Problem Set 6, Part a

Due: Thursday, December 1, 2011

Readings:

Borowsky, Gafni, Lynch, Rajsbaum paper.
Attiya, Welch, Section 5.3.2
Attie, Guerraoui, Kouznetsov, Lynch, Rajsbaum paper
Distributed algorithms book, Chapter 17
Lamport’s “Part-Time Parliament” paper

Next week: Chandra, Toueg paper
Chandra, Hadzilacos, Toueg paper

Problems:

1. As noted in class, the Borowsky, Gafni, et al. paper has a liveness bug in the main protocol. Namely, a simulating process $i$ may repeatedly decide to select the same process $j$ to perform a snapshot, using safe-agreement, neglecting some other process $j'$.

   (a) Why doesn’t the task structure of process $i$, which has a separate task for each simulated process, ensure progress for all the simulated processes?

   (b) Informally describe a simple modification to the algorithm that would fix this problem and guarantee that all the simulated processes get fair turns.

2. Consider the following decision problem, which we call the $c$-shrinking problem, where $c$ is a positive integer. The value domain $V$ is the set of real numbers. For any input vector $I$ of elements of $V$, the allowable output vectors are those vectors $O$ for which (i) every element in $O$ is in the range of the values in $I$, and (ii) the difference between any two values in $O$ is at most $\Delta/c$, where $\Delta$ is the maximum difference between two values in $I$.

   (a) Consider an asynchronous read/write shared-memory system with 10 processes and at most one stopping failure. Describe a very simple algorithm $A$ that solves the 9-shrinking problem for this model. Explain why it is correct.

   (b) Consider the same algorithm run for 3 processes with at most one stopping failure. For what value of $c$ does it solve the $c$-shrinking problem?

   (c) Now consider another 3-process algorithm with one stopping failure, based on applying the BG transformation to the 10-process algorithm from part (a). For what value of $c$ does this solve the $c$-shrinking problem?

3. Exercise 17.10. You may write pseudocode or use Tempo to describe your algorithm.

4. In the first phase of the Paxos consensus algorithm, a participating process $i$ performs a step whereby it abstains from an entire group of ballots at once; namely, the set $B$ of all ballots whose identifiers are strictly less than some particular proposed ballot identifier $b$, and that $i$ has not already voted for. This set $B$ may include ballots that have not yet been created.

   Suppose that, instead, process $i$ simply abstained from all ballots in the set $B$ that it knows have already been created. Does the algorithm still guarantee the agreement property? If so, give a convincing argument. If not, give a counterexample execution.