UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MEng Honours Degrees in Computing Part IV
MEng Honours Degree in Information Systems Engineering Part IV
MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the City and Guilds of London Institute

PAPER 4.37 / I4.12

DISTRIBUTED ALGORITHMS

Tuesday, May 6th 1997, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1. Mutual Exclusion: A system consists of a number of distributed nodes each of which needs exclusive access to a particular resource at various times during its lifetime.

a. What are the main safety and liveness properties required?

b. The nodes are connected together to form a *logical ring*. *Briefly* describe a simple token ring algorithm which provides for exclusive access by one node at a time.

Briefly and informally justify that it satisfies the required safety and liveness conditions.

c. In Lamport's algorithm for mutual exclusion, processes use three types of timestamped messages: request, release and acknowledgement messages. Each process maintains Q, a local copy of the state of all processes. In a system of n processes, outline the following information for process Pi:

i) the information stored in Q and its initial value,

ii) the condition required for exclusion,

iii) the actions of a process on receipt of a timestamped request message.

Briefly and informally justify that it satisfies the required safety and liveness conditions.

d. Lamport's algorithm (in part (c)) requires that there be no overtaking of messages between sites. Show a scenario where, if two processes P0 and P1 both send request messages, and overtaking of an acknowledgement message is possible, it can lead to a violation of mutual exclusion. Indicate the state of the local Q's in each process at each stage.

The four parts carry, respectively, 10%, 20%, 40% , 30% of the marks.

2. Distributed Termination Detection:

a. Define what is meant by distributed termination of a set of processes. What is "pseudo-termination"?

b. A distributed system of connected processes is such that there is a path from any one process to any other in the system.

i) Outline a *diffusing* computation algorithm (such as that of Dijkstra and Scholten) which can be used to detect termination of a distributed computation. State any assumptions that you make.

ii) Briefly and informally justify that your algorithm detects true termination and avoids detection of "pseudo-termination".

c. It has been stated that token loss on a ring can be considered to be a form of termination detection problem. Comment on the validity of this statement. If helpful, you may use the Misra two token "Ping-Pong" algorithm as an example to illustrate your argument.

The three parts carry, respectively, 20%, 60%, 20% of the marks.

3. Atomic Commitment

a. State the properties required of Atomic Commitment, distinguishing between the two variants (weak and strong) of Termination.

b. Consider a synchronous system subject to *link* failures. In such a system describe an execution of the Three-Phase Commit protocol where the *Agreement* property is violated. (Hint: It is enough to consider a system with just two processes.)

c. In the standard Two-Phase Commit protocol, the coordinator *first* decides (after collecting all the votes in round 1), and *then* informs all other processes of its decision (in round 2). Assume that *at most one process* may fail and that there are *no link* failures.

Explain why, even under these assumptions, no extension of the standard Two-Phase Commit protocol with additional rounds can satisfy the Strong Termination property.

d. Consider the following modification of the standard Two-Phase Commit protocol:

The coordinator does not decide in round 1. Instead, in round 2 it informs the other processes of the impending decision and then decides. Assume that *at most one process* may fail and that there are *no link* failures.

Describe how to extend this modified version of Two-Phase Commit with additional rounds so that the resulting algorithm satisfies all the properties of Strong Atomic Commitment. Informally explain why your algorithm satisfies the Agreement and Strong Termination properties.

The four parts carry, respectively, 10%, 20%, 30%, 40% of the marks.

4. Consensus

a. State the safety and liveness properties satisfied by the primitives Send(m) and Recv(m), for sending/receiving message m over a link from process p to process q.

b. A *FIFO link* from process p to process q is defined in terms of primitives FSend(m) and FRecv(m), for sending/receiving message m from p to q which, in addition to the properties in part (a), also satisfy the following:

> q does not receive m from p unless it has previously received all messages that p sent to q before m.

Explain how to implement a FIFO link given a nonFIFO link. In other words, describe an implementation of the FSend(m) and FRecv(m) primitives in terms of the Send and Recv primitives.

c. A *FIFO asynchronous system* is an asynchronous system in which all links are FIFO. Show that Consensus is not solvable in a FIFO asynchronous system where at most one process may crash and no link failures may occur. (Hint: Use the result in part (b); you may do so even if you did not complete part (b).)

Turn over ...

d. Consider a *synchronous* system of three processes connected by a complete network where *at most one process* and *at most one link* may fail. Give a careful proof that Consensus is not solvable in such a system. (Hint: If process A, and the link between processes B and C fail, then processes B and C are unable to communicate.)

The four parts carry, respectively, 10%, 20%, 20%, 50% of the marks.

End of paper