Problem Set 3, Part b

Due: Thursday, October 22, 2015 Problem sets will be collected in class. **Please hand in each problem on a separate page.**

Readings:

Section 8.5
Chapter 14 (skim).
Chapter 15

For next week: Chapter 15 cont'd, Chapter 16

Problems:

6. Consider the following channel D, which is similar to channel C on p. 204 of the Distributed Algorithms book, but allows internal message duplication. In addition to the *send* and *receive* actions, D has two internal actions, *duplicate* and *discard*. When a $send(m)$ occurs, the message m is added to the end of the queue along with a Boolean tag. Tags for successive messages that are sent alternate, $1, 0, 1, 0, \ldots$. A *duplicate* causes an arbitrary message in the queue to be duplicated in place, along with its tag. The channel also keeps track of the tag of the last message delivered. A *receive* delivers the first message on the queue, as before, but only if the tag is unequal to that of the last message delivered. A *discard* discards the first message on the queue, provided the tag is the same as that of the last message delivered.

   (a) Write pseudocode in the style of the book (e.g., Chapter 8, 15) for automaton D.

   (b) Prove carefully that D implements C, in the sense of inclusion of sets of traces. Use a simulation relation.

7. Exercise 15.12.

   Consider the problem of leader election in networks based on bidirectional *line graphs*; such a graph consists of n processes numbered $1, ..., n$, arranged in a line, with birectional edges between each pair of neighbors. Assume that each process knows its neighbors by the local names "right" and "left", with the orientation consistent along the line. Assume that each process knows whether or not it is an endpoint. Assume that the processes have no knowledge of n.

   (a) Give a leader-election algorithm for such networks that uses a small number of messages.

   (b) Why does this result not contradict the lower bound in Lemma 15.14?

8. (Based on Exercise 15.20.) Design an algorithm that allows a distinguished process i_0 in an asynchronous network based on an arbitrary connected undirected graph G to determine the number of nodes in G.

   (a) Explain your algorithm in words.

   (b) Give pseudocode.

   (c) Analyze its message and time complexity.

9. Think about your term project. Reminder that one page proposals are due October 29th.