University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, A. Mályusz, Z. Parsaeian

Theory of Distributed Systems
Sample Solution Exercise Sheet 6

Exercise 1: Leader Election with Failures

Consider the leader election problem on a complete graph in the synchronous message passing model. That is, every node has a unique ID and at end of the algorithm, every node that did not crash has to output the ID of the leader node.

Show that if at most $f \leq n-2$ processes fail during the protocol, at least $f+1$ rounds are needed to solve leader election.

Sample Solution

Intuitively it is clear that Leader election should not be easier than the consensus Problem, for which we know that it takes at least $f+1$ rounds to solve from the lecture. In the following we will see two different ways of using the knowledge that consensus is hard to show that leader election is hard as well.

Adaptation of the proof for consensus: In the *(f+1)-round lower bound proof for consensus* we built a similarity chain between an execution with f rounds with all inputs 0 and no crashes and an execution with f rounds with all inputs 1 and no crashes. But in fact the concrete input values did not play any role when building this chain, which means that we proved an even stronger property: We can build a similarity chain between two executions with f rounds and no crash failures and arbitrary input values.

Now assume there is a protocol for solving leader election in $f \leq n-2$ rounds. Take pairwise distinct integers $x_1, \ldots, x_n, y_1, \ldots, y_n$ which represent the IDs of two leader election instances. Let E be the execution with inputs $x_1, \ldots, x_n$ and no crash failures and E' the execution with inputs $y_1, \ldots, y_n$ and no crash failures. As there is a similarity chain between E and E' (the same as shown in the lecture), all nodes must have the same output in both executions which is a value among $x_1, \ldots, x_n$. But then in E' the output is not one of the IDs $y_1, \ldots, y_n$ a contradiction.

Proof by reduction: Assume there is an algorithm solving leader election in at most f rounds, i.e., after f rounds all non crashed nodes output the same value which is the ID of some node. Now assume every node v has an input $x_v \in \{0,1\}$. Node v changes its identifier $\mathrm{ID}(v)$ to $\mathrm{ID}'(v) = 2 \cdot \mathrm{ID}(v) + x_v$ (take the binary representation of $\mathrm{ID}(v)$ and concatenate it with x_v). Then run the leader election algorithm, but every node only outputs the last bit of the ID of the leader node. Then we have solved binary consensus in f rounds, a contradiction.

Exercise 2: k-Set Agreement with Failures

A generalization of consensus is the k-set agreement problem: Every node has *some* input value and at the end every node has to output a value such that the following validity properties are fulfilled:

1. There must not be more than k different output values.
2. Every node that did not fail must output a value which was input of some node.

Show that on a complete graph in the synchronous message passing model with at most f node *failures*, the k-set agreement problem is solvable in $\lfloor f/k \rfloor + 1$ rounds. Argue why that upper bound is tight.

Sample Solution

Algorithm: Each node v maintains a value x_v which is initially set to v's input. The output of v at the end of the algorithm is the current value of x_v. For $\lfloor f/k \rfloor + 1$ rounds, each node does the following

1. Broadcast x_v
2. Receive values $y_1, \ldots, y_m$
3. Update x_v to $\min\{x_v, y_1, \ldots, y_m\}$

Analysis: Since steps 1 to 3 can be done in a single round, the runtime is clear. Property (2.) is also clear. For Property (1.) assume there is a round with $\ell < k$ crash failures. Let $x_1, \ldots, x_\ell$ be the values of the nodes that crashed in the current round and let V be the set of nodes that survive the round. Then at the end of the round, each node in V chooses a value among $x_1, \ldots, x_\ell, \min\{x_w | w \in V\}$ as the values in $\{x_w | w \in V\}$ arrive at every node. So every node chooses among $\ell + 1 \leq k$ different values and the number of different values can only be decreased in the following rounds. This means that as soon as there is a round with less than k crashes, the algorithm fulfils agreement. In $\lfloor f/k \rfloor + 1$ rounds, there must be at least one round with less than k crashes.