Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
Mohamad Ahmadi, Hamid Ghodselahi May 04, 2016

Network Algorithms, Summer Term 2016
Problem Set 3

hand in by Wednesday, May 11, 2016

Exercise 1: Leader Election (Message Complexity Improvement)

Let us assume that we have a general n-node graph where each node has a unique identifier in $\{1, \ldots, N\}$. The communication is done in synchronous rounds. To solve the leader election problem we would like to have the node with maximum ID to become the leader.

(a) Propose a time-efficient uniform algorithm to solve the problem with a message complexity of $O(mn)$. What is the time complexity of your algorithm? How can the nodes determine whether the election of a leader has completed?

(b) In Chapter 2, we looked at the flooding algorithm. For this question, assume that there can be more than one source node for the flooding. Show that we can still reach all nodes in time $O(D)$ and with message complexity $O(m)$ if we only need to guarantee that all nodes need to receive the broadcast message of at least one of the source nodes.

(c) We will now use the above flooding algorithm to get a leader election algorithm with better message complexity. Show that it is possible to design a synchronous leader election algorithm which is almost as fast as the algorithm of (a), but which has a message complexity of only $O(m \log N)$. For this question, you can assume that all nodes know the diameter D of the network graph.

Exercise 2: Distributed Computation of the AND

Consider an anonymous ring where each processor has a single bit as input. You can assume that nodes can distinguish between their neighbors, i.e., when a node v receives a message, v knows which neighbor has sent the message (note that nodes may not know a consistent clockwise or counterclockwise orientation of the ring!).

(a) Prove that there is no uniform synchronous algorithm for computing the AND of all input bits.

(b) Present an asynchronous (non-uniform) algorithm for computing the AND; the algorithm should send $O(n^2)$ messages in the worst case.

(c) Present a synchronous (non-uniform) algorithm for computing the AND; the algorithm should send $O(n)$ messages in the worst case. What is the time complexity of your algorithm?