Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2011

Prof. R. Wattenhofer
Raphael Eidenbenz

Principles of Distributed Computing
Exercise 1

1 Vertex Coloring

In the lecture, a distributed algorithm ("Reduce") for coloring 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).

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

Hint: Note that the "undecided" messages sent in Line 6 are actually not needed. A node could just as well send no message at all. Therefore neglect these messages in your analysis!

b) Does the algorithm also work in an asynchronous environment? If yes, formulate the asynchronous equivalent to the algorithm, if no, explain why not.

2 Coloring Rings and Trees

Algorithm 7 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 (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 $\mathcal{R}$ 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 Algorithm 7 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 $E = \{\{v_i, v_j\} \mid j = i + 1 \pmod n\}$. You can assume that G is a *directed* ring, i.e., nodes can distinguish between "left" and "right".[1]

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

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

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.

c*) Based on the previous exercise, propose a uniform algorithm that colors any directed tree in $O(\log^* n)$ rounds with at most 3 colors! A distributed algorithm is called *uniform* if it works without the knowledge of the number of nodes n.[2]

[2]Problems marked with an asterisk (*) are hard. Example solutions to these problems will not be provided. However, anybody who solves such a problem will receive a prize!