Deterministic Communication vs. Partition Number

Mika Göös
Harvard University

Toniann Pitassi
University of Toronto

Thomas Watson
University of Memphis

September 5, 2017

Abstract
We show that deterministic communication complexity can be superlogarithmic in the partition number of the associated communication matrix. We also obtain near-optimal deterministic lower bounds for the Clique vs. Independent Set problem, which in particular yields new lower bounds for the log-rank conjecture. All these results follow from a simple adaptation of a communication-to-query simulation theorem of Raz and McKenzie (Combinatorica 1999) together with lower bounds for the analogous query complexity questions.

1 Introduction

The partition number of a two-party function $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ is defined by

$$\chi(F) := \chi_1(F) + \chi_0(F)$$

where $\chi_i(F)$ is the least number of rectangles (sets of the form $A \times B$ where $A \subseteq \mathcal{X}, B \subseteq \mathcal{Y}$) needed to partition the set $F^{-1}(i)$. Yao [Yao79] observed that $\log \chi(F)$ is a lower bound on the deterministic communication complexity of $F$ and inquired about the exact relationship. For upper bounds, it is known that $O(\log^2 \chi(F))$ bits [AUY83], or even $O(\log^2 \chi_1(F))$ bits [Yan91], suffice.

Our results are as follows—here the notation $\tilde{\Omega}(m)$ hides factors polylogarithmic in $m$.

**Theorem 1.** There is an $F$ with deterministic communication complexity $\tilde{\Omega}(\log^{1.5} \chi(F))$.

**Theorem 2.** There is an $F$ with deterministic communication complexity $\tilde{\Omega}(\log^2 \chi_1(F))$.

**Theorem 1** implies that the logarithm of the partition number does not characterize (up to constant factors) deterministic communication complexity, which solves an old problem [KN97, Open Problem 2.10]. The previous best lower bound in this direction was about $2 \cdot \log \chi(F)$ due to Kushilevitz, Linial, and Ostrovsky [KLO99]. In this work, we show—maybe surprisingly—that superlogarithmic lower bounds can be obtained using known techniques!

**Theorem 2** is essentially tight in view of the upper bound $O(\log^2 \chi_1(F))$ mentioned above. A recent work [Göö15] exhibited a different $F$ with $\Omega(\log^{1.128} \chi_1(F))$ conondeterministic communication complexity (i.e., the logarithm of the least number of rectangles needed to cover the set $F^{-1}(0)$);

---

Supported by funding from NSERC and from NSF grant CCF-1657377. A preliminary version of this work was published as [GPW15].
this is quantitatively weaker than Theorem 2 and hence the two results are incomparable. The question about the relationship between \( \log \chi_1(F) \) and deterministic communication complexity is sometimes referred to as the Clique vs. Independent Set problem; see [Juk12, §4.4] for an excellent overview. In particular, Theorem 2 implies that there exists a graph on \( n \) nodes for which the Clique vs. Independent Set problem (Alice is given a clique and Bob is given an independent set: do they intersect?) requires \( \Omega(\log^2 n) \) communication. (The upper bound \( O(\log^2 n) \) holds for all graphs.) Theorem 2 also gives improved lower bounds for the log-rank conjecture [LS88] (see [Lov14] for a survey): Viewing rectangles as all-1 submatrices, we have \( \chi_1(F) \geq \text{rank}(F) \) where the rank is over the reals. Hence Theorem 2 implies a communication lower bound of \( \Omega(\log^2 \text{rank}(F)) \). The previous record was \( \Omega(\log^{1.63} \text{rank}(F)) \) due to Kushilevitz [Kus94].

1.1 Our approach

We follow a recurring theme (e.g., [NW95, RM99, SZ09, She11, HN12, CLRS16, GLM^16, LRS15]):

Instead of proving an ad hoc communication lower bound directly, we prove a lower bound in the simpler-to-understand world of query complexity [BdW02], and then “lift” the result over to the world of communication complexity.

