Distributed Computing, 1998, homework 1.
Due: December 29th.

1. Many distributed algorithms do not have a built-in termination detection mechanism. Given such an algorithm, we have to impose on it a termination detection mechanism, which will signal the nodes of the network that the algorithm has terminated.

   Here is one such mechanism, suggested by J.M. Helary, C. Jard, N. Plouzeau, and M. Raynal from IRISA France:

   Every node v maintains 2 matrices SENT and RECEIVED. Entry $SENT(i,j)$ is the number of messages sent from i to j in the algorithm, to the best of v's knowledge. Entry $RECEIVED(i,j)$ is the number of messages received by j from i in the algorithm, to the best of v's knowledge.

   When ever a node, x, becomes idle (from the underlying algorithm's point of view) it floods the network with a message containing 2 vectors: $SENT(x,\star)$ and $RECEIVED(\star,x)$. Upon receiving a flood message every node updates its matrices accordingly.

   Prove under the following assumptions, that if in some node $SENT(i,j) = RECEIVED(i,j)$ for all pairs (i,j), then there is no message of the underlying algorithm in transit on any link of the network.

   Assume a non-FIFO asynchronous network. Also assume that, magically all the SENT matrices are initialized to the number of messages sent by the spontaneously starting nodes when they start (i.e., the first messages generated by the nodes which wokeup spontaneously to start the underlying algorithm, hence we assume that all start nodes start at the same time). RECEIVED matrices are initialized to 0.

2. There are n identical finite automata connected by n unidirectional, asynchronous links to form a ring, $n > 1$. The same automata work for any size ring. Exactly one automaton receives a WAKE signal from the environment. Design an algorithm which informs the predecessor of the awakened automaton that it is the predecessor of the awakened automaton. Show that your algorithm has asymptotically optimal bit complexity.

   (Hint: The lower bound should hold even for arbitrary identical processors in place of finite automata, and variable length messages, even if the processors know n. Can you relax these restrictions farther?)

   Give a tight upper and lower bound for this problem in the synchronous model.