Superlinear lower bounds for multipass graph processing*

Venkatesan Guruswami[†]
Carnegie Mellon University

Krzysztof Onak[‡]
IBM Research

August 2014

Abstract

We prove $n^{1+\Omega(1/p)}/p^{O(1)}$ lower bounds for the space complexity of p-pass streaming algorithms solving the following problems on n-vertex graphs:

- testing if an undirected graph has a perfect matching (this implies lower bounds for computing a maximum matching or even just the maximum matching size),
- testing if two specific vertices are at distance at most $2(p+1)$ in an undirected graph,
- testing if there is a directed path from s to t for two specific vertices s and t in a directed graph.

Prior to our result, it was known that these problems require $\Omega(n^2)$ space in one pass, but no $n^{1+\Omega(1)}$ lower bound was known for any $p \geq 2$.

These streaming results follow from a communication complexity lower bound for a communication game in which the players hold two graphs on the same set of vertices. The task of the players is to find out whether the sets of vertices at distance exactly $p+1$ from a specific vertex intersect. The game requires a significant amount of communication only if the players are forced to speak in a specific difficult order. This is reminiscent of lower bounds for communication problems such as indexing and pointer chasing. Among other things, our line of attack requires proving an information cost lower bound for a decision version of the classic pointer chasing problem and a direct sum type theorem for the disjunction of several instances of this problem.

1 Introduction

Graph problems in the streaming model have attracted a fair amount of attention over the last 15 years. Formally, a streaming algorithm is presented with a sequence of graph edges and it can read them one by one in the order in which they appear in the sequence. The main computational resource studied for this kind of algorithm is the amount of space it can use, i.e., the amount of information about the graph the algorithm remembers during its execution.

Furthermore, depending on the situation a small number of passes over the stream, instead of just one, may be allowed. This scenario may not be possible in the case of a router observing and analyzing high frequency network traffic, where the data is generated once and cannot be stored if it is extremely large. However, multiple passes over the stream may make sense in the case of processing a huge data set residing in a slow external storage device. For these kinds of devices, sequential access often maximizes readout efficiency and making a few passes over the data stream may be reasonable.

At a more theoretical level, relations between nodes (i.e., how they are connected in the graph and what the distances between them are) are a fundamental property of graphs that is worth studying. When it comes to exploring the structure of graphs, allowing for multiple passes seems to greatly improve the capabilities of streaming algorithms. For instance, the algorithm of Sarma, Gollapudi, and Panigrahy [31], which received the

[*] An extended abstract appeared in the proceedings of the 28th IEEE Conference on Computational Complexity [17]. Supported in part by NSF CCF-0963975 and the Simons Postdoctoral Fellowship.

[†] Supported in part by NSF CCF-0963975.

[‡] Supported by the Simons Postdoctoral Fellowship.

ISSN 1433-8092

PODS best paper award, uses multiple passes to construct long random walks in the graph in order to approximate PageRank for vertices. Also many strong lower bounds of $\Omega(n^2)$ space for one pass easily break if more than one pass is allowed. This is for instance the case for the early lower bounds of Henzinger, Raghavan, Rajogopalan [18] and also the lower bounds of Feigenbaum et al. [13].

On the other hand, constructing lower bounds for graph problems is usually based on constructing obstacles for local exploration, and our paper is not different in this respect. We show that finding out if two vertices are at a specific small distance p essentially requires $p/2$ passes to be accomplished in space $O(n)$. The main idea is similar to what is done for pointer chasing. Namely, we place edges in the order opposite to the sequence which enables easy exploration.

Our results. Let n be the number of vertices in the graph and let p be the allowed number of passes. We show strongly superlinear lower bounds of $T_{n,p} = \Omega\left(\frac{n^{1+1/(2p+2)}}{p^{20}\log^{3/2} n}\right)$ bits of space for three problems:

- testing if the graph has a perfect matching,
- testing if two prespecified vertices u and v are at distance at most $2(p+1)$ for an undirected input graph,
- testing if there is a directed path from u and v, where u and v are prespecified vertices and the input graph is directed.

In general, lower bounds stronger than $\Omega(n)$ require embedding a difficult instance of a problem into the "space of edges" as opposed to the "space of vertices," which turns out to be difficult in many cases. For instance, the $\Omega(n^2)$ lower bounds of [18] and [13] do not hold for algorithms that are allowed more than one pass.

Communication complexity is a standard tool for proving streaming lower bounds. We describe our hard communication problem from which we reduce to the streaming problems in Section 2. We now overview related work.

Matchings. In the maximum matching problem, the goal is to produce a maximum-cardinality set of non-adjacent edges. Streaming algorithms for this problem and its weighted version have received a lot of attention in recent years [13, 25, 11, 1, 24, 15, 22, 12, 32].

Our result compares most directly to the lower bound of Feigenbaum et al. [13], who show that even checking if a given matching is of maximum size requires $\Omega(n^2)$ space in one pass. Our result can be seen as an extension of their lower bound to the case when multiple passes are allowed. Even when $p \geq 2$ passes are allowed, we show that still a superlinear amount of space, roughly $n^{1+\Omega(1/p)}$, is required to find out if there is a perfect matching in the graph. This of course implies that tasks such as computing a maximum matching or even simply the size of the maximum matching also require this amount of space.

For the approximate version of the maximum matching problem, McGregor [25] showed that a $(1-\varepsilon)$-approximation can be computed in $\tilde{O}(n)$ space with the number of passes that is a function of only ε. The only known superlinear lower bound for the approximate matching size applies only to one-pass algorithms and shows that the required amount of space is $n^{1+\Omega(1/\log\log n)}$ if a constant approximation factor better than $1 - e^{-1}$ is desired [15, 22].

Shortest paths. We now move to the problem of computing distances between vertices in an undirected graph. Feigenbaum et al. [14] show that $\tilde{O}(n)$ space and one pass suffice to compute an $O(\log n/\log\log n)$-spanner and therefore approximate all distances up to a factor of $O(\log n/\log\log n)$. They also show a closely matching lower bound of $\Omega(n^{1+1/t})$ for computing a factor t approximation to distances between all pairs of vertices.

In the result most closely related to ours, they show that computing the *set* of vertices at distance p from a prespecified vertex in less than $p/2$ passes requires $n^{1+\Omega(1/p)}$ space. One can improve their lower bound to show that it holds even when the number of allowed passes is $p-1$. [1] As a result, to compute the distance p neighborhood in $O(n)$ space, essentially the best thing one can do is to simulate the BFS exploration with one step per pass over the input, which requires p passes. In this paper, we show a similar lower bound for the problem of just checking if two specific vertices are at distance exactly p. Our problem is algorithmically easier. If two vertices are at distance

[1] This follows by replacing one of their proof components with a stronger pointer chasing result from [16].

p, $\lceil p/2 \rceil$ passes and $O(n)$ space suffice, because one can simulate the BFS algorithm up to the radius of $\lceil p/2 \rceil$ from both vertices of interest. This is one of the reasons why our result cannot be shown directly by applying their lower bound.

A space lower bound of $\Omega(n^2)$ for one pass algorithms to find whether a pair of nodes is at distance 3 can be found in [13].

Directed connectivity. Feigenbaum et al. [13] show that the directed u-v connectivity problem requires $\Omega(n^2)$ bits of space to solve in one pass. However, their lower bound does not extend to more than one pass. Once again our lower bound extends their result to show that a superlinear lower bound holds for multiple passes. (Note that for *undirected* graphs, the problem of connectivity can easily be solved with one pass and $\tilde{O}(n)$ space, using for instance the well known union-find algorithm.)

Optimality of our lower bound. We conjecture that our lower bounds can be improved from $\tilde{\Omega}(n^{1+1/(2p+2)})$ to $\tilde{\Omega}(n^{1+1/(p+1)})$ for $p = O(1)$. For the matching problem, our lower bound is based on showing that finding a single augmenting path is difficult. It is an interesting question if a stronger lower bound can be proved in the case where more augmenting paths have to be found. Currently, no $o(n^2)$-space streaming algorithm is known for this problem with a small number of passes.

Paper organization. We begin in Section 2 with a description of the communication problems we study and a high-level overview of our lower bound approach. We set up some useful information-theoretic preliminaries in Section 3. We state our main communication complexity lower bound (Theorem 4) and use it to show our streaming lower bounds in Section 4. Our communication lower bound is proved in three steps, and we go into the details of these steps in the next three sections. Finally, in Section 8 we put the steps together to give a proof of Theorem 4.

2 Proof overview and techniques

Via simple reductions, our multipass streaming lower bounds for matching and connectivity reduce to proving communication complexity lower bounds for a certain decision version of "set pointer chasing." The reductions to streaming are described in Section 4, and in this section we give an overview of our communication complexity results. We start with a description of the communication problems that will be useful for us.

2.1 Communication Problems

We assume private randomness in all communication problems, unless otherwise stated. Furthermore, all messages are public, i.e., can be seen by all players (the setting sometimes described as the blackboard model). Consider a communication problem with p players P_1, …, P_p. Players speak in r rounds and in each round they speak in order P_1 through P_p. At the end of the last round, the player P_p has to output the solution. We call any such problem a (p, r)-*communication problem.*

We define $[n]$ as $\{1, \ldots, n\}$. For any set A, we write 2^A to denote the power set of A, i.e., the set of all subsets of A. For any function $f : A \to 2^B$, we define a mapping $\overrightarrow{f} : 2^A \to 2^B$ such that $\overrightarrow{f}(S) = \bigcup_{s \in S} f(s)$.

Pointer and Set Chasing. The *pointer chasing* communication problem $\mathsf{PC}_{n,p}$, where n and p are positive integers, is a $(p, p-1)$-communication problem in which the i-th player P_i has a function $f_i : [n] \to [n]$ and the goal is to compute $f_1(f_2(\ldots f_p(1) \ldots))$. The complexity of different versions of this problem was explored thoroughly by a number of works [28, 10, 27, 9, 29, 16].

The *set chasing* communication problem $\mathsf{SC}_{n,p}$, for given positive integers n and p, is a $(p, p-1)$-communication problem in which the i-th player P_i has a function $f_i : [n] \to 2^{[n]}$ and the goal is to compute $\overrightarrow{f_1}(\overrightarrow{f_2}(\ldots \overrightarrow{f_p}(\{1\}) \ldots))$. A two-player version of the problem was considered by Feigenbaum et al. [14].

