Principles of Distributed Computing Lecture 04 *Lecturer: Mohsen Ghaffari* *Scribe:*

1 Locality-Based Lower Bound for Coloring

We have seen an $O(\log^* n)$-round algorithm for 3-coloring rooted trees, in the previous lectures. In this lecture, we prove that this bound is tight. Before that, let us observe and discuss that for 2-coloring, we would need $\Omega(n)$ rounds.

Observation 1. *Any* LOCAL *algorithm for* 2*-coloring an* n*-node directed path requires at least* $\Omega(n)$ *rounds.*

Proof. (Proof Sketch) Notice that the two nodes at the end of the part will have to have the same color or different colors, depending on whether the path has odd or even length. Figuring that out requires $\Omega(n)$ rounds as information propagates at the speed of most one hop per round. □

Next, we prove that any deterministic algorithm for 3-coloring rooted trees requires at least $\frac{1}{2}\log^* n - O(1)$ rounds. This result was first proved by Linial [Lin87, Lin92]. We explain a streamlined proof based on [LS14]. The lower bound holds also for randomized algorithms [Nao91], but we will not cover that generalization, for the sake of simplicity. Furthermore, essentially the same lower bound can be obtained as a direct corollary of *Ramsey Theory*. We will have a brief explanation about that, at the end.

Theorem 2. *Any deterministic algorithm for* 3*-coloring* n*-node directed paths needs at least* $\frac{\log^* n}{2} - 2$ *rounds.*

For the sake of contradiction, suppose that there is an algorithm $\mathcal{A}$ that computes a 3-coloring of any n-node directed path in t rounds for $t < \frac{\log^* n}{2} - 2$. When running this algorithm for t rounds, any node v can see at most the k-neighborhood around itself for $k = 2t + 1$, that is, the vector of identifiers for the nodes up to t hops before itself and up to t hops after itself. Hence, if the algorithm $\mathcal{A}$ exists, there is a mapping from each such neighborhood to a color in $\{1, 2, 3\}$ such that neighborhoods that can be conceivably adjacent are mapped to different colors.

We next make this formal by a simple and abstract definition. For simplicity, we will consider only a restricted case of the problem where the identifiers are set monotonically increasing along the path. Notice this restriction only strengthens the lower bound, as it shows that even for this restricted case, there is no t-round algorithm for $t < \frac{\log^* n}{2} - 2$.

Definition 3. *We say* B *is a* k*-ary* q*-coloring if for any set of identifiers* $1 \le a_1 < a_2 < \cdots < a_k < a_{k+1} \le n$*, we have the following two properties:*

P1: $B(a_1, a_2, \ldots, a_k) \in \{1, 2, \ldots, q\}$,

P2: $B(a_1, a_2, \ldots, a_k) \neq B(a_2, \ldots, a_{k+1})$.

Observation 4. *If there exists a deterministic algorithm* $\mathcal{A}$ *for* 3*-coloring* n*-node directed paths in* $t < \frac{\log^* n}{2} - 2$ *rounds, then there exists a* k*-ary* 3*-coloring* B*, where* $k = 2t + 1 < \log^* n - 3$*.*

Proof. Suppose that such an algorithm $\mathcal{A}$ exists. We then produce a k-ary 3-coloring B by examining $\mathcal{A}$. For any set of identifiers $1 \le a_1 < a_2 < \cdots < a_k \le n$, define $B(a_1, a_2, \ldots, a_k)$ as follows. Simulate algorithm $\mathcal{A}$ on an imaginary directed path where a consecutive portion of the identifiers on the path are set equal to $a_1, a_2, \ldots, a_k$. Then, let $B(a_1, a_2, \ldots, a_k)$ be equal to the color in $\{1, 2, 3\}$ that the node a_{t+1} receives in this simulation.

We now argue that B as defined above is a k-ary 3-coloring. Property P1 holds trivially. We now argue that property P2 also holds. For the sake of contradiction, suppose that it does not, meaning that there exist a set of identifiers $1 \le a_1 < a_2 < \cdots < a_k < a_{k+1} \le n$ such that $B(a_1, a_2, \ldots, a_k) = B(a_2, \ldots, a_{k+1})$. Then, imagine running algorithm $\mathcal{A}$ on an imaginary directed path where a consecutive portion of identifiers are set equal to $a_1, a_2, \ldots, a_{2t+2}$. Then, since $B(a_1, a_2, \ldots, a_k) = B(a_2, \ldots, a_{k+1})$, the algorithm $\mathcal{A}$ assigns the same color to a_{t+1} and a_{t+2}. This is in contradiction with $\mathcal{A}$ being a 3-coloring algorithm. □

To prove Theorem 2, we show that a k-ary 3-coloring B where $k < \log^* n - 3$ cannot exist. The proof is based on the following two lemmas:

Lemma 5. *There is no* 1*-ary* q*-coloring with* $q < n$*.*

Proof. A 1-ary q-coloring requires that $B(a_1) \neq B(a_2)$, for any two identifiers $1 \leq a_1 < a_2 \leq n$. By the Pigeonhole principle, this needs $q \geq n$. □

Lemma 6. *If there is a* k*-ary* q*-coloring* B*, then there exists a* $(k-1)$*-ary* 2^q*-coloring* B'*.*

Proof. For any set of identifiers $1 \leq a_1 < a_2 < \cdots < a_{k-1} \leq n$, define $B'(a_1, a_2, \ldots, a_{k-1})$ to be the set of all possible colors $i \in \{1, \ldots, q\}$ for which $\exists a_k > a_{k-1}$ *such that* $B(a_1, a_2, \ldots, a_{k-1}, a_k) = i$.

