University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, A. Mályusz

Theory of Distributed Systems
Sample Solution Exercise Sheet 8

Due: Wednesday, 19th of June 2024, 12:00 noon

Exercise 1: 2-coloring in paths *(7 Points)*

Show that there is no randomized distributed algorithm that finds a 2-coloring in paths in $o(n)$ rounds with probability at least 0.9. Assume that n is known and IDs are from $\{1, \ldots, n\}$.
Hint: use that the principle of locality can be extended to randomized algorithms i.e. for any possible output, nodes with the same view have the same probability of giving that output.

Sample Solution

We modify the deterministic lower bound seen in the lecture as follows. Assume that there exists a randomized algorithm that solves 2-coloring in $o(n)$ with success probability at least 0.9 , this implies that for large enough n the algorithm runs in less than $n/5$ rounds. We consider two instances. The first has IDs ordered from 1 to n, while in the second we move node n such that it becomes neighbor of nodes with ID $n/5 + 1$ and $n/5 + 2$. For large enough n, nodes 1 and $n/2 + 1$ have the same view in both instances, and this implies that they give the same output in both instances with the same probability distribution. In particular, let $view_1$ be the view of node 1 and $view_2$ be the view of node $n/2 + 1$. We have that:

$$\begin{aligned}
&\Pr[\text{ node 1 outputs White } \mid view_1] = x \\
&\Pr[\text{ node 1 outputs Black } \mid view_1] = 1 - x \\
&\Pr[\text{ node } n/2 + 1 \text{ outputs White } \mid view_2] = y \\
&\Pr[\text{ node } n/2 + 1 \text{ outputs Black } \mid view_2] = 1 - y
\end{aligned}$$

for some values of x and y. Note also that the outputs of nodes 1 and $n/2 + 1$ are independent. Since in the first instance nodes 1 and $n/2 + 1$ have even distance, they must give the same color, hence the probability that they give different colors must be at most 0.1 (otherwise the success probability would be less than 0.9). Hence, we get that

$$x(1 - y) + (1 - x)y \leq 0.1$$

The same nodes in the second instance must give different colors, hence the probability that they give the same colors must be at most 0.1. Hence, we get that

$$xy + (1 - x)(1 - y) \leq 0.1$$

Hence, we get the following system of inequalities:

$$\begin{cases} x(1 - y) + (1 - x)y \leq 0.1 \\ xy + (1 - x)(1 - y) \leq 0.1 \end{cases}$$

Notice that such system has no solution, else we would end up with $0.9 \leq 0.1$.

Exercise 2: Independent sets in paths *(9 Points)*

An independent set (IS) is a subset of nodes such that no two neighboring nodes are in the independent set. A maximal independent set (MIS) is an independent set that cannot be extended. Assume that n is known and IDs are from $\{1, \dots, n\}$. Show that, in paths:

1. it is trivial to find some IS in $O(1)$ time with a deterministic distributed algorithm.
2. there exists an IS with at least $n/2$ nodes.
3. it is not possible to find an IS of size at least $n/2$ in $o(n)$ rounds.
4. there is no deterministic distributed algorithm that finds an MIS in $o(\log^* n)$ rounds.

Bonus: a *vertex cover* of a graph is a subset of nodes that includes at least one endpoint of every edge of the graph. Deduce from number 3 above that it is not possible in paths to find a vertex cover of size at most $n/2$ in $o(n)$ rounds.

Sample Solution

1. All nodes, in 0 rounds, output not in the IS. Another solution would be in $O(1)$ rounds, nodes can check if they are local minima and hence join the IS else they don't.
2. Start from one arbitrary endpoint v of the path and put it in the IS, then proceed by putting in the IS nodes that are at even distance from v.
3. From the lower bound seen in the lecture we know that 2 coloring is hard even on instances where n is odd. Notice that, if n is odd, an IS of size at least $n/2$ forms a 2-coloring, one can prove that via induction (if n is even it may not necessarily be the case). Hence a fast algorithm for this problem would give a fast way to solve 2-coloring in paths having odd size, that is a contradiction.
4. It is enough to show that in constant time it is possible to convert a solution for MIS into a solution for 3-coloring (a $o(\log^* n)$ algorithm for MIS would imply a $o(\log^* n)$ algorithm for 3-coloring, that from the lecture we know it is a contradiction). We can convert an MIS into a 3-coloring as follows:
    - Nodes in the MIS take color 1.
    - Nodes not in the MIS either have two neighbors in the MIS, or one neighbor in the MIS and one outside (hence nodes not in the MIS for components of size at most two). In the first case a node takes color 2. In the other, a node not in the MIS that has ID smaller than the neighbor that is also not in the MIS takes color 2, wile the one that has larger ID takes color 3.

Bonus: notice that a set of nodes $C \subseteq V(G)$ of a graph G is a vertex cover if and only if $V(G) \setminus C$ is an IS. From a vertex cover of size k, in 0 round, one can find an IS of size $n - k$; hence, a $o(n)$ algorithm for finding a vertex cover of size at most $n/2$ would imply a $o(n)$ algorithm for finding an IS of size at least $n/2$, which is a contradiction due to number 3 above.

Exercise 3: Counting *(4 Points)*

Assume we are given a path of size n where nodes know an upper bound on the size of the network in $\{n, \dots, cn + c'\}$ for some constants c, c' (i.e., nodes do not necessarily know the exact value of n but only e.g. a constant approximation). Show that there is no deterministic distributed algorithm that counts the number of nodes in paths in $o(n)$ rounds, where at the end of the algorithm every node needs to output this count. Assume that IDs are from $\{1, \dots, n\}$.

Sample Solution

Consider a path with n nodes that have IDs that are ordered from 1 to n. Assume that, as an upper bound of n, we give the value $n+1$. For large enough n, nodes must terminate in at most $n/100$ and output n. In particular, node 1 must output n. Consider now a path with $n+1$ nodes that have IDs that are ordered from 1 to $n+1$. Assume that, as an upper bound of the size of the network, we give the exact value $n+1$. Node 1 in both instances has the same view (and the same upper bound $n+1$ on the size of the network is given), so it must output the same as before, hence n, that is wrong for this second instance.