Problem Set 2, Part b

Due: Thursday, March 6, 2008
Problem sets will be collected in class.
Please follow the new submission formatting directions e-mailed to the class.

Reading:

Aguilera, Toueg paper
(Optional) Keidar, Rajsbaum paper
Section 7.1-7.3 (skip 7.3.4)
Chapter 8.

Reading for next week: Chapter 14, 15.

Problems:

1. In class, we proved a simple special case of the Keidar-Rajsbaum lower bound on the number of rounds for early-stopping consensus: For $2 \leq f < n$, no f-fault-tolerant uniform stopping consensus algorithm can guarantee that, in all failure-free executions, all nonfaulty processes decide within one round.

   In this problem, we will extend these ideas to obtain the special case for 1 failure: For all $3 \leq f < n$, no f-fault-tolerant uniform stopping consensus algorithm can guarantee that, in all executions with at most one failure, all nonfaulty processes decide within two communication rounds.

   To show this, assume for contradiction that such an algorithm exists. We will consider only executions in which at most one process fails in each round.

   (a) Describe how to construct a "chain" of 1-round executions, each with at most one failure, starting from a failure-free execution in which everyone starts with 0, ending with a failure-free execution in which everyone starts with 1, and such that each consecutive pair is indistinguishable to all except possibly one process.

   (b) Consider the unique failure-free extension of any execution α in this chain. In that extension, all nonfaulty processes must decide within at most one more round. Why?

   (c) Show that there must be two consecutive executions in the chain, α_0 and α_1, such that $val(\alpha_0) = 0$ and $val(\alpha_1) = 1$.

   (d) Let i be the unique process to which α_0 and α_1 look different. Let j be any other process that is nonfaulty in both α_0 and α_1. Define 2-round executions β_0 and β_1, which are like failure-free extensions of α_0 and α_1 respectively, with the exception that i fails. Describe a way for i to fail in each execution such that process j must decide 0 at the end of round 2 in β_0 and must decide 1 by the end of round 2 in β_1.

   (e) Complete the argument to obtain a contradiction.

2. (a) Draw a "skeleton" of a Bermuda Triangle (Section 7.1.3) for $n = 5$, $r = 1$, $k = f = 2$. Label the vertices on the three faces with the appropriate initial value vectors.

   (b) Choose a tiny simplex in the interior of the triangle. Describe in (high-level) words the runs associated with the vertices of the tiny simplex. (e.g., how do they differ from each other?)

3. Clearly describe a specific execution of the Three-Phase Commit protocol that requires as many rounds as possible.

4. Exercise 8.5. Write your algorithm using the pseudocode style used in the book. If you have time, we encourage you to write the (non-task portions of the) code using Tempo. This is not required for this assignment, but will be required for Problem Set 3.

5. The standard FIFO reliable channel automaton $C_{i,j}$, defined on p. 204, can itself be regarded as an "optimized version" of another automaton $C'_{i,j}$, which keeps track in its state of the entire history of messages that have been sent. C' also maintains a pointer that indicates the last message that has been delivered.

   (a) Write I/O automaton pseudocode for $C'_{i,j}$. (You could write this using Tempo.)

   (b) Prove that $C_{i,j}$ implements $C'_{i,j}$, in the sense that $\mathit{traces}(C_{i,j}) \subseteq \mathit{traces}(C'_{i,j})$, using a simulation relation.

   (c) Is it also true that $C'_{i,j}$ implements $C_{i,j}$, in the same sense? If so, give a simulation relation to demonstrate this implementation relationship; you do not need to prove that it is in fact a simulation relation. If not, then show a counterexample.