Notice that B' is a subset of $\{1, \ldots, q\}$. Hence, it has 2^q possibilities, which means that B' has property P1 and it assigns each set of identifiers $1 \leq a_1 < a_2 < \cdots < a_{k-1} \leq n$ to a number in 2^q. Now we argue that B' also satisfies property P2.

For the sake of contradiction, suppose that there exist identifiers $1 \leq a_1 < a_2 < \cdots < a_k \leq n$ such that $B'(a_1, a_2, \ldots, a_{k-1}) = B'(a_2, a_3, \ldots, a_k)$. Let $q^* = B(a_1, a_2, \ldots, a_k) \in B'(a_1, a_2, \ldots, a_{k-1})$. Then, we must have $q^* \in B'(a_2, a_3, \ldots, a_k)$. Thus, $\exists a_{k+1} > a_k$ such that $q^* = B(a_2, a_3, \ldots, a_k, a_{k+1})$. But, in that case we would have $B(a_1, a_2, \ldots, a_k) = q^* = B(a_2, a_3, \ldots, a_k, a_{k+1})$, which is in contradiction with B being a k-ary q-coloring. Having reached at a contradiction by assuming that B' does not satisfy P2, we conclude that it actually does satisfy P2. Hence, B' is a $(k-1)$-ary 2^q-coloring. □

Proof of Theorem 2. For the sake of contradiction, suppose that there is an algorithm $\mathcal{A}$ that computes a 3-coloring of any n-node directed path in t rounds for $t < \frac{\log^* n}{2} - 2$. As stated in Observation 4, if there exists an algorithm $\mathcal{A}$ that computes a 3-coloring of any n-node directed path in t rounds for $t < \frac{\log^* n}{2} - 2$, then there exists a k-ary 3-coloring B, where $k = 2t + 1 < \log^* n - 3$. Using one iteration of Lemma 6, we would get that there exists a $(k-1)$-ary 8-coloring. Another iteration would imply that there exists a $(k-2)$-ary 2^8-coloring. Repeating this, after $k < \log^* n - 3$ iterations, we would get a 1-ary coloring with less than n colors. However, this is in contradiction with Lemma 5. Hence, such an algorithm $\mathcal{A}$ cannot exist. □

An Alternative Lower Bound Proof Via Ramsey Theory

Let us first briefly recall the basics of Ramsey Theory. The simplest case of Ramsey's theorem says that for any ℓ, there exists a number $R(\ell)$ such that for any $n \geq R(\ell)$, if we color the edges of the n-node complete graph K_n with two colors, there exists a monochromatic clique of size ℓ in it, that is, a set of ℓ vertices such that all of the edges between them have the same color. A simple example is that among any group of at least $6 = R(3)$ people, there are either at least 3 of them which are friends, or at least 3 of them no two of which are friends.

A similar statement is true in hypergraphs. Of particular interest for our case is coloring hyperedges of a complete n-vertex hypergraph of rank k, that is, the hypergraph where every subset of size k of the vertices defines one hyperedge. By Ramsey theory, it is known that there exists an n_0 such that, if $n \geq n_0$, for any way of coloring hyperedges of the complete n-vertex hypergraph of rank k with 3 colors, there would be a monochromatic clique of size $k+1$. That is, there would be a set of $k+1$ vertices a_1, $\ldots$, a_{k+1} in $\{1, \ldots, n\}$ such that all of their $\binom{k+1}{k} = k+1$ subsets with cardinality k have the same color.

In particular, consider an arbitrary k-ary coloring B, and let B define the colors of the hyperedges $\{a_1, \ldots, a_k\}$ when $1 \leq a_1 < a_2 < \cdots < a_k \leq n$. By Ramsey's theorem, we would get the following: there exist vertices $1 \leq a_1 < a_2 < \cdots < a_k < a_{k+1} \leq n$ such that B assigns the same color to hyperedges $\{a_1, \ldots, a_k\}$ and $\{a_2, \ldots, a_{k+1}\}$. But this is in contradiction with the property P2 of B being a k-ary coloring. The value of n_0 that follows from Ramsey theory is such that $k = O(\log^* n_0)$. In other words, Ramsey's theorem rules out $o(\log^* n)$-round 3-coloring algorithms for directed paths. See [CFS10] for more on hypergraph Ramsey numbers.

References

[CFS10] David Conlon, Jacob Fox, and Benny Sudakov. Hypergraph ramsey numbers. *Journal of the American Mathematical Society*, 23(1):247–266, 2010.

[Lin87] Nathan Linial. Distributive graph algorithms global solutions from local data. In *Proc. of the Symp. on Found. of Comp. Sci. (FOCS)*, pages 331–335. IEEE, 1987.

[Lin92] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[LS14] Juhana Laurinharju and Jukka Suomela. Brief announcement: Linial's lower bound made easy. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, pages 377–378. ACM, 2014.

[Nao91] Moni Naor. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM Journal on Discrete Mathematics*, 4(3):409–412, 1991.