Operators on Problems. For a (p, r)-communication problem $\mathcal{X}$, we write $\mathsf{EQUAL}(\mathcal{X})$ to denote a $(2p, r)$-communication problem in which the first p players P_1, …, P_p hold one instance of $\mathcal{X}$, the next p players P_{p+1},

…, P_{2p} hold another instance of $\mathcal{X}$, and the goal is to output one bit that equals 1 if and only if the outputs for the instances of $\mathcal{X}$ are equal. See Figure 1 for an example.

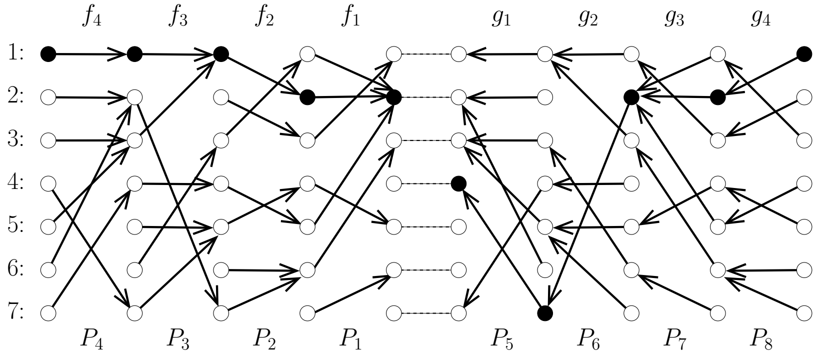

Figure 1: A sample instance of $\mathsf{EQUAL}(\mathsf{PC}_{7,4})$ with a negative solution. It consists of two instances of $\mathsf{PC}_{7,4}$ held by two different sets of players.

Analogously, for a (p, r)-communication problem $\mathcal{X}$ such that the output is a set, we write $\mathsf{INTERSECT}(\mathcal{X})$ to denote the $(2p, r)$-communication problem in which the first p players P_1, …, P_p hold one instance of $\mathcal{X}$, the next p players P_{p+1}, …, P_{2p} hold another instance of $\mathcal{X}$, and the goal is to output one bit that equals 1 if and only if the sets that are solutions to the instances of $\mathcal{X}$ intersect. See Figure 2 for an example.

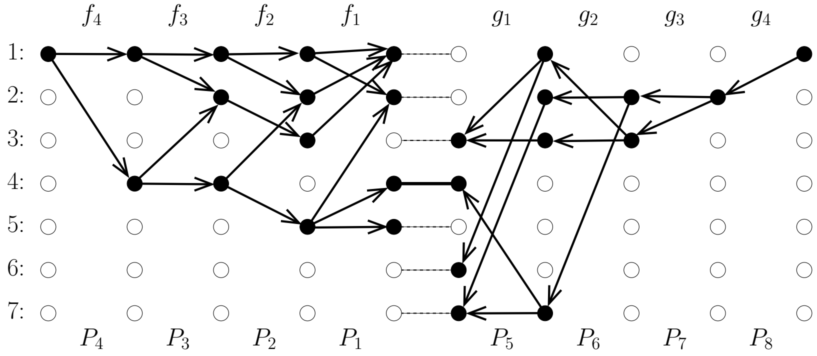

Figure 2: A sample instance of $\mathsf{INTERSECT}(\mathsf{SC}_{7,4})$, where two final sets intersect. The edges outgoing from vertices that are not visited were omitted.

For a (p, r)-communication problem $\mathcal{X}$ with a Boolean output, we write $\mathsf{OR}_t(\mathcal{X})$, where t is a positive integer, to denote the (p, r)-communication problem in which players have t instances of $\mathcal{X}$ and want to output the disjunction of their results.

Limited Pointer Chasing Equality. We say that a function $f : A \to B$ is *r-non-injective*, where r is a positive integer, if there is an $A' \subseteq A$ of size r and a $b \in B$ such that for all $a \in A'$, $f(a) = b$.

We write $\mathsf{LPCE}_{n,k,r}$ to denote a modified version of $\mathsf{EQUAL}(\mathsf{PC}_{n,k})$. In $\mathsf{LPCE}_{n,k,r}$ the last player has to output the same value as in $\mathsf{EQUAL}(\mathsf{PC}_{n,k})$, unless one of the functions in one of the pointer chasing instances is r-non-injective, in which case the last player has to output 1. This is a technical extension to ensure that no element has too many pre-images, which is necessary to make one of our reductions work.

2.2 Lower bound for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$

Our multipass streaming lower bounds for matching and connectivity reduce to proving a communication complexity lower bound for the set chasing intersection problem $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$. Note that if the players spoke in the order P_{2p}, P_{2p-1}, …, P_1, then they would be able to solve both instances of $\mathsf{SC}_{n,p}$, using at most $O(n)$ communication per player, which is enough to solve the intersection problem. If the players spoke in the desired order P_1, P_2, …, P_{2p} but were allowed a total of p rounds then they would be able to solve the instances of $\mathsf{SC}_{n,p}$ with $O(n)$ communication per message by simulating one step in the pointer chasing instance per round. Our main result is however that if the number of allowed rounds is $p-1$, then approximately $n^{1+\Omega(1/p)}$ bits of communication are needed to solve the problem, even for randomized protocols with constant error.

Our result is reminiscent of the classic $\Omega(n)$ communication complexity lower bounds for problems such as indexing and pointer chasing $\mathsf{PC}_{n,p}$ [27, 16] when the players speak in the "wrong" order. Guha and McGregor [16] adapt the proof of Nisan and Wigderson [27] to show that solving $\mathsf{PC}_{n,p}$ (in $p-1$ passes) requires $\Omega(n)/p^{O(1)}$ total communication even if the protocol can be randomized and can err with small constant probability. Increasing the number of rounds to p or letting the players speak in the opposite order (even in just one round) would result in a problem easily solvable with messages of length $O(\log n)$.

Even more directly related is the construction of Feigenbaum et al. [14], who show that solving $\mathsf{SC}_{n,p}$ requires $n^{1+\Omega(1/p)}$ communication in less then p passes[2]. Their proof follows by using a direct sum theorem of Jain, Radhakrishnan and Sen [21] to show that solving $t \approx n^{\Theta(1/p)}$ instances of $\mathsf{PC}_{n,p}$ requires roughly t times more communication than solving a single instance. Then they show that an efficient protocol for solving $\mathsf{SC}_{n,p}$ would result in an efficient protocol for solving t instances of $\mathsf{PC}_{n,p}$ in parallel.

Compared to $\mathsf{SC}_{n,p}$, $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ is a decision problem. In particular, there seems to be no reduction allowing to reconstruct the sets reached in $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$. The only thing that we learn after an execution of the protocol is whether these two sets intersect. Therefore, reducing our question to that of [14] seems unlikely.

Our proof of the above communication complexity lower bound proceeds in three steps:

STEP A: Reduction to proving a communication lower bound for $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$.

STEP B: A direct sum style step lower bounding the communication complexity of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ as roughly t times the communication complexity of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$.

STEP C: An information complexity lower bound for $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$.

The technical body of the paper actually proves these steps in the opposite order (Steps A, B, and C are discussed in Sections 7, 6, and 5, respectively). But here we will expand on the steps in the above order. The actual proof works with a variant of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$, namely $\mathsf{LPCE}_{n,p,r}$ which we defined earlier, in order to deal with functions f_i that may be highly non-injective, and which may break the reduction in Step A. For simplicity, we ignore this aspect in the overview, but it is worth keeping in mind that this complicates the execution of Step C on the information complexity lower bound.

Step A: Reduction to proving a lower bound for $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$. Our idea here is to use a communication protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ to give a protocol that can answer if at least one of t instances of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ has an Yes answer, where $t = n^{\Theta(1/p)}$. (Recall that in the $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ problem, the input consists of two instances of $\mathsf{PC}_{n,p}$ with functions $\{f_i, g_i : [n] \to [n]\}_{i=1}^{p}$ and the goal is to output Yes iff we end up at the same index in both instances, i.e., if $f_1(f_2(\ldots f_p(1)\ldots)) = g_1(g_2(\ldots g_p(1)\ldots))$.) Given t instances of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$, for each instance independently, we randomly scramble the connections in every layer while preserving the answer to $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$. We then overlay all these instances on top of each other to construct an instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ (note that each node has exactly t neighbors in the next layer).

By construction, given an Yes instance of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$, by following the mappings from the instance of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ which has an Yes answer, we also obtain an element that belongs to the intersection of two resulting sets in $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$. Since $t = n^{\Theta(1/p)}$, we have $t^{2p} \ll n$, and we argue that the random scramblings ensure that if none of t instances of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ have an Yes answer, then it is unlikely that the two resulting sets in the instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ will intersect. This constraint on t is what limits our lower bound to $\sim n^{1+1/(2p)}$.

[2] In fact, they show this for roughly less than $p/2$ passes, but replacing the lower bound of [27] with the lower bound of [16] and extending some other complexity results to the setting with multiple players yields the improved bound claimed here.

Step B: A direct sum style argument. In this step, our goal is to argue that the randomized communication complexity of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ is asymptotically $\Omega(t)$ times larger than that of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$. This is reminiscent of direct sum results of the flavor that computing answers to t instances of a problem requires asymptotically t times the resources, but here we only have to compute the OR of t instances. Our approach is to use the information complexity method that has emerged in the last decade as a potent tool to tackle such direct sum like questions [8, 2, 21], and more recently in [3, 5] and follow-up works. The introductions of [3, 19] contain more detailed information and references on direct sum and direct product theorems in communication complexity.

Our hard distribution will be the uniform distribution over all inputs. Being a product distribution, the information complexity will be at least the sum of the mutual information between the i-th input and the transcript, for $1 \le i \le t$. Using the fact that the probability of an Yes answer on a random instance of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ is very small (at most $O(1/n)$), we prove that the mutual information between the i-th input and the transcript cannot be much smaller than the information cost of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ for protocols that err with probability $o(1/n)$ under the uniform distribution.

Step C: Lower bound for information cost of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$. This leaves us with the task of lower bounding the information cost of low error protocols for $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ under the uniform distribution. This is the most technical of the three steps. We divide this step into two parts.

