Exercise Session 9
Non-blocking Atomic Commit – Solutions

Problem 1

Devise two algorithms that, without consensus, implement weaker specifications of NBAC by replacing the termination property with the following ones:

- Weak termination: Let p be a distinguished process, known to all other processes. If p does not crash then all correct processes eventually decide. Your algorithm may use a perfect failure detector.
- Very weak termination: If no process crashes, then all processes decide. Is a failure detector needed to implement this algorithm?

Answer: The first algorithm may rely on the globally known process p to enforce termination. The algorithm uses a perfect failure detector $\mathcal{P}$ and works as follows. All processes send their proposal over a point-to-point link to p. This process collects the proposals from all processes that $\mathcal{P}$ does not detect to have crashed. Once process p knows something from every process in the system, it may decide unilaterally. In particular, it decides COMMIT if all processes propose COMMIT and no process is detected by $\mathcal{P}$, and it decides ABORT otherwise, i.e., if some process proposes ABORT or is detected by P to have crashed. Process p then uses best-effort broadcast to send its decision to all processes. Any process that delivers the message with the decision from p decides accordingly. If p crashes, then all processes are blocked.

Of course, the algorithm could be improved in some cases, because the processes might figure out the decision by themselves, such as when p crashes after some correct process has decided, or when some correct process decides ABORT. However, the improvement does not always work: if all correct processes propose COMMIT but p crashes before any other process, then no correct process can decide. This algorithm is also known as the Two Phase Commit (2PC) algorithm. It implements a variant of atomic commitment that is blocking.

The second algorithm is simpler because it only needs to satisfy termination if all processes are correct. All processes use best-effort broadcast to send their proposals to all processes. Every process waits to deliver proposals from all other processes. If a process obtains the proposal COMMIT from all processes, then it decides COMMIT; otherwise, it decides ABORT. Note that this algorithm does not make use of any failure detector.

Problem 2

Consider a variant of NBAC where we replace the Agreement property (i.e., No two processes decide differently) with **Non-uniform Agreement** (i.e., No two *correct* processes decide differently).

In Figure 1 below, there are three processes and their corresponding timeline. Using the same notation as you saw in the class, depict on this figure an execution where each of the three processes proposes an initial value using NBAC and the execution satisfies this variant of NBAC with Non-uniform Agreement, but does not satisfy the conventional NBAC specification (with Agreement).

Explain why does your execution break the Agreement property.

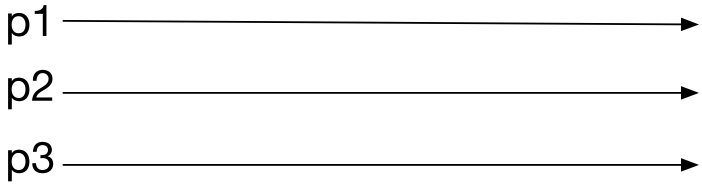

Figure 1: Show on this figure the Proposal events, any Decision events, and any process crash events, such that the execution satisfies NBAC with Non-uniform Agreement (but not with Agreement).

Answer: All processes propose(1). Process p3 decides(1) and then crashes. The other two processes decide(0).

Agreement breaks because process p3 decides to commit (rightfully), whereas the other two correct processes decide both to abort, due to the crash of p3. All other properties of NBAC hold.