Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn

April 22, 2015

Network Algorithms, Summer Term 2015
Problem Set 1

hand in by Tuesday, April 28, 2015

Exercise 1: Vertex Coloring

In the lecture, a simple distributed algorithm ("Reduce") which colors an arbitrary graph with $\Delta + 1$ colors in n synchronous rounds was presented (Δ denotes the largest degree, n the number of nodes of the graph). Each node knows the number of neighbors it has.

1. What is the message complexity, i.e., the total number of messages the algorithm sends in the worst case?

2. Can you make some simple adjustments to the algorithm to reduce the number of messages significantly? What is the amount of messages your algorithm needs to compute a valid coloring?

Exercise 2: Coloring Rings and Trees

Algorithms 4 and 6 in the lecture notes colors any (directed) tree consisting of n nodes with 3 colors in $O(\log^* n)$ rounds. It consists of two phases: In the first phase (Algorithm 4), the initial coloring consisting of all node IDs is reduced to 6 colors, in the second phase (Algorithm 6), 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 the Algorithms 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 4 cannot simply decide to be done once its color is in $\mathcal{R} = \{0, \ldots, 5\}$ 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 (potentially to node v's color!).
In the following, we want to overcome this problem, and make Algorithms 4 and 6 work even if the nodes are unaware of n. To make our lives easier we try to find a solution for the ring topology before we tackle the problem on trees. Formally, a ring is a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{\{v_i, v_j\} \mid j = i+1 \,(\mathrm{mod}\, n)\}$. You can assume that G is a *directed* ring, i.e., nodes can distinguish between "left" and "right".[1]

1. Show how the log-star coloring algorithm for trees can be adapted for rings given that the nodes know n!

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

   Once the color of node v in the ring is in $\mathcal{R}$, it wants to reduce the numbers of colors used to 3. Show how this reduction phase works even if the first phase may not have terminated in some other parts of the ring![2]

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

[1] Note that this assumption is stronger than *sense of direction*, which merely requires that nodes can distinguish their neighbors.

[2] Note that a node cannot wait until it knows that all other nodes are ready to start the second phase as this would require time in the order of n.