First we show that if there were a protocol with low information cost IC on the uniform distribution, then there would exist a deterministic protocol that on the uniform distribution would send mostly short messages and err with at most twice the probability. This is done by adapting the proof of the message compression result of [21] for bounded round communication protocols. We cannot use their result as such since in order to limit the increase in error probability to γ, the protocol needs to communicate $\Omega(1/\gamma^2)$ bits. This is prohibitive for us as we need to keep the error probability as small as $O(1/n)$, and can thus only afford an additive $O(1/n)$ increase. We present a twist to the simulation obtaining a deterministic protocol with at most twice the original error probability. The protocol may send a long message with some small probability ε and in other cases communicates at most $O(\mathrm{IC}/\varepsilon^2)$ bits. In our application, we set ε to be a polynomial in $1/p$.

The second part is a lower bound for $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ against such "typically concise" deterministic protocols. To prove this, we show that if the messages in the deterministic protocol are too short, then with probability at least $1/2$, the protocol will have little knowledge about whether the solutions to two instances of pointer chasing are identical and therefore, will still err with probability $\Omega(1/n)$, which is significant from our point of view. The proof extends the lower bound for pointer chasing due to Nisan and Wigderson [27] and its adaptation due to Guha and McGregor [16]. We have to overcome some technical hurdles as we need a lower bound for the equality checking version and not for the harder problem of computing the pointer's value. Further, we need to show that a *constant* fraction of the protocol leaves are highly uncertain about their estimate of the pointers' values, so that they would err with probability $\Omega(1/n)$ (with $1/n$ being the collision probability for completely random and independent values).

Summarizing, Step C can be seen as a modification of techniques of [27, 16] to prove a communication lower bound for $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ combined with techniques borrowed from [21] to imply a lower bound for information complexity. The relationship between information complexity and communication complexity has been a topic of several papers, starting with [8, 21] for protocols with few rounds, and more recently [3, 5, 4, 7, 6, 23] for general protocols.

3 Preliminaries

Constant $C_\star$. Let $C_\star$ be a constant such that the probability that a function $f : [n] \to [n]$ selected uniformly at random is $C_\star(1 + \log n)$-non-injective is bounded by $1/(2n^2)$. The existence of $C_\star$ follows from a combination of the Chernoff and union bounds.

3.1 Information and Communication Complexity

Communication complexity and information complexity play an important role in our proofs. We now provide necessary definitions for completeness.

The *communication complexity* of a protocol is the function from the input size to the maximum length of messages generated by the protocol on an input of a specific size. For a problem $\mathcal{X}$ and $\delta \in [0,1]$, the communication complexity of $\mathcal{X}$ with error δ is the function from the input size to the infimum communication complexity of private-randomness protocols that err with probability at most δ on any input. We write $R_\delta(\mathcal{X})$ to denote this quantity.

The *information cost*[3] $\mathrm{ICost}_\psi(\Pi)$ of a protocol Π on input distribution ψ equals the mutual information $I(X;\Pi(X))$, where X is the input distributed according to ψ and $\Pi(X)$ is the transcript of Π on input X.

The *information complexity* $\mathrm{IC}_{\psi,\delta}$ of a problem $\mathcal{X}$ on a distribution ψ with error δ is the infimum $\mathrm{ICost}_\psi(\Pi)$ taken over all private-randomness protocols Π that err with probability at most δ for any input.

3.2 Useful information-theoretic lemmas

Let us first recall a result that says that if a random variable has large entropy, then it behaves almost like the uniform random variable on large sets.

Fact 1 ([30], see also [27, Lemma 2.10])**.** *Let X be a random variable on $[n]$ with $H(X) \geq \log n - \delta$. Let $S \subseteq [n]$ and let $\Delta = \sqrt{\frac{4\delta n}{|S|}}$. If $\Delta \leq 1/10$, then* $\Pr[X \in S] \geq \frac{|S|}{n}(1-\Delta)$ *.*

Using the above result, we show that it is hard to guess correctly with probability $1 - o(1/n)$ if two independent random variables distributed on $[n]$ collide if they have large entropy.

Lemma 2. *Let X and Y be two independent random variables distributed on $[n]$ such that both $H(X)$ and $H(Y)$ are at least $\log n - \delta$, where $\delta = 48^{-2}$. Then*

- $\Pr[X = Y] \geq 1/(8n)$*, and*
- *if* $n \geq 4$*,* $\Pr[X \neq Y] \geq 1/4$*.*

Proof. We first prove that there is a set $S_X \subseteq [n]$ such that $|S_X| \geq \frac{3}{4}n$ and for each $x \in S_X$, $\Pr[X = x] \geq 1/(2n)$. Suppose that there is no such set. Then there is a set T_X of size more than $n/4$ in which every element has probability less than $1/(2n)$, and therefore, $\Pr[X \in T_X] < \frac{|T_X|}{2n}$. Note that $\sqrt{\frac{4\delta n}{|T_X|}} \leq 1/12 < 1/10$, which implies that we can apply Fact 1 to T_X. We obtain $\Pr[X \in T_X] \geq \frac{11}{12} \cdot \frac{|T_X|}{n}$, which contradicts the size of T_X and implies that S_X with the desired properties does exist.

Analogously, one can prove that there is a set $S_Y \subseteq [n]$ such that $|S_Y| \geq \frac{3}{4}n$ and for each $y \in S_Y$, $\Pr[Y = y] \geq 1/(2n)$. Note that $|S_X \cap S_Y| \geq n/2$. For each $x \in S_X \cap S_Y$, $\Pr[X = Y = x] \geq 1/(4n^2)$. Hence

$$\Pr[X = Y] \geq \sum_{x \in S_X \cap S_Y} \Pr[X = Y = x] \geq 1/(8n).$$

To prove the second claim, for $n \geq 4$, observe that for every setting x of X, $|S_y \setminus \{x\}| \geq \frac{3n}{4} - 1 \geq n/2$, and therefore, the probability that $Y \neq X$ is at least $|S_y \setminus \{x\}| \cdot 1/(2n) \geq 1/4$. □

The following lemma gives a bound on the entropy of a variable that randomly selects out of two random values based on another 0-1 valued random variable.

[3] Note that this is the *external* information cost following the terminology of [3]. For product distributions ψ, this also equals the *internal* information cost. As product distributions will be our exclusive focus in this paper, this distinction is not relevant to us, and we will simply use the term information cost.

Lemma 3. *Let X_0, X_1, and Y be independent discrete random variables, where X_0 and X_1 are distributed on the same set Ω and Y is distributed on $\{0, 1\}$. Then*

$$H(X_Y) \leq 1 + \sum_{i=0}^{1} \Pr[Y = i] \cdot H(X_i).$$

Proof. For each $i \in \{0, 1\}$ and each $\omega \in \Omega$, let $p_{i,\omega} = \Pr[X_i = \omega]$. Let $q = \Pr[Y = 0]$. For each $\omega \in \Omega$, $\Pr[X_Y = \omega] = qp_{0,\omega} + (1-q)p_{1,\omega}$. The inequality follows from the concavity of the function $f(x) = x \cdot \log(1/x)$ on $[0, 1]$ and simple algebraic transformations:

$$\begin{aligned}
H(X_Y) &= \sum_{\omega\in\Omega} \left(qp_{0,\omega} + (1-q)p_{1,\omega}\right) \cdot \log\left(1/\left(qp_{0,\omega} + (1-q)p_{1,\omega}\right)\right) \\
&\leq \sum_{\omega\in\Omega} \left(qp_{0,\omega} \cdot \log(1/qp_{0,\omega}) + (1-q)p_{1,\omega} \cdot \log(1/(1-q)p_{1,\omega})\right) \\
&= q \cdot \log(1/q) + q \cdot \sum_{\omega\in\Omega} p_{0,\omega} \log(1/p_{0,\omega}) \\
&\quad + (1-q) \cdot \log(1/(1-q)) + (1-q) \cdot \sum_{\omega\in\Omega} p_{1,\omega} \log(1/p_{1,\omega}) \\
&= H(q) + q \cdot H(X_0) + (1-q) \cdot H(X_1) \\
&\leq 1 + \sum_{i=0}^{1} \Pr[Y = i] \cdot H(X_i).
\end{aligned}$$

□

4 The Main Tool and Its Applications

The main tool in our paper is the following lower bound for the communication complexity of set chasing intersection.

Theorem 4. *For n larger than some positive constant and $p \leq \frac{\log n}{\log\log n}$,*

$$R_{1/10}(\mathsf{INTERSECT}(\mathsf{SC}_{n,p})) = \Omega\left(\frac{n^{1+1/(2p)}}{p^{16} \cdot \log^{3/2} n}\right).$$

We now present relatively straightforward applications of this theorem to three graph problems in the streaming model.

Theorem 5. *Solving the following problems with probability at least $9/10$ in the streaming model with $p = O\left(\frac{\log n}{\log\log n}\right)$ passes requires at least $\Omega\left(\frac{n^{1+1/(2(p+1))}}{p^{20} \cdot \log^{3/2} n}\right)$ bits of space:*

PROBLEM 1: *For two given vertices u and v in an undirected graph, check if the distance between them is at most $2(p + 1)$.*

PROBLEM 2: *For two given vertices u and v in a directed graph, check if there is a directed path from u to v.*

PROBLEM 3: *Test if the input graph has a perfect matching.*

Proof. Let us consider the problems one by one. For Problem 1, we turn an instance of $\mathsf{INTERSECT}(\mathsf{SC}_{k,p+1})$ into a graph on $n = (2p + 3) \cdot k$ vertices. We modify the graph in Figure 2 as follows. First, we make all edges undirected. Second, we merge every pair of middle vertices connected with a horizontal line into a single vertex. Any path between the top leftmost vertex u and the top rightmost vertex v is of length at least $2p + 2$. The length of the path is exactly $2p + 2$ if and only if it moves to the next layer in each step. Note that this corresponds

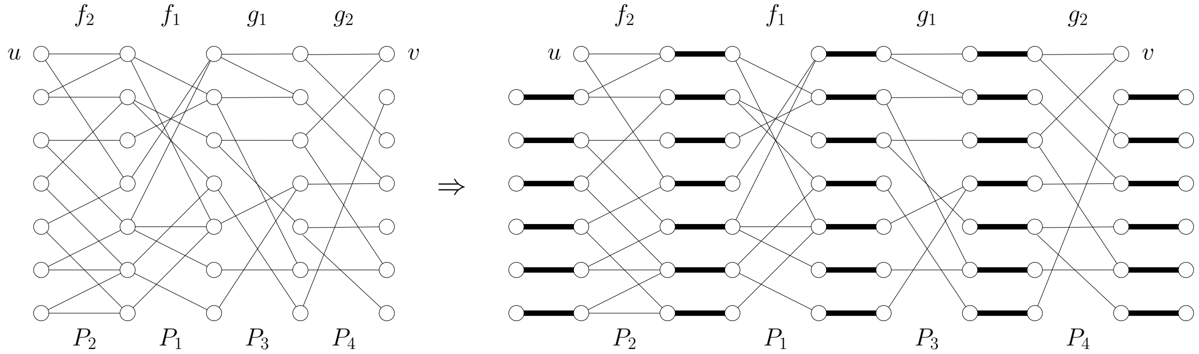

Figure 3: Reduction to the perfect matching problem.

to the case that the final sets for two instances of $\mathsf{SC}_{k,p+1}$ intersect. We create the input stream by inserting first the edges describing the function held by P_1, then by P_2 and so on, until P_{2p+2}. If there is a streaming algorithm for the problem that uses at most T bits of space, then clearly there is a communication protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{k,p+1})$ with total communication $(2p+2) \cdot p \cdot T$ and the same error probability as the streaming algorithm. The protocol can be obtained by the players by simulating the streaming algorithms on their parts of the input and communicating its state. This implies that $T = \Omega\left(\frac{1}{p^2} \cdot \frac{(n/p)^{1+1/(2(p+1))}}{p^{16} \cdot \log^{3/2} n}\right) = \Omega\left(\frac{n^{1+1/(2(p+1))}}{p^{20} \cdot \log^{3/2} n}\right)$.

