Computer Science 2221 **Homework 3 Answers**

Behdad Esfahbod **Student # 993505827**

1. (a) Algorithm: Every process does:
    - Send the input value as your vote to everyone.
    - Receive n votes.
    - If there is any 0 received in votes, decide 0, otherwise decide 1.

    Apparently as there is no fault, everyone would eventually receive the n votes and decide the same thing, so Termination and Agreement hold. No one decides anything that is not a vote of some process, and so it's input value, so Validity holds too. And there is no bivalent initial configuration, since given the input values, everyone would decide the logical AND of the input values, no matter what the actual run of the algorithm is.

    (b) Algorithm: Every process does:
    - Send the input value as your proposal to everyone.
    - Receive the first proposal message, and consider that as your vote.
    - Send your vote to everyone.
    - Receive n votes.
    - If there is any 0 received in votes, decide 0, otherwise decide 1.

    Apparently as there is no fault, everyone would eventually receive a proposal, and then receive the n votes and decide the same thing, so Termination and Agreement hold. No one decides anything that is not a vote of some process, and so the proposal of a process, and so it's input value, so Validity holds too. And there are bivalent initial configurations, since if there are boths zreos and ones in input values, then depending on the first proposal that is delivered to any process, the vote on any proces can be either zero, or one, so the final decision can be either zero, or one.

    (c) Fact (i) is simply no longer valid for 0-tolerant Consensus algorithms, as there exist a 0-tolerant algorithm which has no initial bivalent configuration, as shown above in (a). Even when an algorithm has an initial bivalent configuration, fact (ii) cannot hold, since we have seen that there exist 0-tolerant algorithms with initial bivalent configuration that solve consensus, so the FLP impossibility has no way other than to fail. If there exists an algorithm which violates fact (i) but satisfies fact (ii), is non-sense, as an algorithm with no initial bivalent configuration, means there is no bivalent configuration in any run, so fact (ii) is irrelevant. Putting all together, algorithms in (a) and (b) imply that both Facts (i) and (ii) does not hold for 0-tolerant Consensus algorithms.

2. If we implement a failure detector that is at least as strong as $\diamond S$ as is suggested in the hint, then we can use Mostefaoui and Raynal's 1999 algorithm to solve Consensus. So here we simply implement the failure detector:

    But that is really simple:

    - Initially everyone suspects everyone.
    - Everyone sends a message "I'm not crashed" to everyone.
    - Everyone removes anyone from his suspects list when receives an "I'm not crashed" message from him.

And this simple algorithm is at least as strong as $\diamond S$, as there is a time after which every process that crashes is suspected (completeness) and every process that is not crashed is not suspected (strong accuracy)

3. (a) i. If there are both 0 and 1 among the input values of correct processes, then Strong Validity is satisfied, so we focus on the case that all correct processes have input value v_0. In this case in round one, every correct process receives $n-t$ messages, which at most t of them are from faulty processes, so still $n-2t$ are from correct ones which all have sent v_0. Since $n > 4t$, it is that $n-2t > \frac{n}{2}$, so every correct process selects v_0 and sends it to all processes. Later in that round, again every correct process receives $n-t$ messages, $n-2t$ of them from correct processes with v_0 in it, and since $n-2t > t$, there are at least $t+1$ messages with v_0, so all correct processes would decide v_0 in round 1. From the original proof we know that they would decide the same thing in subsequent rounds too.

   ii. Let's have three processes f, c_1, and c_2 (which one if faulty?). f starts with 1 and sends 1 to the other ones and crashes. c_1 and c_2 start with zero, receive a message from f which is 1, and one 0, so send ?. Both of them receive ?, so toss coins and lets assume they both get 1. So in the following rounds they both decide 1, which is not the initial value of any correct process. So Strong Validity is violated.

   (b) We state and prove the three lemmas in the original proof that are changed here. The rest of the proof (including Termination) is the same, and we do not write again here.

   Lemma 1 It is impossible for a process to propose 0 and another to propose 1 in the same round k.

   Proof If p proposes 0, then it received at least $(n+t+1)/2$, if q proposes 1, it received at least the same number. These two have at most t processes in common, as there are at most t that can vote for both 0 and 1. So we have at least $2(n+t+1)/2 - t = n+1$ unique processes voted, which is a contradiction.

   Lemma 2 If for some $v \in \{0,1\}$ at the end of round $k-1$ we have $x_p = v$ for every correct process p, then all correct processes will decide v in round k.

   Proof By assumption, every report message received in phase k except for at most t, will be for v. Since each process receives $n-t$ reports, at least $n-2t$ of them are for v, and $n-2t > (n+t+1)/2$, it will propose v. Thus, all proposals sent in round k from correct processes will be for v, so there are at least $n-2t >= 3t+1$ of them, each correct process will receive at least $3t+1$ proposals for v in round k and will therefore decide v in round k.

   Lemma 3 If any correct process decides v in round k then all correct processes will decide v in round $k+1$.

   Proof If correct process p decides v in round k, then it received at least $3t+1$ proposals for v in round k. Consider any correct process q. Process q receives $n-t$ proposals in round k, so it can "miss" at most t of the $t+1$ proposals for v that p received, and t more may be from faulty processes. So q receives at least $t+1$ proposal for v, and will therefore set $x_q := v$ in round k. Thus, every q that sends a report in round $k+1$ will have $x_q = v$ at the end of round k. By Lemma 2 then, every correct process will decide v in round $k+1$.

   Now as before, Validity follows from Lemma 2 with $k = 1$, Uniform Agreement follows from Lemma 3, and Termination with probability 1 is like the original proof.