Dairon Andrés Benites Aldaz
DD2443 HT22 (52168) Parallel and Distributed Computing
October 6, 2022

Lecture 10. Exercise

1. Hand-in

Question 1. Explain carefully why it is theoretically justified to increase each n to the nearest multiple of 3 in the lower bound, Byzantine failures proof.

It is theoretically justified to increase each n to the nearest multiple of 3 in the lower bound since it all reduces to proving that the one A and two (B and C) problem cannot be solved, if A is traitorous. when A tells B to attack and C to retreat, and B and C send messages to each other, forwarding A's message, neither B nor C can figure out who is the traitor, since it is not necessarily A—the other (B or C) could have forged the message purportedly from A. It can be shown that if n is the number of elements in total, and t is the number of traitorous components in that n, then there are solutions to the problem only when n > 3t and the communication is synchronous (bounded delay).

Question 2. Does the correctness proof of the Pease-Shostak-Lamport al- gorithm still work if we replace the condition w Q<=f with the condition w Q=f ? Explain carefully why or why not the replacement works.

Is q to be trusted by p?
Exists subset Q of P of $size > (n + f)/2$ and value v such that $told_p(wq) = v$
for all w in $Q^{<=f}$ =(strings in Q* of length <= f)?
If we change it for $Q^{=f} = (stringsinQ * oflength == f)$?

It will not be correct since for the test to work and q is nonfaulty we must show that some choice of Q exists and all choices of Q agree on assignment, thus it is necessary to show that each choice of Q must inlude a non-faulty process and by restricting it for strings in Q* of length == f we may not get it correct.
Question 3. For the problem of leader election each node in a distributed system executes the same control program and is in one of the states
undecided, not leader, leader.
The task is on termination to ensure that exactly one node is in state leader, and all other nodes are in state not leader. A ring topology is one in which the nodes are connected in a ring. A ring is anonymous if there is no feature (such as a node identifier) that allows telling the nodes apart. You can assume that all nodes are non-faulty (no Byzantine nodes, no crash failures).

- Is leader election possible in an anonymous ring using a synchronous, deterministic control program? Either exhibit an algorithm (in suit- able pseudocode) or prove that the task is impossible.

  It is not possible by Theorem 2.5 (Anonymous Leader Election). Deterministic leader election in an anonymous ring is impossible. Proof: If one node ever decides to become a leader (or a non-leader), then every other node does so as well, contradicting the problem specification [Each node eventually decides whether it is a leader or not, subject to the constraint that there is exactly one leader.] for n > 1. This holds for non-uniform algorithms, and therefore also for uniform algorithms. Furthermore, it holds for synchronous algorithms, and therefore also for asynchronous algorithms.

- Is leader election possible in a synchronous, deterministic ring in which all but one of the processors have the same identifier? Either give an algorithm or prove an impossibility result.
  Yes, it is possible, since one of the processors has an id which is different from every other processors id. We can propose different algorithms to choose the process with different id as the leader. One possible algorithm is algorithm 1. We assume no failures happen.
  To start election
  send (election, my ID) to the left and right processors in the ring
  When receiving message (election,id) from both left and right neighbors
  if my ID is not equal to any of my two neighbors id then
  I am selected as the leader
  send (elected,my ID) to my right neighbor
  end if
  (elected,id) is forwarded clockwise in the ring until it comes back to leader (to inform every processor which processor is the leader)

- Consider a synchronous ring in which exactly two nodes have identifier A and all the other nodes have identifier B. Is deterministic leader election possible in this setting? Either give an algorithm or prove an impossibility result.
  Modify the basic ring-based leader election algorithm to elect 2 leaders (two processes with the highest IDs). Answer: To start an election, a process sends a message $<$ election $>$ with its ID appended. Each node that receives this message appends its ID to it. Once the election message reaches the initiator after going through the circle, the top two nodes (nodes with the highest and second highest IDs) are selected, and a message $<$ choose : highest1,highest2 $>$ is sent with the initiator's ID appended to it. Each node that receives this message appends its ID again. When this message reaches the initiator, if the appended ID list contains the top two nodes, then the election is successful and ends. Otherwise, a new election is initiated.