For Problem 2, the reduction is almost the same, with the only difference being that we make all edge directed from left to right and we want to figure out if there is a directed path from the top leftmost vertex to the top rightmost vertex. Such a path exists if and only if the final sets in the instance of $\mathsf{INTERSECT}(\mathsf{SC}_{k,p+1})$ intersect.

For Problem 3, the reduction is slightly more complicated. We show how to modify the hard instance G that we have created for Problem 1. Let us first add a perfect matching before and after every layer of edges of the hard instance for Problem 1, except for the first and the last layer, in which we omit one edge. The omitted edges are incident to the vertices u and v corresponding to value 1, i.e., the vertices that we want to connect using a path going directly from left to right in Problem 1. See Figure 3 for an example. Note that the additional edges constitute a matching M in which all but two vertices are matched. Now the graph has a perfect matching iff there exists an augmenting path in M between u and v, which are the unmatched vertices. Any augmenting path has to alternate between matched and unmatched edges, which implies in our case, that it has to go directly from left to right. Therefore, any augmenting path in M corresponds to a path going directly from left to right in G and connecting u and v. The only difference is that the augmenting path has additional edges coming from the matchings were inserted into G. Therefore the streaming algorithm for testing if a graph has a perfect matching can be used to create a protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{k,p+1})$, where the additional edges can be placed at the beginning of the stream. □

5 Step 1: Information Complexity of Pointer Chasing Equality

To prove the main theorem of the paper, we first show a lower bound for the information complexity of Limited Pointer Chasing Equality.

Lemma 6. *Let n and p be positive integers such that $n \geq 18p^2$. It holds*

$$\begin{aligned}\mathrm{IC}_{\mu,1/(64n)}(\mathsf{LPCE}_{n,p,C_\star(1+\log n)}) &\geq \frac{n}{2^{29} \cdot 3^5 \cdot p^{16}} - \frac{\lceil 2\log n \rceil}{2^{13} \cdot 3^2 \cdot p^8} - 2 \\ &= \Omega\left(\frac{n}{p^{16}}\right) - O\left(\frac{\log n}{p^8} + 1\right),\end{aligned}$$

where μ is the uniform distribution on all possible inputs of the problem.

The proof consists of two smaller steps. First we show that if there is a protocol with low information cost on the uniform distribution, then there is a deterministic protocol that on the uniform distribution sends mostly short messages, and errs with at most twice the probability. Then we show that the messages in the deterministic protocol cannot be too short. Otherwise, with probability at least $1/2$, the protocol would have little knowledge about whether the solutions to two instances of pointer chasing are identical. In this case the protocol would still err with probability $\Omega(1/n)$.

5.1 Transformation to Deterministic Typically Concise Protocols

Let us first define concise protocols, which send short messages most of the time.

Definition 7. *We say that a protocol P is an (m, ε)-*concise *protocol for an input distribution μ if for each i, the probability that the i-th message in the protocol is longer then m is bounded by ε.*

The following three facts from [20, 21] are very useful in our proofs. They regard information theory and random variables. For a distribution P on $\mathbb{N}$, we write $P(i)$ to denote the probability of selecting i from P. For two distributions P and Q, we write $D_{\mathrm{KL}}(P\|Q)$ to denote the Kullback-Leibler divergence of Q from P.

Fact 8 ([21, Fact 1])**.** *Let X, Y, and Z be random variables. The following identity holds: $I(X;Y,Z) = I(X;Y) + I(X;Z|Y)$.*

Fact 9 ([21, Fact 2])**.** *Let X and Y be a pair of random variables. Let P be the distribution of Y and let P_x be the distribution of Y given $X = x$. Then $I(X;Y) = E_X[D_{\mathrm{KL}}(P_X\|P)]$, where $D_{\mathrm{KL}}(Q_1\|Q_2)$ denotes the KL-divergence between distributions Q_1, Q_2.*

Fact 10 ([20, 21, Substate Theorem])**.** *Let P and Q be probability distributions on $\mathbb{N}$ such that $D_{\mathrm{KL}}(P\|Q) = a$. Let $\varepsilon \in (0,1)$ and let $\mathrm{Good} = \{i \in \mathbb{N} : P(i) \cdot 2^{-(a+1)/\varepsilon} \le Q(i)\}$. If X is a random variable distributed according to P, then $\Pr[X \in \mathrm{Good}] \ge 1 - \varepsilon$.*

We now show an auxiliary lemma that shows that if $D_{\mathrm{KL}}(P\|Q)$ is bounded then a relatively short sequence of independent random variables distributed according to Q suffices to generate a random value from P. The lemma is an adaptation of a lemma from [21].

Lemma 11. *Let P and Q be two probability distributions on $\mathbb{N}$ such that $D_{\mathrm{KL}}(P\|Q) < \infty$. Let $(\Gamma_1, \Gamma_2, \Gamma_3, \ldots)$ be a sequence of independent random variables, each distributed according to Q. Let $\Gamma_0 = -1$. Let $\varepsilon \in (0,1)$. There is a set $\mathrm{Good} \subseteq \mathbb{N}$ and a random variable $R \in \mathbb{N}$ such that*

- $\sum_{i \in \mathrm{Good}} P(i) \ge 1 - \varepsilon$,
- *for all* $x \in \mathrm{Good}$, $\Pr[\Gamma_R = x] = P(x)$,
- $E[R] \le 2^{(D_{\mathrm{KL}}(P\|Q)+1)/\varepsilon}$.

Proof. Let the set Good be defined as in Fact 10, i.e., $\mathrm{Good} = \{i \in \mathbb{N} : P(i) \cdot 2^{-(a+1)/\varepsilon} \le Q(i)\}$, where we set $a = D_{\mathrm{KL}}(P\|Q)$. Following [21], we use rejection sampling to prove the lemma. Consider the following process. For consecutive positive integers j, starting from 1, do the following. Look at the value γ_j taken by Γ_j. If $\gamma_j \in \mathrm{Good}$, toss a biased coin and with probability $P(j) \cdot 2^{-(a+1)/\varepsilon}/Q(j)$, set $R = j$ and finish the process. If $\gamma_j \notin \mathrm{Good}$ or the coin toss did not terminate the process, toss another biased coin and with probability

$$2^{-(a+1)/\varepsilon} \cdot \frac{(1 - \sum_{i \in \mathrm{Good}} P(i))}{(1 - 2^{-(a+1)/\varepsilon} \cdot \sum_{i \in \mathrm{Good}} P(i))},$$

set $R = 0$ and also terminate the process. Otherwise, continue with j increased by 1. The process terminates with probability 1.

Let us argue that R and Good have the desired three properties. The first property is a consequence of Fact 10. To prove the other two, observe what happens when the process reaches a specific j. The process terminates with $R = j$ and $\Gamma_R = x$ for a specific $x \in$ Good with probability $P(x) \cdot 2^{-(a+1)/\varepsilon}$. The probability that it terminates with $R = 0$ equals exactly $2^{-(a+1)/\varepsilon} \cdot (1 - \sum_{i \in \text{Good}} P(i))$. Since these probabilities are independent of j, when the process eventually terminates, the probability of $\Gamma_R = x$ for each $x \in$ Good is exactly $P(x)$, which proves the second property. Finally, the probability that the process terminates for a specific j after reaching it is exactly $2^{-(a+1)/\varepsilon}$. Clearly, $E[R]$ is bounded from above by the expected j for which the process stops, which in turn equals exactly $2^{(a+1)/\varepsilon}$. □

The following lemma allows for converting protocols with bounded information cost on a specific distribution into deterministic protocols that mostly send short messages on the same distribution. The proof of the lemma is a modification of the message compression result of [21]. An important feature of our version is that the error probability is only doubled, instead of an additive constant increase which we cannot afford. A simple but key concept we use to achieve this is to allow the protocol to send long messages with some small (constant) probability. We then handle such "typically concise" protocols in our lower bound of Section 5.2.

Lemma 12. *Let Π be a private-randomness protocol for a (p, r)-communication problem $\mathcal{P}$ such that Π errs with probability at most $\delta > 0$ on a distribution μ. For any $q > 0$, there is a protocol Π' for $\mathcal{P}$ such that*

- *Π' is deterministic,*
- *Π' errs with probability at most 2δ on μ,*
- *Π' is (m, q)-concise, where $m = 128 \cdot (\text{ICost}_\mu(\Pi) + 2) \cdot (pr/q)^2$.*

Proof. We start with a proof overview. There are $pr - 1$ messages sent in Π. We construct a series of intermediate protocols Π'_{pr-1}, Π'_{pr-2}, …, Π'_1, where Π'_i is a modification of the protocol Π in which, for $i \leq j \leq pr - 1$, the j-th message is likely to be short. The first $i - 1$ messages of Π'_i are the same as the messages of Π. In particular, Π'_i uses only private randomness to generate the first $i - 1$ messages. Later messages are generated using public randomness. The players in the modified protocols will convey exactly the same information as the original protocol Π and therefore, the protocols will err with the same probability, with the only difference being a different encoding of messages and the use of public randomness.

