

Solutions to Exercise 2

Problem 1.

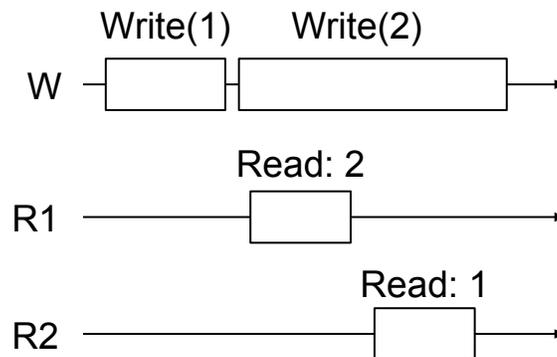


Figure 1: An execution that is possible with a regular register but not with an atomic register. There are three processes: a writer (W) and two readers (R1 and R2). The two reads are concurrent with the second write. The read by R1 completely precedes the read by R2. The execution is not atomic because it is impossible to assign linearization points to all operations: if the linearization point of *Write(2)* is before that of the read by R1, then the read by R2 cannot have a linearization point; if the linearization point of *Write(2)* is after that of the read by R1, then that read cannot have a linearization point.

Problem 2.

Part 1 Please see Chapter 4 of these lecture notes (first item of Supplementary Material on the website), page 51.

Part 2 For this, notice that if the writer first clears the array by writing 0's, it is possible for the value of the array to be all 0's, which is not a valid state.

Part 3 Figure 2 presents an example that violates atomicity. Such an execution can occur if the first *read* operation of p_2 gets 0 while retrieving $Reg[7]$ and gets 1 while retrieving $Reg[1000]$. This can occur since both *write(7)* and *write(1000)* are concurrent with the *read* operation. Afterwards, the second *read* operation of p_2 will return 7 since *write(1000)* has not yet set $Reg[7]$ to zero.

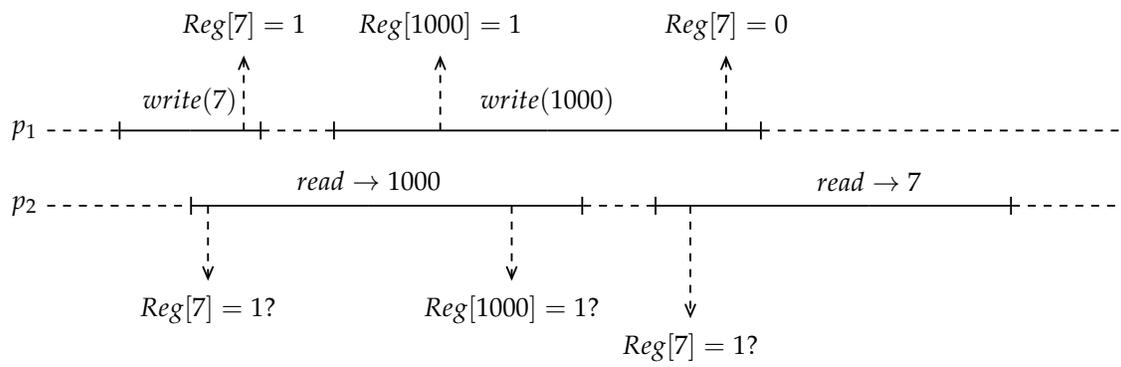


Figure 2: Execution that violates atomicity.