Exercise 11: Counting

1

In the *self-stabilising Byzantine firing squad* problem, in each synchronous round $r \in \mathbb{N}$, each node $v \in V$ receives an external input $\mathrm{GO}(v, r) \in \{0, 1\}$. If $\mathrm{GO}(v, r) = 1$, then we say that v receives a GO input in round r. Moreover, the algorithm determines an output $\mathrm{FIRE}(v, r) \in \{0, 1\}$ at each node $v \in V_g$ in each round $r \in \mathbb{N}$. We say that an execution of an algorithm *stabilizes in round* $r \in \mathbb{N}$, if the following three properties hold:

Agreement: $\mathrm{FIRE}(v, r') = \mathrm{FIRE}(w, r')$ for all $v, w \in V_g$ and $r \leq r' \in \mathbb{N}$.

Safety: If $\mathrm{FIRE}(v, r_F) = 1$ for $v \in V_g$ and $r \leq r_F \in \mathbb{N}$, then there is $r_G < r_F$ s.t.

a) $\mathrm{GO}(w, r_G) = 1$ for some $w \in V_g$ and

b) $\mathrm{FIRE}(v, r') = 0$ for all $r' \in \{r_G + 1, \ldots, r_F - 1\}$.

Liveness: If $\mathrm{GO}(v, r_G) = 1$ for at least $f + 1$ correct nodes $v \in V_g$ and $r \leq r_G \in \mathbb{N}$, then $\mathrm{FIRE}(v, r_F) = 1$ for all nodes $v \in V_g$ and some $r_G < r_F \in \mathbb{N}$.

Note that the liveness condition requires $f + 1$ correct nodes to receive an external GO input, as otherwise it would be impossible to guarantee that a correct node has received a GO input when firing. We say that an execution stabilized by round r has *response time* R *from round* r *on* if

1. if $f+1$ correct nodes $v \in V_g$ satisfy $\mathrm{GO}(v, r_G) = 1$ on some round $r_G \geq r$, then all correct nodes $w \in V_g$ satisfy $\mathrm{FIRE}(w, r_F) = 1$ on some round $r_G \leq r_F \leq r_G + R$, and

2. if there is a round $r_F \geq r$ such that $\mathrm{FIRE}(v, r_F) = 1$ for some correct $v \in V_g$, then there is a round r_G with $r_F > r_G \geq r_F - R$ and some correct node $w \in V_g$ with $\mathrm{GO}(w, t_G) = 1$.

Finally, we say that an algorithm F is an f-resilient firing squad algorithm with stabilization time $S(F)$ and response time $R(F)$ if in any execution of the system with at most f faulty nodes there is a round $r \leq S(F)$ by which the algorithm stabilized and from which on it has response time at most $R(F)$.

a) Given a T-counting algorithm of stabilization time S and message size M_1 alongside a consensus algorithm of running time T and message size M_2, provide a firing squad algorithm with the following properties:

   1. It stabilizes within $\max\{S + T, 2T\}$ rounds.
   2. It has response time $R \leq 2T$.
   3. It has message size $M \leq M_1 + M_2 + 1$.

b) Conclude that a firing squad algorithm with stabilization and response time $\mathcal{O}(f)$ and message size $\mathcal{O}(\log f)$ exists.

c) Prove that any firing squad algorithm must have response time $f + 1$. (Hint: Reduce consensus to firing squad!)

Solution

a) We use the T-counting algorithm to (after it stabilized) run an instance of the consensus algorithm exactly every T rounds, deciding whether to fire or not according to its output exactly T rounds after starting the instance (on all other rounds r, $\mathrm{FIRE}(v,r) = 0$). To determine the input $v \in V_g$ uses for the algorithm, v observes whether it has proof whether any correct node $w \in V_g$ satisfied $\mathrm{GO}(w,r) = 1$ in the T rounds before the current instance started. To this end, all nodes broadcast their GO values each round, and set the input variable to 1 when receiving $f+1$ times 1 in some round; when initializing a consensus instance (at the end of the round the previous one terminated), this variable is used as input and reset to 0.

Some care is necessary here to ensure that the definition is met precisely: When firing in round r, we first reset the input variable to 0, then check whether it is set to 1 again due to receiving $f+1$ times 1, only then use the variable to determine the input of the instance starting to execute in round $r+1$, and finally set it back to 0.

We need to show that this algorithm satisfies the required properties. First, observe that any consensus instance terminating at or after round $S+T$ was executed correctly, as the counting algorithm stabilized by round S. By the agreement property of the consensus routine (and, for other rounds, that nodes simply output 0), the agreement property of the firing squad holds. Concerning safety, any such consensus instance that outputs 1 must have some correct node v use input 1. This input must be the result of receiving v at least $f+1$ times 1 at most T rounds before the instance was started. This, in turn, means that some $w \in V_g$ had $\mathrm{GO}(w,r) = 1$ in the respective round. Moreover, if a firing event happened in the meantime, this implies that the input variable was reset at this time, and must have been set to 1 again in this round, i.e., safety is satisfied either way. This also shows that the second part of having response time $2R$ is satisfied.

