Chapter 8
Local Neighborhoods

Covering maps can be used to argue that a given problem cannot be solved at all with deterministic PN algorithms. Now we will study the concept of locality, which can be used to argue that a given problem cannot be solved fast, in any model of distributed computing.

8.1 Definitions

Let \( N = (V, P, p) \) and \( N' = (V', P', p') \) be simple port-numbered networks, with the underlying graphs \( G = (V, E) \) and \( G' = (V', E') \). Fix the local inputs \( f : V \rightarrow Y \) and \( f' : V' \rightarrow Y \), a pair of nodes \( v \in V \) and \( v' \in V' \), and a radius \( r \in \mathbb{N} \). Define the radius-\( r \) neighborhoods

\[
U = \text{ball}_G(v, r), \quad U' = \text{ball}_{G'}(v', r).
\]

We say that \( (N, f, v) \) and \( (N', f', v') \) have isomorphic radius-\( r \) neighborhoods if there is a bijection \( \psi : U \rightarrow U' \) with \( \psi(v) = v' \) such that

(a) \( \psi \) preserves degrees: \( \deg_N(v) = \deg_{N'}(\psi(v)) \) for all \( v \in U \).

(b) \( \psi \) preserves connections and port numbers: \( p(u, i) = (v, j) \) if and only if \( p'(\psi(u), i) = (\psi(v), j) \) for all \( u, v \in U \).

(c) \( \psi \) preserves local inputs: \( f(v) = f'(\psi(v)) \) for all \( v \in U \).

The function \( \psi \) is called an \( r \)-neighborhood isomorphism from \( (N, f, v) \) to \( (N', f', v') \). See Figure 8.1 for an example.
Figure 8.1: Nodes $u$ and $v$ have isomorphic radius-2 neighborhoods, provided that we choose the port numbers appropriately. Therefore in any algorithm $A$ the state of $u$ equals the state of $v$ at time $t = 0, 1, 2$. However, at time $t = 3, 4, \ldots$ this does not necessarily hold.

8.2 Local Neighborhoods and Executions

**Theorem 8.1.** Assume that

(a) $A$ is a deterministic PN algorithm with $X = \text{Input}_A$,

(b) $N = (V, P, p)$ and $N' = (V', P', p')$ are simple port-numbered networks,

(c) $f : V \to X$ and $f' : V' \to X$ are arbitrary functions,

(d) $v \in V$ and $v' \in V'$,

(e) $(N, f, v)$ and $(N', f', v')$ have isomorphic radius-$r$ neighborhoods.

Let

(f) $x_0, x_1, \ldots$ be the execution of $A$ on $(N, f)$, and

(g) $x'_0, x'_1, \ldots$ be the execution of $A$ on $(N', f')$.

Then for each $t = 0, 1, \ldots, r$ we have $x_t(v) = x'_t(v')$.

**Proof.** Let $G$ and $G'$ be the underlying graphs of $N$ and $N'$, respectively. We will prove the following stronger claim by induction: for each $t = 0, 1, \ldots, r$, we have $x_t(u) = x'_t(\psi(u))$ for all $u \in \text{ball}_G(v, r - t)$.

To prove the base case $t = 0$, let $u \in \text{ball}_G(v, r)$, $d = \text{deg}_N(u)$, and $u' = \psi(u)$; we have

$$x'_0(u') = \text{init}_{A,d}(f'(u')) = \text{init}_{A,d}(f(u)) = x_0(u).$$
For the inductive step, assume that $t \geq 1$ and $u \in \text{ball}_G(v, r - t)$.

Let $u' = \psi(u)$. By inductive assumption, we have

$$x_{t-1}^'(u') = x_{t-1}(u).$$

Now consider a port $(u, i) \in P$. Let $(s, j) = p(u, i)$. We have $(s, u) \in E$, and therefore

$$\text{dist}_G(s, v) \leq \text{dist}_G(s, u) + \text{dist}_G(u, v) \leq 1 + r - t.$$

Define $s' = \psi(s)$. By inductive assumption we have

$$x_{t-1}^'(s') = x_{t-1}(s).$$

The neighborhood isomorphism $\psi$ preserves the port numbers: $(s', j) = p'(u', i)$. Hence all of the following are equal:

(a) the message sent by $s$ to port $j$ on round $t$,
(b) the message sent by $s'$ to port $j$ on round $t$,
(c) the message received by $u$ from port $i$ on round $t$,
(d) the message received by $u'$ from port $i$ on round $t$.

