# Problem Set 3, Part a

**Due:** Thursday, October 22, 2009

## Reading:

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

**Reading for next week:** Chapter 16.

**Coding Collaboration Policy:** For coding exercises in this assignment, you may collaborate on the design and debugging of code. You must, however, individually write any accompanying text, proofs, and analysis.

**The Required Use of Tempo:** For any problem that asks you to describe an asynchronous algorithm, please provide syntax-checked Tempo code.

## Problems:

1. Exercise 15.34. Try to find the smallest graph (i.e., with the fewest nodes) and the shortest execution (i.e., with the fewest number of messages)that exhibits the behavior described.

2. Consider a channel $D$, which is similar to channel $C$ on p. 204, except that it allows internal message duplication.

   More specifically, 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 I/O automaton pseudocode for automaton $C$, using Tempo. For this and all your Tempo code, please syntax-check using the Tempo language processor.

   (b) Write I/O automaton pseudocode for automaton $D$, using Tempo.

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

3. Exercise 15.7. Write and syntax-check your code using Tempo.

4. (Based on Exercise 15.27) Design an efficient algorithm (state if you are optimizing for message complexity, time complexity, or both) that allows a distinguished process $i_0$ in an asynchronous network based on an arbitrary connected undirected graph $G$ to determine the maximum distance $k$ from $i_0$ to the furthest node in the network. Write the code for this algorithm in Tempo. Analyze its message and time complexity.

5. Exercise 15.30. Use Tempo. You don't need to prove correctness of your algorithm, but you do need to state your time bound, and give a convincing argument that it holds. Try to come up with the best time bound you can.

6.852J / 18.437J Distributed Algorithms

Fall 2009