Dr. Lucia Draque Penso
Dr. Jens Maßberg

Institut für Optimierung und Operations Research
Wintersemester 2014/2015

Online and Distributed Algorithms

Exercise Session 2

1. *Log-Star Coloring Algorithm for Rings with Unique Identifiers*: Algorithm 7 colors any tree consisting of n nodes with 3 colors in $O(log^*n)$ rounds. It consists of two phases: In the first phase (Line 2), the initial coloring consisting of all node IDs is reduced to 6 colors, in the second phase (Lines 3-8), the 6 colors are further reduced to 3. Note that, in order to decide when to switch from Phase 1 to Phase 2, the nodes running Algorithm 7 actually count log^*n rounds. However, this is only possible if the nodes are aware of the total number of nodes n. If n is unknown the nodes do not know when the first phase is over: A node v running Algorithm 5 cannot simply decide to be done once its color is in $\mathcal{R} = \{0, \dots 5\}$ since its parent w might still change its color in the future. Even if the color of w is also in $\mathcal{R}$, w might receive a message from its parent that forces w to change its color once more. In the following, we want to overcome this problem, and make Algorithm 7 work even if the nodes are unaware of n, but for the case of a ring with unique identifiers. Assume that nodes in the ring can distinguish between clockwise and counterclockwise.

   (i) Show how the log-star coloring algorithm for trees (Algorithm 7) can be adapted for such rings given that the nodes know n!

   (ii) Now adapt your algorithm so that it also works if the ring nodes do not know n. Preserve the running time of $O(\log^* n)$!

   Hint: You can use additional colors to segment the ring, and switch phases locally.

2. *AND function Distributed Computation in an Anonymous Ring*: Consider an anonymous ring where each node has a single bit as input. Assume that nodes can distinguish between their neighbors, i.e., v knows which neighbor has sent a received message. Note that nodes may not know a consistent clockwise or counterclockwise orientation of the ring.

   (i) In this setting, is there a uniform synchronous deterministic algorithm for computing the AND of all input bits? Either give an algorithm or prove an impossibility result.

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

   (ii) Present in this setting a synchronous non-uniform deterministic algorithm for computing the AND; the algorithm should have $\lceil \frac{n}{2} \rceil$ as time complexity and send $2n$ messages in the worst case. Show that the message complexity can decrease to n in an algorithm with n as time complexity if the nodes can now distinguish between a consistent clockwise or counterclockwise orientation of the ring.

3. *Leader Election in an "Almost" Anonymous Ring*

   (i) Is deterministic leader election possible in a synchronous ring in which all but one nodes have the same identifier? Either give an algorithm or prove impossibility.

   (ii) Consider a synchronous ring in which exactly two nodes have identifier A and all the other nodes have identifier B. Is deterministic leader election possible in this setting? Either give an algorithm or prove an impossibility result. Does it make a difference if the ring is even or odd?

4. *Leader Election in Complete Graphs with Unique Identifiers:* Devise a synchronous deterministic algorithm that just requires $O(n \log n)$ message complexity to the cost of a $O(\log n)$ time complexity, instead of a $O(n^2)$ message complexity with a O(1) time complexity which may be obtained from the trivial algorithm which makes every node send its identifier to every other node in the first round and then deciding on the second one.

   Hint: Make a candidate for leadership communicate to only 2^{k-1} other candidates at round k, when trying to compare its unique identifier. Which candidates should proceed?