Problem Set 3, Part a

Due: Thursday, October 17, 2013

Readings:
Section 8.5
Chapter 14 (skim).
Chapter 15

For next week: Chapter 16.

Problems:

1. Consider the following channel $D$, which is similar to channel $C$ on p. 204, 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, … 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 $C$.
(b) Write pseudocode for automaton $D$.
(c) Prove carefully that $D$ implements $C$, in the sense of inclusion of sets of traces. Use a simulation relation.

2. Exercise 15.11.

3. 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 edges in $G$.

(a) Explain your algorithm in words.
(b) Give pseudocode.
(c) Analyze its message and time complexity.

4. Design a version of the $AsynchBFS$ algorithm in Section 5.4 that allows node $i_0$ to discover when the tree construction has been completed. You should follow the strategy described in the paragraph on “Termination” on p. 503-504.

(a) Explain your algorithm in words.
(b) Give pseudocode.
(c) Analyze its message and time complexity.

5. Exercise 15.30: Design an algorithm to find the shortest paths from a fixed source node $i_0$ to all other nodes in the network. Your algorithm should have a much better time bound than the $AsynchBellmanFord$ algorithm, say $O(n(d + ℓ))$. 
(a) Explain your algorithm in words.
(b) Give pseudocode.
(c) Analyze its time complexity.