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

Examination Distributed Algorithms (IN4150)

9 April 2013, 9-12 AM

Notes

1. The number of pages of this exam is 3, and the number of exercises in this exam is 4.
2. Students who do the exam of this course **for the first time**, and students who have done the exam of this course previously but who have still **submitted a paper review**, do only **the first three exercises** of this exam; their paper review counts as the fourth exercise.
3. Students who have done the exam of this course previously and who have not submitted a paper review, do **all four exercises** of this exam. They have to indicate at the top of the first sheet of their exam the date when they did the exam for the last time.
4. The maximum number of points to be obtained for each part of each exercise is indicated between parentheses.
5. The final grade is computed as 12 plus the total number of points obtained for the three exercises and the paper review (or: for the four exercises) divided by 10 and rounded to the nearest integer or integer+0.5.
6. The solutions to the exercises can be either in Dutch or in English.
7. Give **short, concise, and precise answers**.

1. (a) (8) Give **in words** the algorithm of Chandy and Lamport for detecting the global state of a distributed system.

   Consider a system consisting of three bank accounts (processes) P_1, P_2, and P_3 with some initial values and with a separate channel each way between every pair of accounts. Processes can transfer amounts of money to other processes by sending messages.

   (b) (5) Present a scenario in which every process sends at least one message to every other process and an execution of the algorithm in that scenario. Clearly specify the global state that has been detected.

   (c) (5) List the *pre-recording* and the *post-recording* events of every process, indicate in which order all of these events across all processes have occurred in the execution of the algorithm, and argue whether or not the detected state has actually occurred.

   (d) (4) If the detected state has occurred, show a different order of the events such that the same state will still be detected but will not have occurred.

   If the detected state has not occurred, show a different order of the events such that the detected state would have occurred.

2. (a) (6) Give definitions of the concepts of *fragment*, the *level* of a fragment, and the *minimum-weight outgoing edge* of a fragment in the Minimum-Weight Spanning Tree (MST) algorithm of Gallager, Humblet, and Spira (GHS).

   (b) (6) Explain the concepts of *merging* two fragments and of *absorbing* one fragment into another in this algorithm.

   (c) (4) Let G be a complete binary tree with 7 nodes (i.e., all non-leaf nodes have two child nodes and all leaf nodes are at distance 2 from the root). Show an assignment of the weights $1, 2, \ldots, 6$ to the edges of G such that the level of the MST created by the GHS algorithm is 2, and show an execution of the GHS algorithm in G with this weight assignment such that at some point in the execution all fragments are of level 1.

   (d) (6) Argue that for any weighted graph there is an execution of the GHS algorithm such that at some point in the execution of the algorithm, all fragments are of level 1.

3. (a) (4) Formulate the Byzantine agreement problem. In particular, state the conditions for *agreement* and *validity*.

   Assume in the rest of this exercise that no authentication is used.

   (b) (5) If there are in total n processes (generals), what is the maximum number f of failing processors that can be tolerated? Show that there is no solution to Byzantine agreement with $n = 3$ and $f = 1$.

   (c) (5) Give in words or in pseudo-code the algorithm OM(f) for Byzantine agreement.

   (d) (4) What is the message complexity of OM(f)? Explain your answer.

   (e) (4) Show the execution of OM(f) for $n = 4$ and $f = 1$ in which the commander does not exhibit failures (is loyal).

Note: Exercise 4 has to be done only by students who have done this exam previously and who have not submitted a paper review. Solutions by any other students will be ignored.

4. (a) (5) Explain the concept of *synchronizers*.

   (b) (6) Give **in words** the algorithm of the *alpha-synchronizer*.

   (c) (7) Give in words or in pseudocode the algorithm for agreement in synchronous systems with stopping (crash) failures.

   (d) (4) Argue why, with at most f incorrect processes, $f + 1$ rounds are sufficient for the correctness of this algorithm.