Concerning liveness (and the first part of response time $2R$), suppose in round $r \geq S+T$ at least $f+1$ correct nodes $v \in V_g$ satisfy $\mathrm{GO}(v,r) = 1$. If a consenus instance terminates in round r, all nodes will use input 1 in the next instance starting in round $r+1$, it will (T rounds later) output 1 by validity, and hence liveness is satisfied. Otherwise, denote by $r' \in \{r+1, \dots, r+T-1\}$ the next round when an instance terminates. If its output is 1, the nodes will fire and the liveness condition is met. Otherwise, they will use input 1 for the instance starting in round $r' \leq r+T$ and fire in round $r'+T-1 < r+2T$. We conclude that liveness holds in all cases and the first condition of response time $2R$ is satisfied.

The message size bound is immediate from the fact that there is one instance of the counting algorithm, one instance of the consensus algorithm, and one additional broadcasted bit due to sending the GO signals.

b) This follows by using the Phase King algorithm (running time $\mathcal{O}(f)$ and message size 1) together with the $\mathcal{O}(f)$-counting algorithm of stabilization time $\mathcal{O}(f)$ and message size $\mathcal{O}(\log f)$ obtained by plugging it into the counting framework from the lecture.

c) Given a firing squad algorithm with stabilization and response times S and R, respectively, we solve consensus as follows. For an arbitrary initial state, we set $\mathrm{GO}(v,r) = 0$ for all rounds $r \neq S+R+1$ and $\mathrm{GO}(v, S+R+1) = b_v$, i.e., to the input of v in the consensus routine. If there is a round $r \in \{S+R+1, \dots, S+2R+1\}$ in which v fires, it outputs 1. By agreement, safety, and liveness of the firing squad algorithm, this output satisfies the agreement and validity properties of consensus. To make this into an R-round consenus algorithm, we initialize each node with the state it has at the beginning of round $S+R+1$; it knows whether it fires in the

prescribed range within R rounds, i.e., we end up with an R-round consensus algorithm. As we know that consensus requires at least $f+1$ rounds in the worst case, it follows that $R \geq f+1$.

2

In this exercise, you show how to obtain a silent (binary) consensus algorithm from an arbitrary consensus algorithm. As usual, we assume that $f < n/3$. Here's a description of the new algorithm up to determining its output:

The new protocol C' can be seen as a "wrapper" protocol that manipulates the inputs and then lets each node decide whether it participates in an instance of the original protocol. In the first round of the new protocol, C', each participating node broadcasts its input if it is 1 and otherwise sends nothing. If a node receives fewer than $n-f$ times the value 1, it sets its input to 0. In the second round, the same pattern is applied.

Subsequently, C is executed by all nodes that received at least $f+1$ messages in the first round. If during the execution of C a node

1. cannot process the messages received in a given round in accordance with C (this may happen e.g. when not all of the correct nodes participate in the instance, which is not covered by the model assumptions of C),
2. would have to send more bits than it would have according to the known bound $M(C)$, or
3. would violate the running time bound of C,

then the node (locally) aborts the execution of C.

a) Prove that the protocol is silent.

b) Define suitable rules for determining the output of the new protocol C' based on the first two rounds of the wrapper, whether the execution of C was aborted, and, if it wasn't, on its output. Show agreement and validity of C' with these rules.

Solution

a) If all correct nodes have input 0, they will not transmit in the first two rounds. In particular, they will not receive more than f messages in the first round and not participate in the execution of C. Hence correct nodes do not send messages at all.

b) A node outputs 0 in the new protocol if it did not participate in the execution of C, aborted it, or received f or fewer messages in the second round, and it outputs the result according to the run of C otherwise.

We distinguish two cases. First, suppose that all correct nodes participate in the execution of C at the beginning of the third round. As all nodes participate, the bounds on resilience, communication complexity, and running time that apply to C hold in this execution, and no node will quit executing the protocol before termination. To establish agreement and validity, again we distinguish two cases. If all nodes output the outcome of the execution of C, these properties follow right away since C satisfies them; here we use that although the initial two rounds might affect the inputs of nodes, a node will change its input to 0 only if there is at least one correct node with input 0. On the other hand, if some node outputs 0 because it received f or fewer messages in the second round of C', no node received more than $2f < n-f$ messages in the second round. Consequently, all nodes executed C with

input 0 and computed output 0 by the agreement property of C, implying agreement and validity of the new protocol.

The second case is that some correct node does not participate in the execution of C. Thus, it received at most f messages in the first round of C', implying that no node received more than $2f < n - f$ messages in this round. Consequently, correct nodes set their input to 0 and will not transmit in the second round. While some nodes may execute C, all correct nodes will output 0 no matter how C behaves. Since nodes abort the execution of C if the bounds on communication or time complexity are about to be violated, the claimed bounds for the new protocol hold.

3*

a) Contemplate your experience with the lecture.

b) Come up with clever ideas on what we could do better next time.

c) Join us for ice cream and spill the beans!