The general idea is to start with a boolean function \( f : \{0,1\}^n \rightarrow \{0,1\} \) (called the outer function) and then study a composed function \( F := f \circ g^n \) where \( g : \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\} \) is a small two-party function (called the gadget). More precisely, the communication problem is to compute, on input \( x \in \mathcal{X}^n \) to Alice, and \( y \in \mathcal{Y}^n \) to Bob, the output

\[
F(x, y) := f(g(x_1, y_1), \ldots, g(x_n, y_n)).
\]

**Deterministic simulation.** We use tools that were introduced already in 1997 by Raz and McKenzie [RM99] (building on [EIRS01]). They proved a simulation theorem that converts a deterministic protocol for \( F := f \circ g^n \) (where \( f \) is arbitrary but the gadget \( g \) is chosen carefully) into a deterministic decision tree for \( f \). Unfortunately, their result was originally formulated only in case \( f \) was a certain “structured” search problem (canonical search problem associated with a DNF tautology), and this is how their result has been applied subsequently [BEGJ00, Joh01]. However, we observe that, with minor modifications, their proof actually works without any assumptions on \( f \). Such a simulation theorem (for functions \( f \)) was conjectured in [Dru09]. We provide (in Section 3) a self-contained and streamlined exposition (including some simplifications) of the following version of the Raz–McKenzie result—here \( \text{P}^{cc}(F) \) denotes the deterministic communication complexity of \( F \) and \( \text{P}^{dt}(f) \) denotes the deterministic decision tree complexity of \( f \).

**Theorem 3 (Simulation Theorem).** There is a gadget \( g : \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\} \) where the size of Alice’s input is \( \log |\mathcal{X}| = \Theta(\log n) \) bits such that for all \( f : \{0,1\}^n \rightarrow \{0,1\} \) we have

\[
\text{P}^{cc}(f \circ g^n) = \text{P}^{dt}(f) \cdot \Theta(\log n).
\]

The gadget in the above can be taken to be the usual indexing function \( g : [m] \times \{0,1\}^m \rightarrow \{0,1\} \) where \( m := \text{poly}(n) \) and \( g(x, y) := y_x \). The upper bound in Theorem 3,

\[
\text{P}^{cc}(f \circ g^n) \leq \text{P}^{dt}(f) \cdot O(\log n), \tag{1}
\]

follows simply because a communication protocol can always simulate a decision tree for \( f \) with an overhead of factor \( \text{P}^{cc}(g) \leq \lceil \log m \rceil + 1 = \Theta(\log n) \). Indeed, whenever the decision tree queries the
i-th input bit of $f$, Alice and Bob exchange $\Theta(\log n)$ bits to compute the output $g(x_i, y_i)$ of the i-th gadget. The nontrivial part of Theorem 3 is to show that this type of protocol is optimal: there are no shortcuts to computing $f \circ g^n$ other than to “query” individual input bits of $f$ in some order.

**Nondeterministic models.** Recall that a nondeterministic protocol (e.g., [KN97, Juk12]) is a protocol that is allowed to make guesses—an input is accepted iff there is at least one accepting computation. Combinatorially, a nondeterministic protocol for $F$ of communication cost $k$ can be visualized as a covering of the set $F^{-1}(1)$ using at most $2^k$ (possibly overlapping) rectangles. Thus, the nondeterministic communication complexity of $F$, denoted $NP^{cc}(F)$ in analogy to the classical (Turing machine) complexity class $NP$, is just the logarithm of the least number of rectangles needed to cover $F^{-1}(1)$. A nondeterministic protocol is unambiguous if for each input, there is at most one accepting computation. Combinatorially, this means that the associated rectangles covering $F^{-1}(1)$ do not overlap. Hence we use the notation $UP^{cc}(F) := \lceil \log \chi_1(F) \rceil$ in analogy to the classical class $UP$. We also define $\text{coUP}^{cc}(F) := \lceil \log \chi_0(F) \rceil$, and, using the shorthand $2UP := UP \cap \text{coUP}$, we define the two-sided measure $2UP^{cc}(F) := \lceil \log \chi(F) \rceil \in \max\{UP^{cc}(F), \text{coUP}^{cc}(F)\} + \Theta(1)$.

Analogously, a nondeterministic decision tree (e.g., [Juk12, §14.2]) is a decision tree that is allowed to make guesses. Formally, we treat a nondeterministic decision tree for $f$ as a collection of l-certificates (accepting computations), that is, partial assignments to variables of $f$ that force the output of the function to be 1; the cost is the maximum number of variables fixed by a partial assignment. In other words, a nondeterministic decision tree is just a DNF formula; the cost is the maximum width of its terms. We denote by $NP^{dt}(f)$ the minimum cost of a nondeterministic decision tree for $f$, that is, its DNF width. A nondeterministic decision tree is unambiguous if for each input, there is at most one accepting certificate. We denote by $UP^{dt}(f)$ the minimum cost of an unambiguous decision tree for $f$. We also let $\text{coUP}^{dt}(f) := UP^{dt}(-f)$ and $2UP^{dt}(f) := \max\{UP^{dt}(f), \text{coUP}^{dt}(f)\}$.

