University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, A. Mályusz, Z. Parsaeian

Theory of Distributed Systems
Sample Solution Exercise Sheet 8

Exercise 1: Matching

A *matching* of a graph $G = (V, E)$ is a subset of edges $M \subseteq E$ such that no two edges in M are adjacent. A matching is maximal if no edge can be added without violating this property.

Give an algorithm that computes a maximal matching (MM) in $O(\log n)$ rounds w.h.p. in the synchronous message passing model. That is, after the algorithm terminates each node needs to know which of its adjacent edges are part of the maximal matching.
Hint: Try to construct a new graph G', such that solving MIS on G' gives a solution for MM in G.

Sample Solution

Let $G = (V, E)$ be the graph for which we want to construct the matching. The so-called line graph G' is defined as follows: for every edge in G there is a node in G'; Two nodes in G' are connected by an edge if their respective edges in G are adjacent. G' can be simulated on G with constant overhead. A (maximal) independent set in the line graph G is a (maximal) matching in the original graph G, and vice versa. If G has n nodes, G' has at most n^2 nodes and so we can compute an MIS on G' in time $\mathcal{O}(\log n^2) = \mathcal{O}(\log n)$.

Exercise 2: Coloring

Assume we have $C = \alpha(\Delta + 1) \in \mathbb{N}$ colors for some $\alpha \geq 1$. Consider the following algorithm in the synchronous message passing model to color the graph with C colors. Each node v repeats the following steps (corresponding to a phase) until it has a color:

- Let U_v be the set of yet uncolored neighbors of v and let C_v be the set of colors that v's neighbors already chose (initially C_v are all of v's neighbors and $C_v = \emptyset$).
- Node v picks a random number $r_c(v) \in [0, 1]$ for every remaining color $c \in \{1, \dots, C\} \setminus C_v$ and informs its neighbors about those numbers.
- If $r_c(v) < r_c(u)$ for some $c \in \{1, \dots, C\} \setminus C_v$ and every $u \in U_v$, then v colors itself with c, informs its neighbors and terminates (if this holds for several c, node v decides on the *'smallest'* of these colors).

(a) What is the probabillity that for some fixed color $c \in \{1, \dots, C\} \setminus C_v$, $r_c(v) < r_c(u)$ for each uncolored neighbor u of v?

(b) Show that the probability that a node obtains a color in a given phase is at least $1 - e^{-\alpha}$.

(c) Show that the algorithm terminates after $\mathcal{O}\big(1 + \frac{\log n}{\alpha}\big)$ rounds in expectation.

(d) How should we choose the domain of available colors C to get an algorithm that terminates in a constant number of phases in expectation?

(e) We now want to show that after $\mathcal{O}\big(\log n\big)$ phases the algorithm terminates even with high probability. More concrete, show that after $\lceil 2 \cdot \ln n \rceil$ phases, the probability that each node is colored is at least $1 - n^{-\alpha}$.

Sample Solution

(a) Note that $r_c(v) < r_c(u)$ for all $u \in U_v$ if in the list of all these $|U_v| + 1$ values of r_c the value of v is the smallest one. Clearly, this list has $(|U_v| + 1)!$ potential permutations and in $|U_v|!$ of them is $r_c(v)$ the smallest element. Hence, the probablilty for the statment is

$$\frac{|U_v|!}{(|U_v|+1)!} = \frac{|U_v|!}{|U_v|! \cdot (|U_v+1|)} = \frac{1}{|U_v|+1}.$$

(b) The algorithm uses a similar techniques as for computing an MIS seen in the lecture (Luby's algorithm). For each color we compute an independent set. A node joins one of those independent sets by having the smallest random number.

Note that in any phase it holds that $|U_v| + |C_v| \leq deg(v) \leq \Delta$ and thus $|C_v| \leq \Delta - |U_v|$. Then the number of colors that v can still compete for is

$$C - |C_v| \geq \alpha(\Delta+1) - (\Delta - |U_v|) = (\alpha-1)\Delta + \alpha + |U_v| \geq (\alpha-1)|U_v| + |U_v| + \alpha = \alpha(|U_v|+1)$$

The chance for v to join some remaining color c is at least $\frac{1}{|U_v|+1}$ (by task (a)). The probability that a node does not get any color in a fixed phase is at most

$$(1 - \tfrac{1}{|U_v|+1})^{\alpha(|U_v|+1)} = \left((1 - \tfrac{1}{|U_v|+1})^{(|U_v|+1)}\right)^{\alpha} \leq \left(\tfrac{1}{e}\right)^{\alpha}.$$

(c) By the previous statment we know that a node remains uncolored with probabillity $\leq e^{-\alpha}$. Thus, the expected number of nodes remaining uncolored after i iterations is $\leq n \cdot \prod_{j=1}^{i} e^{-\alpha} = n \cdot e^{-\alpha i}$. Let's define i_T as the iteration where we expect at most 1 node remains uncolored.

$$\begin{aligned} & n \cdot e^{-i_T \cdot \alpha} \leq 1 \\ \iff & \ln n - i_T \cdot \alpha \leq 0 \\ \iff & \tfrac{\ln n}{\alpha} \leq i_T \end{aligned}$$

So we expect that there is at most one color left after $\lceil \frac{\ln n}{\alpha} \rceil$ phases. Coloring the remaining node takes at most one additional round, so the expected number of rounds is $1 + \lceil \frac{\ln n}{\alpha} \rceil$.

(d) We can choose $C = \lceil \ln n \cdot \Delta \rceil$ for example.

(e) The probabillity that some fixed node v is not colored after $\lceil 2 \cdot \ln n \rceil$ phases is by task (b) at least the following:

$$\prod_{i=1}^{\lceil 2 \cdot \ln n \rceil} e^{-\alpha} \leq e^{-2\alpha \cdot \ln n} = \left(\frac{1}{n}\right)^{2\alpha}$$

For termination all n nodes have to be colored. So, the probabillity that there exists at least one node that is not colored after that many iterations can be bounded by a union bound as written subsequently:

$$n \cdot \left(\frac{1}{n}\right)^{2\alpha} = n^{1-2\alpha} \leq n^{-\alpha}$$