For convenience, let Π'_{pr} be the original protocol Π. We now explain how we convert Π'_{i+1} into Π'_i. Let M_0 be the random variable corresponding to the sequence of the first $i - 1$ messages in Π'_{i+1}. Let M_1 be the random variable describing the i-th message in Π'_{i+1}. Let $M = (M_0, M_1)$. Recall that M is distributed in the same way as its equivalent for the original protocol P. Let P_j be the player sending the i-th message (i.e., $j \equiv i \pmod p$). Let X be the combined inputs of the other players, and let Y be the input of the j-th player. We write $M_1^{m_0}$ to describe the distribution of M_1 when $M_0 = m_0$. Moreover, we write M_1^{x,y,m_0} to describe the distribution of M_1 when $X = x$, $Y = y$, and $M_0 = m_0$. The distribution M_1^{x,y,m_0} does not depend on x, because the protocol uses only private randomness and to generate the i-th message it only uses the previous messages and y, the input of the i-th player. It follows from Fact 8 and Fact 9 that

$$\begin{aligned} I(X, Y; M) &= I(X, Y; M_0) + I(X, Y; M_1 | M_0) \\ &= I(X, Y; M_0) + E_{M_0, X, Y}\left[D_{\text{KL}}\left(M_1^{X,Y,M_0} \middle\| M_1^{M_0}\right)\right]. \end{aligned}$$

We define a_i as $E_{M_0,X,Y}\left[D_{\text{KL}}\left(M_1^{X,Y,M_0} \middle\| M_1^{M_0}\right)\right]$ for this specific setting of i. Overall, it follows by induction that the mutual information between the input and the protocol transcript, i.e., $\text{ICost}_\mu(P)$, equals $\sum_{i=1}^{pr-1} a_i$. This also implies that $D_{\text{KL}}(M_1^{x,y,m_0} \| M_1^{m_0}) < \infty$ for any setting $X = x$, $Y = y$, and $M_0 = m_0$ that has nonzero probability.

Recall that the first $i - 1$ messages of Π'_i are generated in the same way as in Π'_{i+1}. We now describe how P_j generates the i-th message. Let m_0 be the messages sent so far. The distribution of the i-th message, $M_1^{m_0}$ is known to all the players. Let $(\Gamma_1, \Gamma_2, \Gamma_3, \ldots)$ be an infinite sequence of independent random variables, where each Γ_i is drawn independently from the distribution $M_1^{m_0}$. The sequence of Γ_i's is generated using public randomness, so it is known to all the players as well. We now use Lemma 11, where we set $Q = M_1^{m_0}$, $P = M_1^{x,y,m_0}$, and $\varepsilon = q/8pr$. Note that the distribution M_1^{x,y,m_0} does not depend on x, because the randomness is private in the first i messages in Π'_{i+1}. P_j will transmit exactly the same information in Π'_i as in Π'_{i+1}. The player P_j fixes a set Good and a random variable R as in Lemma 11. If $\Gamma_R \in$ Good, then the player sends a single bit 0 followed by a prefix-free encoding of the value R. Due to the concavity of the logarithm, the expected length of the message can be bounded by $1 + 16pr(D_{\mathrm{KL}}(M_1^{x,y,m_0} \| M_1^{m_0}) + 1)/q + 2 \le 16pr(D_{\mathrm{KL}}(M_1^{x,y,m_0} \| M_1^{m_0}) + 2)/q$, where the additional factor of 2 and the additional additive term of 1 come from the prefix free encoding. Overall, the expected length of the message starting with 0 equals $16pr(a_i + 2)/q$.

If $\Gamma_R \notin$ Good, the player generates the message from the part of distribution M_1^{x,y,m_0} restricted to $\mathbb{N} \setminus$ Good and transmits the selected value prefixing it with a single bit 1. Overall, all players can decode a message generated according to M_1^{x,y,m_0} and then behave in the same way as in the protocol Π'_{i+1}.

After applying a sequence of $pr - 1$ steps of the transformation, we obtain a randomized protocol Π'_1 that still errs with probability δ. We now show that there is a suitable setting of random bits in the protocol to obtain the desired deterministic protocol Π'. First, let $Z \in \{0, 1\}$ be a random variable indicating that the algorithm returned an incorrect answer. We have $E[Z] = \delta$, where this expectation and all expectations in the remainder of this proof are taken over both inputs drawn from μ and random bits of the protocol Π'_1. It follows from Markov's inequality that the probability that fixing the random bits makes the algorithm err with probability higher than 2δ on μ is at most $1/2$. Consider now the i-the message in Π'_1, where $1 \le i \le pr - 1$. Let Z'_i be the random indicator variable that equals 1 if the i-th message starts with 1. It follows from our construction that $E[Z'_i] \le \varepsilon = q/8pr$. If we fix the random bits of the protocol, the probability that the i-th message starts with 1 with probability higher than $q/2$ is bounded by Markov's inequality by $1/4pr$. Finally, let Z''_i be the random indicator variable that the i-th message in Π'_i starts with 0 and has length greater than $128(a_i + 2)(pr/q)^2$. Consider now a random variable W_i that equals the length of the i-th message if the message starts with 0 and 0 if it starts with 1. We know that $E[W_i] \le 16pr(a_i + 2)/q$, and therefore, by Markov's inequality $E[Z''_i] \le q/8pr$. Applying Markov's inequality again, for a random settings of the protocol's randomness, the expectation becomes larger than $q/2$ with probability at most $1/4pr$. Summarizing, by fixing the protocol's random bits, with probability at least $1 - 1/2 - 1/(4pr) \cdot (pr - 1) - 1/(4pr) \cdot (pr - 1) = 1/2 - (pr - 1)/(2pr) > 1/2 - 1/2 = 0$, we obtain a deterministic protocol that errs with probability at most 2δ, and sends the i-th message longer than $128(a_i + 2)(pr/q)^2$ with probability bounded by q. The final claim follows from the fact that all a_i are bounded by $\mathrm{ICost}_\mu(\Pi)$. □

5.2 Lower Bound for Deterministic Typically Concise Protocols

In this section, we show that a deterministic concise protocol for the Limited Pointer Chasing Equality cannot send short messages very often, unless it errs with probability $\Omega(1/n)$. The proof follows along the lines of the lower bound for Pointer Chasing due to Nisan and Wigderson [27] and its adaptation due to Guha and McGregor [16]. The main technical differences come from the fact that we want to show a lower bound for Limited Pointer Chasing Equality. First, this requires ruling out the impact of the easy case when one of the functions is t-non-injectve for large t. Second, this requires showing that with constant probability, the last player is unlikely to know what the solutions to the input instances are, and since they are independent, they will collide with probability $\Omega(1/n)$.

Lemma 13. *If $n^2 \ge 18p^2$, then any deterministic (m, q)-concise protocol for $\mathsf{LPCE}_{n,p,C_\star(1+\log n)}$, where $m \le \varepsilon n/(4p^2) - \lceil 2 \log n \rceil$, $q = \frac{1}{12p^2}$, and $\varepsilon = (48p^2)^{-3}$, errs with probability at least $1/(16n)$ on the uniform distribution over all possible inputs.*

Proof. Recall that in the $\mathsf{LPCE}_{n,p,C_\star(1+\log n)}$ problem, there are $2p$ players $P_1, \ldots, P_{2p}$, with players P_i and P_{p+i}, $1 \le i \le p$, holding functions $f_i : [n] \to [n]$ and $g_i : [n] \to [n]$, respectively. The goal of the problem is to output "1" if one of the functions is $C_\star(1 + \log n)$-non-injective or $f_1(f_2(\ldots f_p(1) \ldots)) = g_1(g_2(\ldots g_p(1) \ldots))$. Otherwise, "0" is the correct output. The players speak in order P_1 through P_{2p} and this repeats $p - 1$ times.

Let $a_{p+1} = 1$ and by induction, let $a_i = f_i(a_{i+1})$ for each $i \in [p]$. Analogously, let $b_{p+1} = 1$ and let $b_i = g_i(b_{i+1})$ for each $i \in [p]$. Unless one of the functions is $C_\star(1 + \log n)$-non-injective, the goal of the problem is to find out whether $a_1 = b_1$.

We make two modifications to the protocol:

1. We augment the (m, q)-concise protocol by simulating in parallel the following natural protocol. Initially, we append the pair (a_{p+1}, b_{p+1}) to each message until we reach the player P_p, who can compute $a_p = f_p(a_{p+1})$. Then we pass the pair (a_p, b_{p+1}) until it reaches P_{2p}, who can compute $b_p = g_p(b_{p+1})$ and pass (a_p, b_p) to the next player. In general, whenever a message (a_i, b_i) reaches P_{i-1}, it is updated to (a_{i-1}, b_i), and whenever a message (a_{i-1}, b_i) reaches P_{p+i-1}, it is updated to (a_{i-1}, b_{i-1}). This protocol finally computes (a_2, b_2). Appending the information, increases the length of each messages by $\lceil 2 \log n \rceil$. This way, we obtain a deterministic $(m + \lceil 2 \log n \rceil, q)$-concise protocol.

2. The first time a player whose function is $C_\star(1+\log n)$-non-injective is reached in the protocol, we make the player send a message longer than $m + \lceil 2 \log n \rceil$ bits. This may require modifying other messages sent by the player. We now describe how this can be done depending on the protocol's behavior.

   (a) If the player is already sending long messages in some cases, we can relabel and extend some of the long messages to accommodate the transition of some short messages into long messages. Recall that the player's function is $C_\star(1 + \log n)$-non-injective with probability at most $1/(2n^2)$. Hence, in this case, the probability of sending a long message increases by at most $1/(2n^2)$.

   (b) Likewise, if one of at least n^2 prefixes of length $\lceil 2 \log n \rceil$ is not used by the protocol at all, we can use this prefix to transmit long messages. In this case, the probability of a long message increases by at most $1/(2n^2)$ as well.

   (c) Finally, if the player does not send long messages and all prefixes of length $\lceil 2 \log n \rceil$ are used, one of these prefixes occurs with probability at most $1/n^2$. We extend all messages starting with this prefix to turn them into long messages and reduce the case to the first case. The extension increases the probability of long messages by at most $1/n^2$.

   Overall, the probability of long messages can increase by at most $1/(2n^2) + 1/n^2 \leq 2/n^2$. As a result we obtain a deterministic $(m + \lceil 2 \log n \rceil, q + 2/n^2)$-concise protocol.

Let $m' = m + \lceil 2 \log n \rceil$ and $q' = q + 2/n^2$.

