (Hint: argue by reduction: if we had a protocol to solve binary consensus for n threads, then we can transform it into a two-thread protocol.)

8. (6 marks) Show that if binary consensus using atomic registers is impossible for n threads, then so is consensus over k values, where k > 2.

Total: 96 marks