

Concurrent Algorithms: Exercise Set 3, Problems 2-3

October 16, 2012

Problem 2

Consider the *binary consensus* problem. Recall that this abstraction satisfies the following properties:

Agreement No two processes decide different values.

Validity The value decided is one of the values proposed.

Let's assume that processes might in fact *crash*, i.e. stop taking steps, at any point during the execution of an algorithm. We add an extra condition on progress:

Termination Every process that does not crash will eventually decide.

Assume the following theorem is true:

Theorem 1 (FLP) *Binary consensus is impossible among N processes, if one of them might fail by crashing, in an asynchronous system that disposes of binary SRSW safe registers¹.*

Using what you know about registers, prove the following result:

Theorem 2 (Students' FLP) *Binary consensus is impossible among N processes, if one of them might fail by crashing, in an asynchronous system that disposes of MRMW atomic registers.*

Solution for Problem 2

Assume for the sake of contradiction that there exists an algorithm \mathcal{A} that solves consensus in an asynchronous system using MRMW atomic registers. We prove that in this case there exists an algorithm \mathcal{A}' that solves consensus in an asynchronous system using SRSW safe registers, which contradicts the FLP theorem.

Let us write down the code of algorithm \mathcal{A} . The algorithm makes use of a (possibly infinite) number of MRMW atomic registers R_1, R_2, \dots . We know from the course that there exists an implementation of MRMW atomic registers using (an infinite number of) SRSW safe registers. Therefore, for each call of R_i .read or R_i .write that the algorithm \mathcal{A} makes, we replace the call

¹In fact, this is one of the major results in distributed computing, first stated by Fisher, Lynch and Patterson, in their paper "Impossibility of Distributed Consensus with One Faulty Process" in 1985. For details, see the presentation in the Encyclopedia of Algorithms (available on Google Books), or the original paper, which is quite readable.

with the implementation of R_i using only SRSW safe registers. We call the resulting algorithm code \mathcal{A}' .

To conclude the proof, we notice that algorithm \mathcal{A}' implements consensus using only SRSW registers. The correctness of \mathcal{A}' follows from the (assumed) correctness of \mathcal{A} and the correctness of the register implementation, which has been shown in class.

Solution for Problem 3

Follows from 1 and 2.