**Communication ↔ query.** Generalizing (1), it is straightforward to check that a communication protocol can simulate a corresponding type of decision tree also in the case of our nondeterministic models. That is, for any $f: \{0,1\}^n \to \{0,1\}$ and for the gadget $g$ from Theorem 3 we have

\[
C^{cc}(f \circ g^n) \leq C^{dt}(f) \cdot O(\log n) \quad \forall C \in \{2UP, UP\}.
\] (2)

(It is not known whether the corresponding lower bounds hold above—we conjecture they do—but luckily we need only the upper bounds in this work.)

We can rephrase our communication results with the new notation defined above.

**Theorem 1 (Rephrased).** There is an $F$ such that $P^{cc}(F) \geq \tilde{\Omega}(2UP^{cc}(F)^{1.5})$.

**Theorem 2 (Rephrased).** There is an $F$ such that $P^{cc}(F) \geq \tilde{\Omega}(UP^{cc}(F)^2)$.

Our goal is to prove analogous query complexity separations (in Section 2):

**Theorem 4.** There is an $f$ such that $P^{dt}(f) \geq \tilde{\Omega}(2UP^{dt}(f)^{1.5})$.

**Theorem 5.** There is an $f$ such that $P^{dt}(f) \geq \tilde{\Omega}(UP^{dt}(f)^2)$.

Theorems 1–2 can now be derived by simply applying Theorem 3 and the upper bounds (2) to Theorems 4–5. We only add that the functions in Theorems 4–5 will actually satisfy $P^{dt}(f) = n^{\Theta(1)}$ and hence the factor $\Theta(\log n)$ overhead that is introduced by the gadget gets hidden in our $\Omega$-notation.
A few comments about Theorems 4–5 are in order. Firstly, Savický [Sav02] and Belovs [Bel06] have previously exhibited a function with $P^{dt}(f) \geq \Omega(2UP^{dt}(f)^{1.261})$. This means that a quantitatively weaker (but still superlogarithmic) version of Theorem 1 follows already by combining the Savický–Belovs result with Theorem 3. Secondly, it is not hard to see that $UP^{dt}(f) \geq \deg(f)$ where $\deg(f)$ is the minimum degree of a multilinear real polynomial that agrees with $f$ on boolean inputs. (The communication analogue of this inequality, namely $UP^{cc}(F) \geq \log \text{rank}(F)$, was discussed above.) Consequently, Theorem 5 gives the largest known gap between $P^{dt}(f)$ and $\deg(f)$. The previous record was $P^{dt}(f) \geq \Omega(\deg(f)^{1.63})$ by Kushilevitz [Kus94], and the current best upper bound in this context is $P^{dt}(f) \leq O(\deg(f)^{3})$ for all $f$ by Midrijānis [Mid04].

The work [Göö15] also uses the “query separation plus simulation theorem” approach to exhibit an $F$ with $coNP^{cc}(F) \geq \Omega(U^{cc}(F)^{1.128})$. The query separation in that paper involves a recursive composition that is intricate and delicate, due to the one-sided nature of $coNP$ and $UP$. In contrast, our query separation is direct (without recursive composition) and much simpler to prove. However, the deterministic simulation theorem we employ is a fair bit more complicated to prove than the conondeterministic simulation theorem used in [Göö15] (which is a relatively simple special case of a general simulation theorem from [GLM+16]).

2 Query Separations

In proving the query complexity separations it is convenient to work with functions $f: \Sigma^{n} \rightarrow \{0, 1\}$ that have a larger-than-boolean input alphabet $\Sigma$. For such functions the understanding is that it still costs one query for a decision tree to learn a particular input variable. At the end, we may always convert such an $f$ back to a boolean function $f \circ h^{n}$ where $h: \{0, 1\}^{\lceil \log |\Sigma| \rceil} \rightarrow \Sigma$ is some surjection. The following trivial bounds suffice for us:

$$C^{dt}(f) \leq C^{dt}(f \circ h^{n}) \leq C^{dt}(f) \cdot [\log |\Sigma|], \quad \forall C \in \{P, 2UP, UP\}. \quad (3)$$

We start with the proof of Theorem 5 since the proof of Theorem 4 uses Theorem 5 (as a black box).

