Concurrent Algorithms November 30, 2020

Exercise 9 Solutions

Problem 1. No, the algorithm is no longer correct if we omit line 11 because it might violate the **agreement** property of consensus. In fact, line 11 is crucial for the process in question (that is trying to decide) to detect the presence of other concurrent processes also attempting to decide at the same time. If processes do not make sure they still have the highest timestamp after having written their values in *Reg* in line 10, they might decide too early, breaking agreement.

Problem 2. Recall the properties of Non-Equivocating Broadcast (NEB):

1. Liveness: If a correct process p broadcasts m, then all correct processes eventually deliver m from p.
2. Agreement: If p and q are correct processes, p delivers m from r, and q delivers m' from r, then $m = m'$.
3. Validity: If a correct process delivers m from p, p must have broadcast m.

All three properties are violated by this algorithm. Consider a system with 3 processes, p, q and r, where r is the designated broadcaster.

1. Liveness: Suppose r is correct and initially writes $m = 1$ to its register, then Byzantine p writes $m = 0$ to their register, then process q starts executing alone. Process q will return after satisfying the condition at line 7, violating liveness.
2. Agreement: Consider the following execution. r first writes $m = 1$ to their register. Then, p executes from line 4 to line 10, and sleeps just before writing. Then, r writes $m = 0$ to their register. Then, q executes from line 4 to 10. Then, p and q write and deliver. Note that p delivered $m = 0$ and q delivered $m = 1$.
3. Validity: Suppose q and r are initially asleep, and p executes RECEIVE. Then, p will deliver $\perp$, which was not broadcast by r.