ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2010 Prof. R. Wattenhofer / C. Lenzen

Principles of Distributed Computing
Exercise 3: Sample Solution

1 Deterministic distributed algorithms in the port-numbering model

Consider an oriented ring of $n \geq 3$ nodes where we assign at each node portnumber 1 to the outgoing and portnumber 2 to the incoming edge. Thus, all neighborhoods appear identical, rendering any deterministic symmetry breaking impossible.

More precisely, all nodes run the same deterministic algorithm and their initial states are identical to each other. Hence in the first round, whatever message one of the nodes sends to its port 1, all other nodes will send the same message to their port 1; by the same reasoning, the messages sent to port 2 are mutually identical as well. Hence the messages received from port 1 are mutually identical and the same applies to the messages received from port 2. Therefore the local states of the nodes are mutually identical after one communication round. The same reasoning holds for any communication round: if the nodes have identical local states before the round, they will also have dentical local states after the round. In particular, if one of them decides to stop and announce its output, all nodes will do the same and announce the same output.

Therefore, *any* deterministic algorithm must produce the same output at all nodes. This is in conflict with the output specifications of the problems from Parts **a)**, **b)**, **d)** and **e)**. Since the only possible valid output for Part **c)** is 0 (i.e., not in the set) at all nodes, the "approximation ratio" must always be ∞. Finally, to output a vertex cover, all nodes must output 1 (i.e., in the set), while an optimal solution picks every second node.

If randomization is allowed, we do the following. All nodes "flip a coin". There is a positive probability that exactly one node gets head. This node is the leader. Starting with itself, we assign node "identifiers" $1, \ldots, n$, traversing the graph in a depth first search manner. Whenever we encounter an already labeled node, we go back to the node we came from; if this node runs out of edges, we go back to its parent. It is not difficult to see that all edges and thus all nodes will be visited, and once we return to the leader and all of its edges are dealt with, we know that all nodes have been labeled.

Having established distinct identifiers, it is now trivial to collect the graph topology at the leader and deterministically compute an (optimal) solution to any problem that can be solved by computation, in particular Parts **a)** to **f)** from the exercise.

Remark: If n is known, we can modify our approach to ensure success *with high probability*, i.e., with probability $1 - 1/n^c$ for any chosen constant c. Any node picks a random "identifier" of $\lceil (c+1)\log n \rceil$ bits. The maximum of the chosen values is with probability larger than $1 - 2^{-(c+1)\log n} n = 1/n^c$, i.e., with high probability, unique.[1] Now all nodes assume to be the leader and in parallel initiate the above strategy, however, labeling their messages with their "identifier". Once a node learns about a larger value than it currently assumes to be the leader's, it discards all state information and messages regarding the smaller value. Thus, in the end, with high probability only the unique maximum value will "survive".

[1] Two nodes have different values with probability $2^{-\lceil (c+1)\log n \rceil}$. Applying the union bound and the fact that we have n nodes yields the probability bound.

2 Calculations with the $\log^*$ function

Recall that $\log^* x = 1$ if $x \leq 2$ and $\log^* x = 1 + \log^*(\log x)$ if $x > 2$.

a) $k = (W(\Delta!)^\Delta)^\Delta \leq M((M!)^M)^M < M((M^M)^M)^M = M \cdot M^{M^3} = M^{M^3+1} < M^{2M^3}$.

b) Since $M \geq 4$, we have $(2\log M)' = \frac{2}{M} < 1 = M'$ for all possible M. Because for the minimum value of $M = 4$ we have $2\log 4 = 4$, it holds that $2\log M \leq M$ for all $M \geq 4$. We conclude from Part **a)** that $\log k \leq \log\left(M^{2M^3}\right) = 2M^3 \log M \leq M^4$.

c) Applying the logarithm again, we get $\log\log k \leq \log(M^4) = 4\log M$. As $M \geq 4$, $\log M \geq 2$. Similarly to Part **b)**, we see that $2 + \log M \leq M$. Thus, using the definition of $\log^*$, we get that $\log^* k \leq 2 + \log^*(4\log M) \leq 3 + \log^*(\log(4\log M)) \leq 3 + \log^*(2 + \log M) \leq 3 + \log^*(M) \in O(\log^* M)$.

d) Using Part **c)** and that $\log^*$ is non-decreasing, we obtain that $\log^* k \leq 3 + \log^* M = 3 + \max\{\log^* W, \log^* \Delta, \log^* 4\} \in O(\log^* \Delta + \log^* W)$. Finally, certainly we have that $\log^* \Delta \in O(\Delta)$, implying that $\Delta + \log^* k \in O(\Delta + \log^* W)$.