Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2008 Prof. R. Wattenhofer / Dr. F. Kuhn / T. Locher / Y. A. Oswald / C. Lenzen

Principles of Distributed Computing
Exercise 8

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 30). In the remarks it says that the algorithm does not only work if nodes crash, but also if up to f nodes deliberately send arbitrary information in each round (Byzantine behavior).

a) Argue why the algorithm tolerates Byzantine behavior! (Do not worry about the time complexity!)

Now we again assume that nodes may simply crash. We simplify Algorithm 30 in that we delete lines 3-9, i.e., the bid of each node u is its local value x_u.

b) Does this simplified algorithm still guarantee that nodes reach consensus in the crash failure model? If your answer is yes, prove that consensus is reached. Otherwise, give an example where the simplified algorithm fails!

c*) Give an algorithm that solves the consensus problem for arbitrary initial inputs from an alphabet $\alpha_1, \ldots, \alpha_m$!

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