Distributed Systems Group
Department Software Technology
Faculty EEMCS
DELFT UNIVERSITY OF TECHNOLOGY

Examination Distributed Algorithms (IN4150)

2 February 2018, 9:00-11:30 AM

Notes

1. The **end time** of this exam is **11:30 AM**.
2. **No material** is allowed except for the sheet with the list of algorithms from the course that will be handed out.
3. The **number of exercises** in this exam is 3, and the **number of pages** of this exam is 2.
4. Do give **clear, precise, and concise** answers.
5. The **maximum number of points** to be obtained for each part of each exercise is indicated between parentheses. These numbers of points add up to 19. The remaining 3 points can be earned for the clarity, preciseness, and conciseness of the answers. Students are strongly encouraged to first write (or at least sketch) their answers on scratch paper before writing them on the sheets they will hand in.
6. The **final grade** for the exam is computed as 12 plus the total number of points obtained for the three exercises and the paper summary divided by 10 and rounded to the nearest integer or integer+0.5.

1. (a) (3) Give the definition of causal ordering of point-to-point messages.
   (b) (5) Give an example of a system with three processes in which causal ordering of point-to-point messages is violated.
   (c) (6) Explain in your own words the algorithm of Schiper-Eggli-Sandoz for causal ordering of point-to-point messages.
   (d) (5) Show exactly how the algorithm of (c) would have enforced causal message ordering in your example of (b).

2. (a) (3) Explain the notion of *request sets* in Maekawa's algorithm for mutual exclusion.
   (b) (3) What is the order of the minimal size of these sets when there are N processes? Give an example where this order is attained.
   (c) (5) Explain in your own words Maekawa's algorithm for mutual exclusion without the part dealing with deadlock prevention.
   (d) (8) Explain in your own words the part of Maekawa's algorithm for mutual exclusion that deals with preventing deadlocks.

3. (a) (5) Show that there is no solution to the problem of Byzantine agreement in synchronous systems with (in total) three generals and one traitor.
   (b) (5) Explain why executing $f + 1$ rounds in the algorithm for achieving agreement in synchronous systems with stopping failures indeed does guarantee agreement.
   (c) (4) Explain the problem of *state machine replication* (SMR).
   (d) (5) Explain the condition $n > 3f$ for a solution to the SMR problem where n is the (total) number of servers and f is the maximum number of servers that may exhibit Byzantine failures.