From now on we think of our deterministic protocol as a decision tree of depth $2p(p-1)$. The i-th layer of nodes, $1 \leq i \leq 2p(p-1)$, corresponds to the situation when the control is passed to the player P_j, where $j \equiv i \pmod{2p}$. Each leaf in the tree is labeled with either a "0" or a "1", corresponding to the decision made by the algorithm. Each edge outgoing from nodes at layers 1 through $2p(p-1) - 1$ is labeled with the message that the corresponding player sends, given his input and the previous messages. Edges between the last two layers are not labeled, because the last player does not send a message.

We now introduce a few definitions for each node z in the decision tree:

- c_z: We set c_z to the total length of the messages sent on the path from the root to z.

- $F_1^z \times \cdots \times F_p^z \times G_1^z \times \ldots \times G_p^z$: Let $\mathcal{F}$ be the set of all functions from n to n. Since the protocol is deterministic, for each node z, the set of input functions $(f_1, \ldots, f_p, g_1, \ldots, g_p)$ for which the protocol reaches z can be described as a product $F_1^z \times \cdots \times F_p^z \times G_1^z \times \ldots \times G_p^z \subseteq \mathcal{F}^{2p}$. Note that if the node is reached then the probability of each tuple in $F_1^z \times \cdots \times F_p^z \times G_1^z \times \ldots \times G_p^z$ is identical.

- i_z and j_z: We make i_z and j_z be the indices of the last pair (a_{i_z}, b_{j_z}) sent on the path from the root to z. For the root we assume that the pair is $(a_{p+1}, b_{p+1}) = (1, 1)$, i.e., $i_{\text{root}} = j_{\text{root}} = p + 1$. Recall that for all z, $i_z \geq 2$ and $j_z \geq 2$.

- (A_z, B_z): (A_z, B_z) is a pair of random variables. Its random value is generated by selecting two functions $f_\star \in F^z_{i_z-1}$ and $g_\star \in G^z_{j_z-1}$ independently and uniformly at random and applying them to (a_{i_z}, b_{j_z}) to obtain $(f_\star(a_{i_z}), g_\star(b_{j_z}))$. (A_z, B_z) describes the possible values of the pair (a_i, b_j) if we move one step ahead in applying functions f_i and g_j, compared to the trivial algorithm that we simulate in parallel. Since the protocol is deterministic, the inputs of the players are independent, and A_z and B_z depend on inputs of disjoint sets of players, the variables A_z and B_z are independent.

We say that a node z is *confusing* if it has the following properties:

1. All messages sent on the path to z have length bounded by m'.
2. z is a leaf or for all $i \in [p]$, both $|F^z_i| \geq 2^{-2c_z}|\mathcal{F}|$ and $|G^z_i| \geq 2^{-2c_z}|\mathcal{F}|$, where c_z is the total length of the messages on the path from the root to z.
3. $H(A_z) \geq \log n - \delta$ and $H(B_z) \geq \log n - \delta$, where $\delta = \varepsilon^{2/3}$.

It is easy to see that the root of the decision tree is confusing. We now prove by induction that the probability that for a random input, the protocol reaches a non-confusing node in step i is bounded by $(i-1)\cdot(q+\frac{3}{2}n^{-2}+4\varepsilon^{1/3})$. Suppose that the claim is true for step i and let us prove it for step $i+1$. We bound the probability that a specific property is violated.

1. The probability that the first property is violated is bounded by q', because the protocol is (m', q')-concise.
2. Consider a confusing node z in step i. If $i = 2p(p-1)$, the children of z are leaves, and the property holds. So it suffices to focus on the case that $i < 2p(p-1)$. What is the probability that the second property is violated for some child w of z? Let P_j be the player in control of step i. Without loss of generality, let us assume that $1 \leq j \leq p$. Note that for all $t \in [p]$, $G^w_t = G^z_t$ and for all $t \in [p] \setminus \{j\}$, $F^w_t = F^z_t$. The property may only be violated for F^w_j. For each child w of z, let $m_w = c_w - c_z$. Let W be the random variable representing the distribution of children of z. It holds

$$\Pr\left[\frac{|F^W_j|}{|\mathcal{F}|} < 2^{-2c_W}\right] \leq \Pr\left[\frac{|F^W_j|}{|F^z_j|} < 2^{-2m_W}\right] \leq \sum_w 2^{-2m_w}$$
$$\leq \frac{1}{n^2}\sum_w 2^{-m_w} \leq \frac{1}{n^2},$$

where the second to last inequality follows from the fact that $m_w \geq 2\log n$, and the last inequality follows from Kraft's inequality. Therefore the probability that a confusing node loses the second property in the next step is bounded by $1/n^2$.

3. It remains to bound the probability that the third property is lost. Let z be a confusing node in step i and let P_j be the player in charge of this step. If neither $j = i_z - 1$ nor $j - p = j_z - 1$, then for any child w of z, $F^w_{i_z-1} = F^z_{i_z-1}$ and $G^w_{j_z-1} = G^z_{j_z-1}$. In this case the pairs of variables (A_z, B_z) and (A_w, B_w) have the same distribution and therefore the respective entropies remain the same. Consider now the case that $j = i_z - 1$. P_j computes $a_j = f_j(a_{i_z})$ and we need to bound the entropy $H(A_w)$ of A_w for all children w of z, which is essentially the entropy of a_{j-1} given all the information communicated so far. The information about f_{j-1} at each child w can be expressed as a vector $f^w_{j-1} = (f^w_{j-1}(1), f^w_{j-1}(2), \ldots, f^w_{j-1}(n))$ of random

variables in $[n]$. We have $H(A_w) = H(f^w_{j-1}(a_j))$. Moreover, it holds

$$\begin{aligned}\sum_{t\in[n]} H(f^w_{j-1}(t)) &\geq H(f^w_{j-1})\\ &= \log|F^w_{j-1}|\\ &= \log|F^z_{j-1}|\\ &\geq \log(2^{-2c_z}|\mathcal{F}|)\\ &= \log|\mathcal{F}| - 2c_z\\ &\geq n(\log n - \varepsilon).\end{aligned}$$

The first inequality above follows from subadditivity of entropy. The second and third inequalities follow from the fact that the function is uniformly distributed on $F^w_{j-1} = F^z_{j-1}$ of size bounded by the fact that z is confusing (Property 2). Finally the last inequality follows from the fact that z is confusing (Property 1), and therefore, $\varepsilon n/2 \geq 2p(p-1)\cdot m' \geq c_z$.

For t uniformly distributed on $[n]$, by Markov's inequality, we have

$$\Pr_t[H(f^w_{j-1}(t)) \leq \log n - \delta] \leq \varepsilon/\delta.$$

Unfortunately, a_j may not be uniformly distributed. However, we can exploit its high entropy, at least $\log n - \delta$. We apply Fact 1. Let S be the set of t such that $H(f^w_{j-1}(t)) \geq \log n - \delta$. We already know that $|S| \geq (1-\varepsilon/\delta)n$. Note that we can apply Fact 1, because

$$\Delta = \sqrt{\frac{4\delta}{|S|/n}} \leq \sqrt{\frac{4\delta}{1-\frac{\varepsilon}{\delta}}} \leq \sqrt{8\delta} \leq \sqrt{8}\cdot\varepsilon^{1/3} \leq \sqrt{8}\cdot 48^{-1} < 1/10.$$

The probability that a_j belongs to S is at least

$$\begin{aligned}\frac{|S|}{n}\left(1-\sqrt{\frac{4\delta}{|S|/n}}\right) &\geq \left(1-\frac{\varepsilon}{\delta}\right)\left(1-\sqrt{\frac{4\delta}{1-\frac{\varepsilon}{\delta}}}\right) \geq \left(1-\varepsilon^{1/3}\right)\left(1-\sqrt{8\delta}\right)\\ &\geq 1-(1+\sqrt{8})\varepsilon^{1/3} \geq 1-4\varepsilon^{1/3}.\end{aligned}$$

This implies that

$$\Pr_{a_j}[H(f^w_{j-1}(a_j)) \leq \log n - \delta] \leq 4\varepsilon^{1/3}.$$

The case that $j - p = j_z - 1$ is analogous, and therefore, the probability that the third property is lost in the next step is bounded by $4\varepsilon^{1/3}$.

Summarizing, the probability that moving from step i to step $i+1$, we move from a confusing node to a non-confusing node is bounded by $q' + 1/n^2 + 4\varepsilon^{1/3} = q + \frac{3}{2}n^{-2} + 4\varepsilon^{1/3}$, which finishes the inductive proof.

Overall, it follows that the protocol finishes in a non-confusing leaf with probability bounded by $2p(p-1)\cdot(q+\frac{3}{2}n^{-2}+4\varepsilon^{1/3}) \leq 2p^2\cdot q + \frac{3p^2}{n^2} + 8p^2\cdot(48p^2)^{-1} \leq 1/6+1/6+1/6 = 1/2$.

Consider now a confusing leaf z. Recall that we modified the protocol so that if one of the functions is $C_\star(1+\log n)$-non-injective a message longer than m' is transmitted. By definition, in such a case, the simulation of the protocol leads to a leaf that is not confusing. Therefore, the correct solution to an input instance that leads to z is solely based on whether $a_1 = b_1$. We know that the random variables A_z and B_z, which model a_1 and b_1, respectively, are independent and both have entropy at least $\log n - \delta$, where $\delta \leq 48^{-2}$. Observe also that $n \geq \sqrt{18p^2} \geq 4$. Hence it follows from Lemma 2 that whatever solution the protocol claims at z, be it "0" or "1", the claim is incorrect with probability at least $1/(8n)$. Overall, on all inputs the protocol has to err with probability at least $\frac{1}{2}\cdot\frac{1}{8n} = \frac{1}{16n}$. □

5.3 Proof of Lemma 6

We now combine Lemmas 12 and 13 to prove Lemma 6, the main result of Section 5.