As the same holds for any port of $u$, we conclude that

$$x^t_t(u') = x^t_t(u).$$

To apply Theorem 8.1 in the LOCAL model, we need to include unique identifiers in the local inputs $f$ and $f'$.

### 8.3 Example: 2-Coloring Paths

We know from Chapter 1 that one can find a proper 3-coloring of a path very fast, in $O(\log^* n)$ rounds. Now we will show that 2-coloring is much harder; it requires linear time.
To reach a contradiction, suppose that there is a deterministic distributed algorithm $A$ that finds a proper 2-coloring of any path graph in $o(n)$ rounds in the LOCAL model. Then there has to be a number $n_0$ such that for any number of nodes $n \geq n_0$, the running time of algorithm $A$ is at most $(n-3)/2$. Pick some integer $k \geq n_0/2$, and consider two paths: path $G$ contains $2k$ nodes, with unique identifiers $1, 2, \ldots, 2k$, and path $H$ contains $2k + 1$ nodes, with unique identifiers $1, 2, \ldots, k, 2k + 1, k + 1, k + 2, \ldots, 2k$.

Here is an example for $k = 3$:

**G:**

```
1 2 3 4 5 6
```

**H:**

```
1 2 3 7 4 5 6
```

We assign the port numbers so that for all degree-2 nodes port number 1 points towards node 1:

**G:**

```
1 1 1 2 1 3 2 1 4 2 1 5 2 1 6
```

**H:**

```
1 1 1 2 1 3 2 1 7 2 1 4 2 1 5 2 1 6
```

By assumption, the running time of $A$ is at most

$$(n-3)/2 \leq (2k + 1 - 3)/2 = k - 1$$

rounds in both cases. Since node 1 has got the same radius-$(k-1)$ neighborhood in $G$ and $H$, algorithm $A$ will produce the same output for node 1 in both networks:

**G:**

```
1 1 1 2 1 3 2 1 4 2 1 5 2 1 6
```

**H:**

```
1 1 1 2 1 3 2 1 7 2 1 4 2 1 5 2 1 6
```
By a similar reasoning, node $2k$ (i.e., the last node of the path) also has to produces the same output in both cases:

$$\begin{align*}
G: & \quad 1 \quad 1 \quad 2 \quad 1 \quad 3 \quad 2 \quad 1 \quad 4 \quad 2 \quad 1 \quad 5 \quad 2 \quad 1 \quad 6 \\
H: & \quad 1 \quad 1 \quad 2 \quad 1 \quad 3 \quad 2 \quad 1 \quad 7 \quad 2 \quad 1 \quad 4 \quad 2 \quad 1 \quad 5 \quad 2 \quad 1 \quad 6
\end{align*}$$

However, now we reach a contradiction. In path $H$, in any proper 2-coloring nodes 1 and $2k$ have the same color—for example, both of them are of color 1, as shown in the following picture:

$$H: \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1$$

If algorithm $A$ works correctly, it follows that nodes 1 and $2k$ must produce the same output in path $H$. However, then it follows that nodes 1 and $2k$ produces the same output also in $G$, too, but this cannot happen in any proper 2-coloring of $G$.

$$\begin{align*}
G: & \quad 1 \quad ? \quad ? \quad ? \quad ? \quad ? \quad 1 \\
H: & \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1
\end{align*}$$

We conclude that algorithm $A$ fails to find a proper 2-coloring in at least one of these instances.

### 8.4 Quiz

The caption of Figure 8.1 says: “However, at time $t = 3, 4, \ldots$ this does not necessarily hold.” Design some deterministic algorithm $A$ that demonstrates this for the PN model. When you run algorithm $A$ in the network of Figure 8.1, the following should happen:

- At time $t = 0, 1, 2$, nodes $u$ and $v$ have the same local state.
- At time $t = 3, 4, \ldots$, nodes $u$ and $v$ have different local states.
Please try to design your algorithm so that its behavior does not depend on the choice of the port numbering: for each node $x$ and each time step $t$, the state of $x$ has to be the same regardless of how you choose the port numbers. A short, informal description of the algorithm is enough.

