CS425/CS525 Final Exam

May 10th, 2010

Write your answers in the blue book(s). Justify your answers. Work alone. Do not use any notes or books.

There are four problems on this exam, each worth 20 points, for a total of 80 points. You have approximately three hours to complete this exam.

1 Anti-consensus (20 points)

A wait-free *anti-consensus* protocol satisfies the conditions:

Wait-free termination Every process decides in a bounded number of its own steps.

Non-triviality There is at least one process that decides different values in different executions.

Disagreement If at least two processes decide, then some processes decide on different values.

Show that there is no deterministic wait-free anti-consensus protocol using only atomic registers for two processes and two possible output values, but there is one for three processes and three possible output values.

Clarification: You should assume processed have distinct identities.

2 Odd or even (20 points)

Suppose you have a protocol for a synchronous message-passing ring that is anonymous (all processes run the same code) and uniform (this code is the same for rings of different sizes). Suppose also that the processes are given inputs marking some, but not all, of them as leaders. Give an algorithm for determining if the size of the ring is odd or even, or show that no such algorithm is possible.

Clarification: Assume a bidirectional, oriented ring and a deterministic algorithm.

3 Implementing atomic snapshot arrays using message-passing (20 points)

Consider the following variant of Attiya-Bar-Noy-Dolev for obtaining snapshots of an array instead of individual register values, in an asynchronous message-passing system with $t < n/4$ crash failures. The data structure we are simulating is an array a consisting of an atomic register $a[i]$ for each process i, with the ability to perform atomic snapshots.

Values are written by sending a set of $\langle i, v, t_i \rangle$ values to all processes, where i specifies the segment $a[i]$ of the array to write, v gives a value for this segment, and t_i is an increasing timestamp used to indicate more recent values. We use a set of values because (as in ABD) some values may be obtained indirectly.

To update segment $a[i]$ with value v, process i generates a new timestamp t_i, sends $\{\langle i, v, t_i \rangle\}$ to all processes, and waits for acknowledgments from at least $3n/4$ processes.

Upon receiving a message containing one or more $\langle i, v, t_i \rangle$ triples, a process updates its copy of $a[i]$ for any i with a higher timestamp than previously seen, and responds with an acknowledgment (we'll assume use of nonces so that it's unambiguous which message is being acknowledged).

To perform a snapshot, a process sends SNAPSHOT to all processes, and waits to receive responses from at least $3n/4$ processes, which will consist of the most recent values of each $a[i]$ known by each of these processes together with their timestamps (it's a set of triples as above). The snapshot process then takes the most recent versions of $a[i]$ for each of these responses and updates its own copy, then sends its entire snapshot vector to all processes and waits to receive at least $3n/4$ acknowledgments. When it has received these acknowledgments, it returns its own copy of $a[i]$ for all i.

Prove or disprove: The above procedure implements an atomic snapshot array in an asynchronous message-passing system with $t < n/4$ crash failures.

4 Priority queues (20 points)

Let Q be a priority queue whose states are multisets of natural numbers and that has operations enq(v) and deq(), where enq(p) adds a new value v to the queue, and deq() removes and returns the smallest value in the queue, or returns null if the queue is empty. (If there is more than one copy of the smallest value, only one copy is removed.)

What is the consensus number of this object?