Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2009 Prof. R. Wattenhofer / C. Lenzen / T. Locher

Principles of Distributed Computing
Exercise 6

1 Deterministic Consensus

In this exercise, we assume that communication takes place in synchronous communication rounds. Consider the following simple deterministic algorithm:

Algorithm 1 Simple Deterministic Consensus

```
Broadcast own value to all other processors
Receive values from all other processors
Decide on the minimum value
```

We assume that nodes can crash at any point in time, in particular while sending messages to other processors.

a) Give an example that shows that Algorithm 1 does not guarantee that all processors reach consensus!

b) Assuming that at most f nodes crash, give an algorithm, similar to Algorithm 1, that guarantees that nodes reach consensus![1] How many rounds does your algorithm need?

2 Randomized Consensus

In the lecture, we studied a randomized consensus algorithm (Algorithm 27). In the remarks it says that the algorithm can be modified to tolerate up to $f < n/4$ crashing nodes.

a) Show that the algorithm tolerates $f < n/8$ crash failures, but is incorrect if $f \geq n/8$.

Hint: The algorithm breaks down even if no processes crash at all!

b) Modify the algorithm such that it deals with $f < n/4$ crash failures.

c*) Give an algorithm that solves the (asynchronous, crash-failure) consensus problem for arbitrary initial inputs from an alphabet $\alpha_1, \ldots, \alpha_m$! Prove that the number $f(n, m)$ of tolerated failures is optimal.

[1] You can use the number f in your algorithm.