Institute of Computer Science, University of Bern
Distributed Algorithms, Spring 2020

Prof. Christian Cachin
Jovana Mićić

Exercise 9

9.1 Flooding Consensus (3pt)

Consider *Flooding Uniform Consensus* [CGR11, Algorithm 5.3], in which every process always uses N rounds before deciding.

a) Can one reduce the number of rounds for some process? Consider a system with two processes p and q.

b) Is this algorithm correct if the failure detector is not perfect ($\mathcal{P}$) but is only the eventually perfect one ($\Diamond\mathcal{P}$)? Justify your answer.

9.2 Leader-Driven Consensus (4pt)

Study the *Leader-Driven Consensus* [CGR11, Chap. 5.3], where epoch-change and epoch consensus are implemented as described.

a) Why does this algorithm require a majority of correct processes?

b) Which property is violated if there is no majority of correct processes?

c) Draw or describe an execution with four processes p, q, r, s that justifies your previous answers.

9.3 Leader-Driven Consensus, optimized (3pt)

In practice, *Leader-Driven Consensus* [CGR11, Chap. 5.3] protocol is often implemented with slightly fewer messages. This optimiziation assumes that all processes start with the same process ℓ_0 as the first leader and that ℓ_0 is correct. Describe why the first epoch consensus instance may omit the first round of message exchange (of the READ and STATE messages) between ℓ_0 and the other processes.

References

[CGR11] C. Cachin, R. Guerraoui, and L. Rodrigues, *Introduction to reliable and secure distributed programming (Second Edition)*, Springer, 2011.