Proof of Lemma 6. Consider any protocol protocol Π for $\mathsf{LPCE}_{n,p,C_\star(1+\log n)}$ that errs with probability at most $1/(64n)$ on μ. By Lemma 12, there is a deterministic $(m, 1/(12p^2))$-concise protocol Π' for $\mathsf{LPCE}_{n,p,C_\star(1+\log n)}$ that errs with probability at most $1/(32n)$ on μ, where $m = 128 \cdot (\mathrm{ICost}_\mu(\Pi) + 2) \cdot \left(\frac{2p^2}{(1/(12p^2))}\right)^2 = 2^{13} \cdot 3^2 \cdot p^8 \cdot (\mathrm{ICost}_\mu(\Pi) + 2)$. It follows from Lemma 13 that $m \geq \frac{n}{(48p^2)^3 \cdot 4p^2} - \lceil 2\log n \rceil = \frac{n}{2^{16} \cdot 3^3 \cdot p^8} - \lceil 2 \log n \rceil$. Therefore, $\mathrm{ICost}_\mu(\Pi) \geq \frac{n}{2^{29} \cdot 3^5 \cdot p^{16}} - \frac{\lceil 2\log n \rceil}{2^{13} \cdot 3^2 \cdot p^8} - 2$. Since this bound holds for any protocol Π that is correct with probability $1 - 1/(64n)$, this is also a lower bound for the information complexity of the problem. □

6 Step 2: Direct Sum Theorem for Pointer Chasing Equality

The following lemma is the main result of this section.

Lemma 14. *Let n, p, and t be integers such that $n^2 \geq 18p^2$ and $t \leq n/4$. Let $r = C_\star(1 + \log n)$. It holds*

$$R_{1/3}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})) = \Omega\left(\frac{tn}{p^{16}\log n}\right) - O\left(pt^2\right).$$

Before we prove it, let us first recall two classic results. First, the information complexity is a lower bound for the randomized communication complexity of a protocol that errs with the same probability.

Lemma 15 ([2, Proposition 4.3])**.** *Let $\delta \in (0,1)$. For any communication problem $\mathcal{P}$ and any distribution ψ on inputs, $R_\delta(\mathcal{P}) \geq \mathrm{IC}_{\psi,\delta}(\mathcal{P})$.*

Second, if the input distribution is a product distribution on multiple instances of a subproblem, then the total information revealed by the protocol transcript equals at least the information revealed for each of the instances.

Lemma 16 ([2, Lemma 5.1])**.** *Let $\mathcal{P}$ be a communication problem with Boolean output and let Π be a private-randomness protocol for $\mathsf{OR}_t(\mathcal{P})$ for a positive integer t. Let ψ be a distribution on inputs of $\mathcal{P}$. Let $X = (X_1, \ldots, X_t)$ be a vector of independent random variables with each distributed according to ψ. For any input x, let $\Pi(x)$ be the transcript of Π on x. Then the following inequality holds:*

$$I(X; \Pi(X)) \geq \textstyle\sum_{i=1}^t I(X_i; \Pi(X)).$$

Now we show the main ingredient, which is a proof that any correct protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ has to reveal almost as much information about each coordinate as if it was separately solving the corresponding instances of $\mathsf{LPCE}_{n,p,r}$.

Lemma 17. *Let Π be a private-randomness protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ that errs with probability at most δ. Let $X = (X_1, \ldots, X_t)$ be a random vector with each coordinate X_i independently selected from the uniform distribution μ on all possible inputs to $\mathsf{LPCE}_{n,p,r}$. Let $\Pi(x)$ be the transcript of Π on input x.*

If $r \geq C_\star \cdot (\log n + 1)$, $t \leq n/4$, and $p \leq n$, then for each $i \in [t]$, it holds

$$I(X_i; \Pi(X)) \geq \mathrm{IC}_{\mu,2\delta}(\mathsf{LPCE}_{n,p,r}) - 4pt\log n - 1.$$

Proof. For each $j \in [t]$, let $Y_j \in \{0,1\}$ be the the solution to $\mathsf{LPCE}_{n,p,r}$ on a specific coordinate X_j. The probability that $Y_j = 1$ is bounded by $1/n + 2p \cdot 1/(2n^2) = 2/n$, where the first term comes from the probability that the equality of two instances of $\mathsf{PC}_{n,p}$ holds and the other is a bound on the probability that one of the functions is r-non-injective.

Fix $i \in [t]$. By the union bound, the probability that $\bigvee_{j \neq i} Y_j = 1$ is bounded by $t \cdot 2/n \leq 1/2$. If $\bigvee_{j \neq i} Y_j = 0$, the solution to $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ on the input instance equals Y_i. Therefore, Π has to compute Y_i with probability at least $1-2\delta$, provided $\bigvee_{j\neq i} Y_j = 0$. If it erred with higher probability, it would overall err on the input instance of $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ with probability greater than δ. This implies that $I(X_i;\Pi(X)|\bigvee_{j\neq i} Y_j = 0) \geq \mathrm{IC}_{\mu,2\delta}(\mathsf{LPCE}_{n,p,r})$.

We now use $I(X_i;\Pi(X)|\bigvee_{j\neq i} Y_j = 0)$ to bound $I(X_i;\Pi(X))$ from below. By definition, we have $I(X_i;\Pi(X)) = H(X_i) - H(X_i|\Pi(X))$. Note that $H(X_i) = H(X_i|\bigvee_{j\neq i} Y_j = 0)$, because the coordinates of X_i are independent. Let us now upper bound $H(X_i|\Pi(X))$. For each transcript π and each $z\in\{0,1\}$, let $p_{z,\pi} = \Pr[\Pi(X) = \pi|\bigvee_{j\neq i} Y_j = z]$. It follows from Lemma 3 that

$$\begin{aligned}
&H(X_i \mid \Pi(X)) = E_\pi[H(X_i \mid \Pi(X)=\pi)]\\
&\leq E_\pi\Big[1+\sum_{z=0}^{1}\Pr\Big[\bigvee_{j\neq i} Y_j = z \mid \Pi(X)=\pi\Big]\cdot H\Big(X_i \mid \Pi(X)=\pi, \bigvee_{j\neq i} Y_j = z\Big)\Big]\\
&= 1+E_\pi\Bigg[\sum_{z=0}^{1} p_{z,\pi}\cdot\frac{\Pr[\bigvee_{j\neq i} Y_j = z]}{\Pr[\Pi(X)=\pi]}\cdot H\Big(X_i \mid \Pi(X)=\pi, \bigvee_{j\neq i} Y_j = z\Big)\Bigg]\\
&= 1+\sum_{z=0}^{1}\Pr\Big[\bigvee_{j\neq i} Y_j = z\Big]\cdot E_\pi\Big[\frac{p_{z,\pi}}{\Pr[\Pi(X)=\pi]}\cdot H\Big(X_i \mid \Pi(X)=\pi, \bigvee_{j\neq i} Y_j = z\Big)\Big]\\
&= 1+\sum_{z=0}^{1}\Pr\big[\bigvee_{j\neq i} Y_j = z\big]\cdot \sum_{\pi} p_{z,\pi}\cdot H\Big(X_i \mid \Pi(X)=\pi, \bigvee_{j\neq i} Y_j = z\Big)\\
&= 1+\sum_{z=0}^{1}\Pr\big[\bigvee_{j\neq i} Y_j = z\big]\cdot H\Big(X_i \mid \Pi(X), \bigvee_{j\neq i} Y_j = z\Big)\,.
\end{aligned}$$

Note that the entropy of X_i, and therefore also any conditional entropy of X_i, is always bounded by $2p\log n$. Hence

$$\begin{aligned}
H(X_i \mid \Pi(X)) &\leq 1+H\Big(X_i \mid \Pi(X), \bigvee_{j\neq i} Y_j = 0\Big)+\Pr\big[\bigvee_{j\neq i} Y_j = 1\big]\cdot 2pn\cdot\log n\\
&\leq 1+H\Big(X_i \mid \Pi(X), \bigvee_{j\neq i} Y_j = 0\Big)+4pt\log n.
\end{aligned}$$

Thus we obtain

$$\begin{aligned}
I(X_i;\Pi(X)) &= H(X_i)-H(X_i\mid\Pi(X))\\
&\geq H\Big(X_i \mid \bigvee_{j\neq i} Y_j = 0\Big) - H\Big(X_i \mid \Pi(X), \bigvee_{j\neq i} Y_j = 0\Big) - 1 - 4pt\log n\\
&\geq I\Big(X_i;\Pi(X) \mid \bigvee_{j\neq i} Y_j = 0\Big) - 1 - 4pt\log n\\
&\geq \mathrm{IC}_{\mu,2\delta}(\mathsf{LPCE}_{n,p,r}) - 1 - 4pt\log n.
\end{aligned}$$

□

We can finally prove the main lemma of this section.

Proof of Lemma 14. Let Π be a private-randomness protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ that errs with probability at most $1/(128n)$. It follows from Lemmas 16 and 17 that

$$I(X;\Pi(X)) \geq \sum_{i=1}^{t} I(X_i;\Pi(X)) \geq t\cdot\mathrm{IC}_{\mu,1/(64n)}(\mathsf{LPCE}_{n,p,r}) - 4pt^2\log n - t.$$

By definition, this quantity bounds also $\mathrm{IC}_{\mu^t,1/(128n)}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r}))$. Therefore, by Lemma 15 and Lemma 6, we get

$$R_{1/(128n)}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})) \geq \Omega\left(\frac{tn}{p^{16}}\right) - O\left(pt^2 \log n\right).$$

Via standard amplification bounds, $R_{1/(128n)}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})) \leq R_{1/3}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})) \cdot O(\log n)$, which gives us

$$R_{1/3}(\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})) = \Omega\left(\frac{tn}{p^{16}\log n}\right) - O\left(pt^2\right). \qquad \square$$

7 Step 3: Reduction to Set Chasing Intersection

We give a reduction showing that under specific conditions, a protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ can be used to create a communication protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$.

Lemma 18. *Let n, p, r, and t be positive integers such that $t^{2p}r^{p-1} \leq n/10$. If there is a communication protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ that uses C bits of communication and errs with probability at most $1/10$, then there is a public-randomness communication protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ that uses $C + 2p$ bits of communication and errs with probability at most $2/10$.*

Proof. Consider an instance of $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$. There are $2p$ players, who have t instances of $\mathsf{LPCE}_{n,p,r}$. Each instance of $\mathsf{LPCE}_{n,p,r}$ consists of two instances of $\mathsf{PC}_{n,p}$. Let $f_{i,j}$ and $g_{i,j}$, where $1 \leq i \leq p$ and $1 \leq j \leq t$, be the functions that describe these two instances. For each $i \in [p]$, player i knows $f_{i,j}$ and player $p+i$ knows $g_{i,j}$. If any of the functions $f_{i,j}$ or $g_{i,j}$ is r-non-injective, then the solution to the problem is 1. The players can check if this is the case in one round of communication with each player communicating only one bit. It therefore suffices to show a protocol that solves $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$, i.e., computes

$$\bigvee_{j=1}^{t} \Big(f_{1,j}(f_{2,j}(\ldots f_{p,j}(1)\ldots)) = g_{1,j}(g_{2,j}(\ldots g_{p,j}(1)\ldots))\Big),$$

using C bits communication, under the assumption that no $f_{i,j}$ or $g_{i,j}$ is r-non-injective. To this end, we show a randomized reduction of this problem to $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$.

