Model Solutions to Exercises of Week 9

Juho Hirvonen

September 11, 2015

Exercise 9.1

a) Both endpoints of the path send a propagation towards the middle, colouring each edge in turn with colours 1 and 2. If the path has an even number of nodes, these propagations cross in the middle using the same colour. Nodes in the middle send messages back to the leaves telling all nodes to stop. If the path has an odd number of nodes, these propagations collide at the middle node, both incident edges of that node having the same colour proposal. In this case, we use the port numbers to break symmetry: the node in the middle sends a propagating stop message to port 1 and sends a new propagation to port 2, colouring the edges in with the reversed colours. See Figure 1 for illustration.

   The messages propagate from the leaves to the middle and back, so the algorithm runs in time $O(n)$.

b) We could use the same technique to prove both 9.1b and 9.1c, but we will need different techniques 9.2b and 9.1c, so we will prove these separately.

   Consider a path $G = (V, E)$ such that $V = \{v_1, v_2, \ldots, v_{2k}\}$, that is, G has odd length. Let $N = (G, P, p)$ with $p(v, 1) = (u, 2)$ for all non-leaf nodes. This can

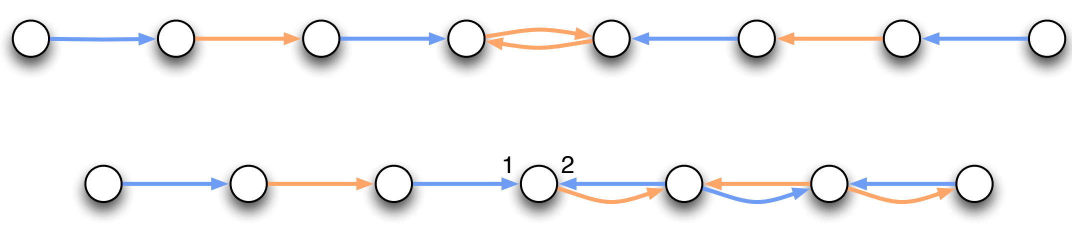

Figure 1: Illustration for the Exercise 9.1a. First a path of even length and then a path of odd length.

be achieved by starting from the, say, left end of the path and port-numbering each edge with 1 and 2 from left to right. Now consider the nodes v_k and v_{k+1}. We wanto to apply Theorem 9.1 of the course material. It is easy to see that if the radius-r neighbourhoods of the nodes v_k and v_{k+1} do not contain the endpoints of the path, then they must be isomorphic. This happens exactly when $r \leq k - 1 = \frac{n}{2} - 1$. By Theorem 9.1 the outputs of v_k and v_{k+1} must be the same. Let us assume, without loss of generality, that v_k colours the edge in its port 1 with colour 1 and the edge in its port 2 with colour 2. Now v_{k+1} must also colour the edge in its port 2 with colour 2, but now this edge is coloured with both colours 1 and 2, so the output of the algorithm does not correctly encode any solution. Thus we have shown that any distributed algorithm needs at least $n/2$ rounds and that therefore $o(n)$ rounds is not enough.

c) Consider two paths $G_n = (V, E)$ and $G_{n+1} = (V \cup \{w\}, E')$ with $|V| = n = 2k$. Let $V = \{v_1, v_2, \ldots, v_n\}$. We set

$$E = \big\{\{v_1, v_2,\}, \{v_2, v_3\}, \ldots, \{v_i, v_{i+1}\}, \ldots, \{v_{n-1}, v_n\}\big\}.$$

The set E' is almost as E, except $E' = (E \setminus \{\{v_k, v_{k+1}\}\}) \cup \{\{v_k, w\}, \{w, v_{k+1}\}\}$. We set unique identifiers so that $f(v_i) = i$ both in G_n and G_{n+1}, and $f(w) = n + 1$. Now we can apply Theorem 9.1. Since G_n has odd length, any 2-edge colouring has both $\{v_1, v_2\}$ and $\{v_{n-1}, v_n\}$ coloured with the same colour. In comparison, G_{n+1} has even length, so that in any 2-edge colouring $\{v_1, v_2\}$ and $\{v_{n-1}, v_n\}$ have different colours. Assuming we have an algorithm that stops on G_n in $r \leq n/2 - 1$ rounds, both v_1 and v_n have isomorphic radius-r neighbourhoods in G_n and G_{n+1}. By the previous observation, at least one of them must change its output between G_n and G_{n+1}, but by Thorem 9.1 this is impossible in r rounds. Thus $o(n)$ rounds is not enough.

Exercise 9.2

a) We can use the edge colouring algorithm from 9.1a: it is easy to see that both edge colour classes form a maximal matching.

b) We use the proof from 9.1b: the nodes v_k and v_{k+1} must still produce the same output, so w.l.o.g. both output 1, which is not a proper encoding of a solution, or both output empty, which is not a feasible solution.

c) We find a 3-colouring of the path in $O(\log^* n)$ rounds using the techniques presented already in Chapter 1 of the book (and Exercise 1.3). Given a 3-colouring, run the bipartite maximal matching algorithm from Chapter 4, with nodes of colour 1 as white nodes and the rest as black nodes, then with nodes of colour 2 as white nodes and nodes of colour 3 as black nodes. This was also presented in Exercise 4.4.

Exercise 9.5

a) Every node v sends its initial colour $c(v)$ to its neighbours. Then, if $c(v) = 4$, the node v picks a new colour $c'(v) = \min\{1, 2, 3\} \setminus \{c(u) : u \in N(v)\}$. The minimum $c'(v)$ always exists, as $|\{c(u) : u \in N(v)\}| \leq 2$. The algorithm clearly runs in one round.

b) Nodes decide their outputs based on only their own initial colour in the 4-colouring. Any such algorithm can be described as a mapping A from $\{1, 2, 3, 4\}$ to $\{1, 2, 3\}$. It is thus clear, that there must be at least two colours c_1 and c_2 such that $A(c_1) = A(c_2) = c_3 \in \{1, 2, 3\}$. Now we simply consider a graph where two adjacent nodes u and v have initial colours $f(v) = c_1$ and $f(u) = c_2$.

c) We can use a very similar algorithm to the 2-edge colouring algorithm from Exercise 9.1. Denote the path of length n by $G_n = (V_n, E)$, where $V_n = \{v_1, v_2, \ldots v_n\}$ and $E = \{\{v_i, v_{i+1}\} : i \leq n - 1\}$. Each leaf colours itself with colour one and then sends a propagation of alternating colours two and one towards the centre. If the path has odd length, these propagations will be sent over the middle edge $\{v_{n/2}, v_{n/2+1}$ on round $n/2$. We can use the 4-colouring to break symmetry: the node with the smaller colour keeps its colour. The node with the larger colour sends back towards the leaf a new propagation that changes the colours. If the path has even length, the algorithm works, and the node that sees the propagations collide simply sends a new propagation telling everyone to stop.

d) We can reuse the argument from 9.1c. We consider the paths G_n and G_{n+1} where the 4-colouring f agrees between G_n and G_{n+1} on nodes $v_1, v_2, \ldots, v_n$ and $f(v_{n+1}) \in \{1, 2, 3, 4\} \setminus \{f(v_{n/2}), f(v_{n/2+1})\}$. Following the argument from 9.1c, we see that if a distributed algorithm runs in time $r \leq n/2$ on G_n, then it also stops in time r on G_{n+1}. On G_n (with n even) the outputs of v_1 and v_n must differ. On G_{n+1}, however, they must be the same. By Theorem 9.1 this is impossible.