Concurrent Algorithms October 1, 2024

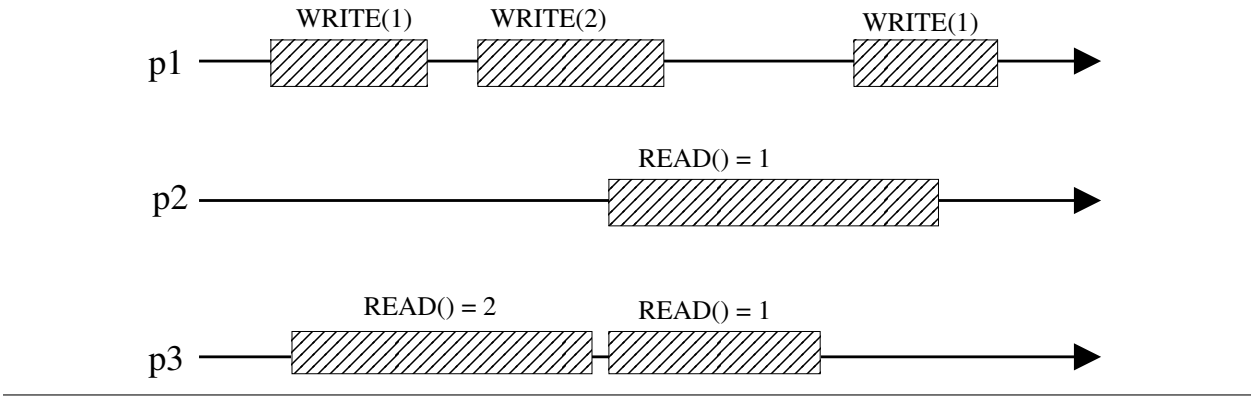
Exercise 1

Problem 1. Explain the difference between a regular register and an atomic register. Provide an example execution that is allowed for a regular register but not allowed for an atomic register.

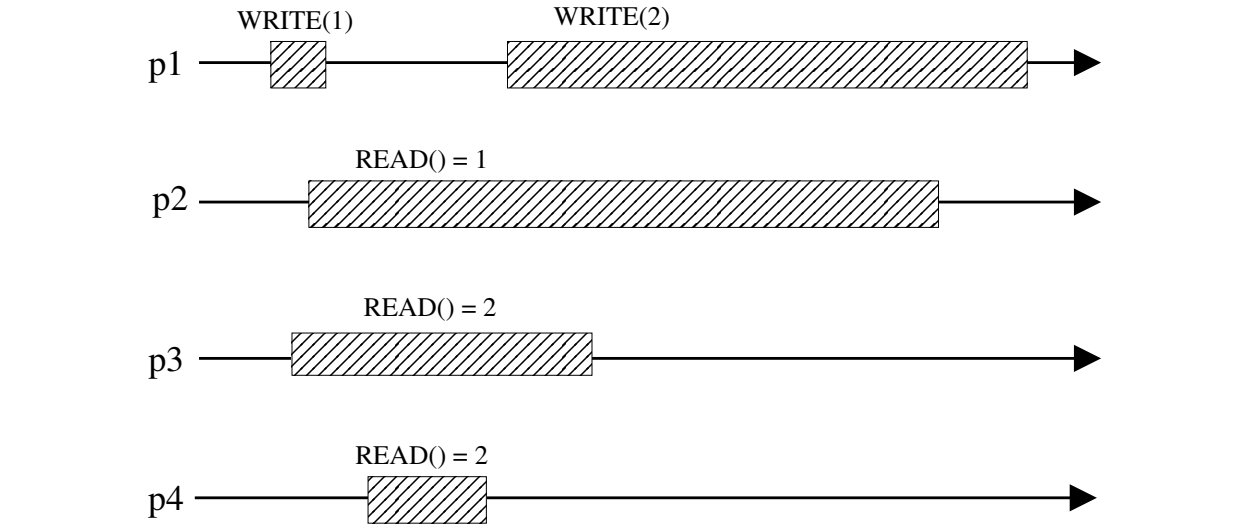
Problem 2. Each of the following executions represents a run of an algorithm that implements a read/write register. For each execution:

- Specify whether the execution is: *atomic*, *regular*, *safe*, or *none-of-the-above*. Explain why this is the case.
- If the execution is atomic, draw in the serialization points.

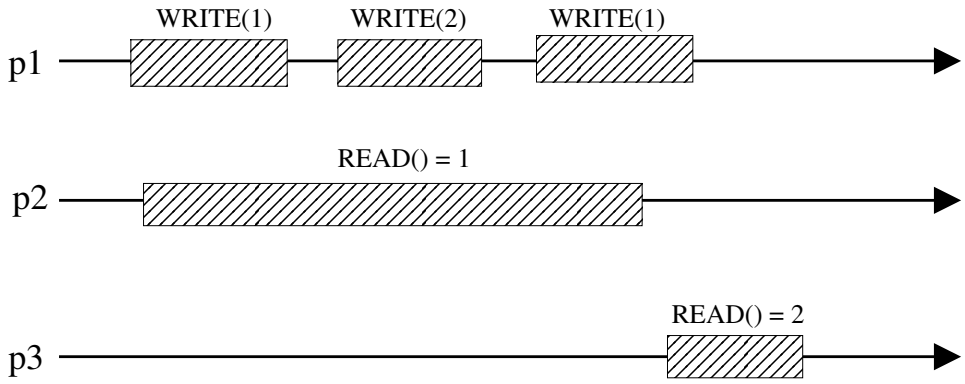
Part 2.a.



Part 2.b.



Part 2.c.



Problem 3. Consider the transformation from **binary** MRSW **safe** registers to **binary** MRSW **regular** registers, given in class.

Part 3.a. Prove that the transformation does **not** generate **multi-valued** MRSW **regular** registers (from **multi-valued** MRSW **safe** base registers) by providing a counterexample that breaks regularity.

Part 3.b. Also, prove that the resulting registers (in the original transformation) are not binary **atomic** (just regular) by providing a counterexample that breaks atomicity.

Problem 4. Consider the transformation from binary MRSW safe registers to binary MRSW regular registers, given in class. Prove that the transformation does **not** generate multi-valued MRSW regular registers (by providing a counterexample that breaks regularity). Also, prove that the resulting registers are not binary atomic (by providing a counterexample that breaks atomicity).

Problem 5. Consider the transformation from binary regular to M-valued MRSW regular registers given in class. Prove that:

1. The resulting registers are regular.
2. The transformation would not work if the Write operation would first write 0, and then 1. (You should provide a counterexample that breaks regularity.)
3. The resulting registers are not atomic. (You should provide a counterexample execution that breaks atomicity.)