KTH Computer Science
and Communication

Hand-in 2
DD2451 + FDD3008, Parallel and Distributed Computing
Fall 2011

Posted Nov 18 16.00. Due Nov 25 16.00. Answers can be mailed to mfd@kth.se as pdf or dropped in Mads' intray, located on level 4, Lindstedtsvägen 3 (enter through the doors with the label "NADA", after going a small distance to the left proceed straight and you find the department mail boxes on your right). Make sure answers are clearly marked with name and mail account on each sheet. The general rules on homework solutions available at the course home-page apply. In particular, discussions of ideas in groups of up to at most two people are allowed but solutions should be written down individually, and you should note the name of your discussion partner. Some of the problems below are "classical" and hence their solutions are probably posted on the Internet. It is *not* allowed to use such solutions *in any way*. The order of the problems is "random" and hence do not expect that the lowest numbered problems are the easiest. Any corrections or clarifications on this problem set will be posted under "Exercises and hand-ins" on the course home page (google DD2451).

1. (20p) A read/increment object has initial value 0 and operations `read()` and `inc()`. We implement an n-thread shared read/increment object I thus: I has n MRSW integer valued registers, each with initial value 0. Upon an increment by thread i, I increments the i'th MRSW register. Upon a read, thread i reads all MRSW register one at a time and returns the sum. Show that if each MRSW register is atomic then the read/increment register is linearizable. Is this also the case when the MRSW registers are regular? Explain.

2. (15p) H&S exercise 61

3. (15p) Consider carefully the lock-free list algorithm in H&S chapter 9. Explain carefully why the `contains` method is wait-free, and why the `add` and `remove` methods are lock-free, and not wait-free.

4. (15p) H&S exercise 117

5. (15p) Formulate a version of FloodSet for arbitrary connected graphs. That is, graphs are connected at the outset and no crash failure causes the graph to become disconnected. Prove that it works correctly.

6. (20p) Below, try to avoid excessive listings of e.g. p said p1 said q said v assertions. Explain how the scenarios are set up and how they work, and why they are counterexamples to Byzantine Agreement.

   6a (10p) Describe a scenario for which the simple Byzantine Agreement algorithm fails to solve consensus, for the case n = 6 and f = 2.
   6b (10p) Do the same for the King algorithm.

Good luck!