MPRI 2.18.2: Mid-Term Homework, due 14.10.2019

Problem 1: Bakery with Safe Registers

Show that the original Lamport's bakery algorithm (slide 21 in class01-intro.pdf) is correct even when all the registers it uses are only *safe*.

Problem 2: Safety and Liveness

A *property* is a set of histories. Here we consider histories in which processes propose values in $\{0, 1\}$ and then output values in $\{commit, abort\}$. We assume that in a history, a process proposes a value at most once, outputs a value at most once, and only if it previously proposed a value.

Classify the following properties into safety/liveness. If a property is an intersection of the two, specify the corresponding safety and liveness properties. Justify your answers.

- Every process eventually outputs a value.
- If every process proposes 1 and no process crashes (stops taking steps), then no process can output *abort*.
- Eventually, all processes output the same value.

Problem 3: Progress Conditions

We say that a property P *is stronger than* a property P' if $P \subseteq P'$. What is the relation between *starvation-freedom* (SF) and *lock-freedom* (LF)? Explain why.

Problem 4: Atomic Registers

Consider the implementation of a one-writer N-reader ($1WNR$) atomic register (Transformation V in the slides).

In the *read()* operation, the process writes the value it just read back to $RR[\,][\,]$. Is it possible to find an implementation in which the reader *does not* write? Justify your answer.

Problem 5: ABA in Atomic Snapshots

Show that the atomic snapshot is subject to the ABA problem (affecting correctness) in case the written values are *not unique*.