Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2010
Prof. R. Wattenhofer / C. Lenzen

Principles of Distributed Computing
Exercise 12: Sample Solution

1 Coloring Rings

a) Let $n \geq 4$ be even, and $r = n/2 - 2$. Consider the r-neighborhood graph $\mathcal{N}_r(R_n)$ of the ring R_n with n nodes. Note that for $r = n/2 - 2$ the r-neighborhood of a node contains all but three identifiers, ordered according to their occurence.

The ring can be colored legally with two colors if and only if $\mathcal{N}_r(R_n)$ is bipartite, i.e., the r-neighborhood contains no odd cycle. However, there is one of length $n - 1$:
$(1, \ldots, n-3), (2, \ldots, n-2), (3, \ldots, n-1), (4, \ldots, n-1, 1), (5, \ldots, n-1, 1, 2), \ldots,$
$(n-1, 1, 2 \ldots, n-4), (1, \ldots, n-3)$.
Thus no coloring of the ring with 2 colors is possible in less than $n/2 - 1$ rounds.

b) Each node informs its two neighbors whether it is in the MIS or not and additionally sends its identifier. If node v is in the MIS, it sets its color to 1. If v is not in the MIS but both of its neighbors are, then v sets its color to 2. If v has a neighbor w not in the MIS, v chooses color 2 if its identifier is larger than w's identifier, otherwise v chooses the color 3.

The algorithm only needs one communication round. Correctness follows from the fact that either a node v is in the MIS or at least one of its neigbhors is. Thus, a MIS can at best be computed one round faster than a 3-coloring, which implies that computing a MIS costs at least $(\log^* n)/2 - 2$ rounds.

c) The only modification of the algorithm we need is that the identifiers of both neighbors are considered when computing the new identifier. For this purpose, all nodes must send their identifiers to both neighbors in each round. In the algorithm from the lecture, a node w always considers just one, say, its clockwise neighbor v when computing its new identifier id_w. Node v computes its new identifier id_v based on its old identifier and the identifier of its clockwise neighbor u. In the next round, w uses id_w and id_v to compute its next identifier. In order to execute two rounds in one, we use the following trick. After one communication round, v knows the identifiers of its counterclockwise neighbor w and its clockwise neighbor u. Thus, v has all the information needed to compute the identifier of w after the next round! Instead of executing this additional round, we simply "shift" the identifiers: Node v adopts the identifier that w would get after two rounds in the normal Cole-Vishkin algorithm. If all nodes always adopt the color of their counterclockwise neighbors, we always have a legal coloring and we can in fact compute two rounds in one, reducing the time complexity to $(\log^* n)/2 + O(1)$.

As discussed in Exercise 1, we might run into trouble because nodes do not know when to stop the first phase of the algorithm since n is unknown. However, we may again resolve this problem using some helper colors to avoid conflicts when nodes stop the first phase in different rounds. For further details please refer to the solution of Exercise 1.