First, using common randomness, the players select random permutations $\pi_{i,j}, \rho_{i,j} : [n] \to [n]$ for $1 \leq i \leq p$ and $1 \leq j \leq t$. Permutations are selected independently, except that $\pi_{1,j} = \rho_{1,j}$ for all $1 \leq j \leq t$. Furthermore they are generated using public randomness, so they are known to all players. (For functions $f : A \to B$ and $g : B \to C$, we write $g \circ f$ to denote the function from A to C such that $(g \circ f)(x) = g(f(x))$ for all $x \in A$.) For all $1 \leq j \leq t$, let

$$f'_{p,j} = \pi_{p,j} \circ f_{p,j} \quad \text{and} \quad g'_{p,j} = \rho_{p,j} \circ g_{p,j}.$$

For all $1 \leq i \leq p-1$ and $1 \leq j \leq t$, let

$$f'_{i,j} = \pi_{i,j} \circ f_{i,j} \circ \pi^{-1}_{i+1,j} \quad \text{and} \quad g'_{i,j} = \rho_{i,j} \circ g_{i,j} \circ \rho^{-1}_{i+1,j}.$$

It is easy to see that an instance of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ with $f'_{i,j}$ and $g'_{i,j}$ is equivalent to the original instance with $f_{i,j}$ and $g_{i,j}$. The permutations randomly relabel intermediate and final values with final values relabeled in the same way on both sides.

We construct an instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ by giving the i-th player, $1 \leq i \leq p$, a function $f^\star_i : [n] \to 2^{[n]}$ such that for any $x \in [n]$,

$$f^\star_i(x) = \{f'_{i,j}(x) : 1 \leq j \leq t\}$$

and by giving the $p+i$-th player, $1 \leq i \leq p$, a function $g^\star_i : [n] \to 2^{[n]}$ such that for any $x \in [n]$,

$$g^\star_i(x) = \{g'_{i,j}(x) : 1 \leq j \leq t\}.$$

The goal in this instance is to compute

$$\left(\overrightarrow{f_1^\star}(\overrightarrow{f_2^\star}(\ldots\overrightarrow{f_p^\star}(\{1\})\ldots)) \cap \overrightarrow{g_1^\star}(\overrightarrow{g_2^\star}(\ldots\overrightarrow{g_p^\star}(\{1\})\ldots))\right) \neq \emptyset.$$

The instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ that we have just defined can be seen as stacking mappings from different instances of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ on top of each other. Instead of following a single function $f'_{i,j}$ or $g'_{i,j}$ for a given instance j, we follow all of them simultaneously, obtaining subsets of $[n]$ instead of just single values in $[n]$.

Let us show a likely correspondence between the new instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ and the original instance of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$. First, if the result of solving the instance of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ is 1, then clearly, by following the mappings from the instance of $\mathsf{EQUAL}(\mathsf{PC}_{n,p})$ resulting in 1, we also obtain an element that belongs to the intersection of two resulting sets in $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$.

Consider the case that the result of solving the instance of $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ is 0. We bound the probability that the sets appearing in the instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ intersect. Each element of these two sets can be expressed as

$$f'_{1,a_1}(f'_{2,a_2}(\ldots f'_{p,a_p}(1)\ldots))$$

or

$$g'_{1,b_1}(g'_{2,b_2}(\ldots g'_{p,b_p}(1)\ldots)),$$

respectively, where the sequences $a_1, \ldots, a_p$ and $b_1, \ldots, b_p$ describe which of the instances the mapping is followed. There are t^{2p} different pairs of such sequences. What is the probability that we obtain the same value for a specific pair of sequences? We want to show that this probability is bounded by r^{p-1}/n. If $a_1 = \ldots = a_p = b_1 = \ldots = b_p$, then we obtain different values, because the a_1-th instance in $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ results in 0. Suppose now that it is not the case that $a_1 = \ldots = a_p = b_1 = \ldots = b_p$. If $a_1 \neq b_1$, then the probability that we obtain the same value is exactly $1/n$, because the final values are created by two independent random permutations π_{1,a_1} and ρ_{1,b_1}. If $a_1 = b_1$, let k be the lowest number greater than 1 such that $a_k \neq a_{k-1}$ or $b_k \neq b_{k-1}$. Since the functions $f'_{1,a_1} \circ \ldots \circ f'_{k-1,a_1}$ and $g'_{1,a_1} \circ \ldots \circ g'_{k-1,a_1}$ are not r^{p-1}-non-injective and are applied to two values randomly distributed by π_{r,a_r} and ρ_{r,b_r}, the probability of collision is at most r^{p-1}/n. By the linearity of expectation, the expected size of the intersection between the two sets in the instance of $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ is bounded by $t^{2p} \cdot r^{p-1}/n \leq 1/10$. By Markov's inequality, the probability that the intersection is nonempty is bounded by $1/10$, so the probability that the reduction fails is bounded by $1/10$. Therefore, if we have a communication protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ that errs with probability at most $1/10$, we can use this protocol to obtain a public-randomness protocol for $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ that errs with probability at most $2/10$, provided no function in $\mathsf{OR}_t(\mathsf{EQUAL}(\mathsf{PC}_{n,p}))$ is r-non-injective. □

8 Proof of Main Tool (Theorem 4)

We now combine the results of Steps 1,2,3 to conclude our main communication complexity lower bound (Theorem 4 from Section 4).

Proof of Theorem 4. Let $r = C_\star(1+\log n)$ and let $t = \left\lfloor \frac{n^{1/(2p)}}{\sqrt{10r}} \right\rfloor$. Due to the result of Newman [26], we know that every protocol with public randomness can be simulated using private randomness if we allow for using additional $O(\log(\text{input-size-in-bits}))$ communication bits and for increasing the probability of error by an arbitrarily small constant. By combining this fact with Lemma 14 (which can be applied for n greater than some constant), we find out that any public-randomness protocol for $\mathsf{OR}_t(\mathsf{LPCE}_{n,p,r})$ that errs with probability at most $2/10$ has to use at least

$$\Omega\left(\frac{tn}{p^{16}\log n}\right) - O\left(pt^2\right) - O(\log(t \cdot 2p \cdot n \cdot \log n)) = \Omega\left(\frac{tn}{p^{16}\log n}\right) - O\left(pt^2 + \log n\right)$$

bits of communication. Note that for n greater than some positive constant, the first term dominates the second, so we can express the lower bound as simply $\Omega\left(\frac{tn}{p^{16}\log n}\right)$.

Note that for our setting of t, $t^{2p} r^{p-1} \leq n/10$. We can therefore apply Lemma 18. We learn that any communication protocol for $\mathsf{INTERSECT}(\mathsf{SC}_{n,p})$ that errs with probability at most $1/10$ has to use at least $\Omega\left(\frac{tn}{p^{16}\log n}\right) - 2p$ bits of communication. As before, the first term dominates the second for sufficiently large n and the lower bound becomes

$$\Omega\left(\frac{n^{1+1/(2p)}}{p^{16} \cdot \log^{3/2} n}\right).$$

□

References

[1] Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In *ICALP (2)*, pages 526–538, 2011.

[2] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *FOCS*, pages 209–218, 2002.

[3] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In Leonard J. Schulman, editor, *STOC*, pages 67–76. ACM, 2010.

[4] Mark Braverman. Interactive information complexity. In *STOC*, pages 505–524, 2012.

[5] Mark Braverman and Anup Rao. Information equals amortized communication. In Rafail Ostrovsky, editor, *FOCS*, pages 748–757. IEEE, 2011.

[6] Mark Braverman and Omri Weinstein. A discrepancy lower bound for information complexity. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 459–470. Springer, 2012.

[7] Amit Chakrabarti, Ranganath Kondapally, and Zhenghui Wang. Information complexity versus corruption and applications to orthogonality and Gap-Hamming. In *APPROX-RANDOM*, pages 483–494, 2012.

[8] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

[9] Carsten Damm, Stasys Jukna, and Jiri Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Computational Complexity*, 7(2):109–127, 1998.

[10] Pavol Duris, Zvi Galil, and Georg Schnitger. Lower bounds on communication complexity. *Inf. Comput.*, 73(1):1–22, 1987.

[11] Sebastian Eggert, Lasse Kliemann, Peter Munstermann, and Anand Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1-2):490–508, 2012.

[12] Leah Epstein, Asaf Levin, Julián Mestre, and Danny Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011.

[13] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.

[14] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008.

[15] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *SODA*, pages 468–485, 2012.

[16] Sudipto Guha and Andrew McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. Comput.*, 38(5):2044–2059, 2009.

[17] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *IEEE Conference on Computational Complexity*, 2013.

[18] Monika Rauch Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. Technical Report 1998-011, DEC System Research Center, 1998.

[19] Rahul Jain, Attila Pereszlényi, and Penghui Yao. A direct product theorem for bounded-round public-coin randomized communication complexity. *CoRR*, abs/1201.1666, 2012.

[20] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. Privacy and interaction in quantum communication complexity and a theorem about the relative entropy of quantum states. In *FOCS*, pages 429–438, 2002.

[21] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In *ICALP*, pages 300–315, 2003.

[22] Michael Kapralov. Better bounds for matchings in the streaming model. In *SODA*, pages 1679–1697, 2013.

[23] Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:38, 2012.

[24] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, pages 231–242, 2012.

[25] Andrew McGregor. Finding graph matchings in data streams. In *APPROX-RANDOM*, pages 170–181, 2005.

[26] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

[27] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1):211–219, 1993.

[28] Christos H. Papadimitriou and Michael Sipser. Communication complexity. *J. Comput. Syst. Sci.*, 28(2):260–269, 1984.

[29] Stephen Ponzio, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. The communication complexity of pointer chasing. *J. Comput. Syst. Sci.*, 62(2):323–355, 2001.

[30] Ran Raz and Avi Wigderson. Probabilistic communication complexity of Boolean relations (extended abstract). In *FOCS*, pages 562–567, 1989.

[31] Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. Estimating PageRank on graph streams. *J. ACM*, 58(3):13, 2011.

[32] Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de