### 8.5 Exercises

**Exercise 8.1** (edge coloring). In this exercise, the graph family $F$ consists of path graphs.

(a) Show that it is possible to find a 2-edge coloring in time $O(n)$ with deterministic PN-algorithms.

(b) Show that it is not possible to find a 2-edge coloring in time $o(n)$ with deterministic PN-algorithms.

(c) Show that it is not possible to find a 2-edge coloring in time $o(n)$ with deterministic LOCAL-algorithms.

**Exercise 8.2** (maximal matching). In this exercise, the graph family $F$ consists of path graphs.

(a) Show that it is possible to find a maximal matching in time $O(n)$ with deterministic PN-algorithms.

(b) Show that it is not possible to find a maximal matching in time $o(n)$ with deterministic PN-algorithms.

(c) Show that it is possible to find a maximal matching in time $o(n)$ with deterministic LOCAL-algorithms.

**Exercise 8.3** (optimization). In this exercise, the graph family $F$ consists of path graphs. Can we solve the following problems with deterministic PN-algorithms? If yes, how fast? Can we solve them any faster in the LOCAL model?

(a) Minimum vertex cover.

(b) Minimum dominating set.
(c) Minimum edge dominating set.

**Exercise 8.4** (approximation). In this exercise, the graph family $\mathcal{F}$ consists of path graphs. Can we solve the following problems with deterministic PN-algorithms? If yes, how fast? Can we solve them any faster in the LOCAL model?

(a) 2-approximation of a minimum vertex cover?

(b) 2-approximation of a minimum dominating set?

**Exercise 8.5** (auxiliary information). In this exercise, the graph family $\mathcal{F}$ consists of path graphs, and we are given a 4-coloring as input. We consider deterministic PN-algorithms.

(a) Show that it is possible to find a 3-coloring in time 1.

(b) Show that it is not possible to find a 3-coloring in time 0.

(c) Show that it is possible to find a 2-coloring in time $O(n)$.

(d) Show that it is not possible to find a 2-coloring in time $o(n)$.

* **Exercise 8.6** (orientations). In this exercise, the graph family $\mathcal{F}$ consists of cycle graphs, and we are given some orientation as input. The task is to find a consistent orientation, i.e., an orientation such that both the indegree and the outdegree of each node is 1.

(a) Show that this problem cannot be solved with any deterministic PN-algorithm.

(b) Show that this problem cannot be solved with any deterministic LOCAL-algorithm in time $o(n)$.

(c) Show that this problem can be solved with a deterministic PN-algorithm if we give $n$ as input to all nodes. How fast? Prove tight upper and lower bounds on the running time.
Exercise 8.7 (local indistinguishability). Consider the graphs $G_1$ and $G_2$ illustrated in Figure 8.2. Assume that $A$ is a deterministic PN-algorithm with running time 2. Show that $A$ cannot distinguish between nodes $v_1$ and $v_2$. That is, there are simple port-numbered networks $N_1$ and $N_2$ such that $N_i$ has $G_i$ as the underlying graph, and the output of $v_1$ in $N_1$ equals the output of $v_2$ in $N_2$.

\[\text{Hint A}\]

8.6 Bibliographic Notes

Local neighborhoods were used to prove negative results in the context of distributed computing by, e.g., Linial [1].

8.7 Bibliography


8.8 Hints

A. Argue using both covering maps and local neighborhoods. For $i = 1, 2$, construct a network $N'_i$ and a covering map $\phi_i$ from $N'_i$ to $N_i$. Let $v'_i \in \phi_i^{-1}(v_i)$. Show that $v'_1$ and $v'_2$ have isomorphic radius-2 neighborhoods; hence $v'_1$ and $v'_2$ produce the same output.
Then use the covering maps to argue that $v_1$ and $v_2$ also produce the same outputs. In the construction of $N_1'$, you will need to eliminate the 3-cycle; otherwise $v_1'$ and $v_2'$ cannot have isomorphic neighborhoods.