2.1 Proof of Theorem 5

Motivating example. Let $n := k^{2}$, and consider the function $f: \{0, 1\}^{k \times k} \rightarrow \{0, 1\}$ defined on boolean matrices $M \in \{0, 1\}^{k \times k}$ such that $f(M) = 1$ iff $M$ contains a unique all-1 column. We claim that

$$NP^{dt}(f) \leq 2k - 1,$$
$$P^{dt}(f) \geq k^{2}.$$

For the upper bound, consider 1-certificates that read the unique all-1 column and a single 0-entry from each of the other columns. (Note that this collection of certificates is not unambiguous!) For the lower bound, it suffices to give an adversary argument (see, e.g., [Juk12, §14]), that is, a strategy to answer queries made by a decision tree such that even after $k^{2} - 1$ queries, the output of the function is not yet determined. Here is the strategy: Suppose the decision tree queries $M_{ij}$. If $M_{ij}$ is the last unqueried entry in the $j$-th column, answer $M_{ij} = 0$. Otherwise answer $M_{ij} = 1$. It is straightforward to check that this strategy forces the decision tree to query all of the entries.
Actual gap example. We modify the function described above with the goal of establishing
\[
\text{UP}^{dt}(f) \leq 2k - 1,
\]
\[
\text{P}^{dt}(f) \geq k^2.
\]
The modified function, which we still call \(f\), has input variables that take on values from the alphabet \(\Sigma := \{0, 1\} \times ([k] \times [k] \cup \{\bot\})\). Here \([k] \times [k] \cup \{\bot\}\) is a set of pointer values, where we interpret an entry \(M_{ij} = (m_{ij}, p_{ij}) \in \Sigma\) as pointing to another entry \(M_{p_{ij}}\) given that \(p_{ij} \neq \bot\). If \(p_{ij} = \bot\) then we have a null pointer. We define the function \(f: \Sigma^{k \times k} \rightarrow \{0, 1\}\) by describing an unambiguous decision tree computing it. (We give an “algorithmic” definition rather than writing a list of certificates.)

Unambiguous decision tree: Nondeterministically guess a column index \(j \in [k]\). Read the entries \(M_{ij} = (m_{ij}, p_{ij})\) for \(i \in [k]\) while checking that \(m_{ij} = 1\) for all \(i\) and that \(p_{ij} \neq \bot\) for at least one \(i\). Let \(i\) be the first index for which \(p_{ij} \neq \bot\). Next, iteratively follow pointers for \(k - 1\) steps starting at \((i_1, j_1) := p_{ij}\). Namely, at the \(s\)-th step, read \(M_{i_s,j_s}\) and if \(s \leq k - 2\) then check that \(p_{i_s,j_s} \neq \bot\) and define \((i_{s+1}, j_{s+1}) := p_{i_s,j_s}\). Finally, check that the resulting sequence \((i_1, j_1), \ldots, (i_{k-1}, j_{k-1})\) visits all but the \(j\)-th column (i.e., \(\{j_1, \ldots, j_{k-1}\} = [k] \setminus \{j\}\)) and that \(m_{i_s,j_s} = 0\) for all \(s \in [k - 1]\).

Thus the upper bound holds by construction. For the lower bound, we use the below strategy; here a query to an entry \(M_{ij}\) is called critical if \(M_{ij}\) is the last unqueried entry in its column.

Adversary strategy: Always answer queries with \((1, \bot)\) unless the query is critical. On the first critical query, answer \((0, \bot)\). On subsequent critical queries, answer \((0, p)\) where \(p \in [k] \times [k]\) points to where the previous critical query took place.

The function value remains undetermined after \(k^2 - 1\) queries, because we can answer the last \((k^2\)-th\) query with \((0, \bot)\) to make the function evaluate to 0, or with \((1, p)\), where \(p\) is as above, to make the function evaluate to 1. This proves \(\text{P}^{dt}(f) \geq \Omega(\text{UP}^{dt}(f)^2)\) for a function with a non-boolean alphabet. If we convert \(f\) into a boolean function \(f' := f \circ h^n\) (where \(n := k^2\)) as in (3) we end up with the claimed gap \(\text{P}^{dt}(f') \geq \tilde{\Omega}(\text{UP}^{dt}(f')^2)\) since the conversion introduces only some \([\log |\Sigma|] = \Theta(\log n)\) factors.

2.2 Proof of Theorem 4

Let \(g\) be given by Theorem 5 such that \(\text{P}^{dt}(g) = \tilde{\Theta}(q^2)\) where \(q := \text{UP}^{dt}(g)\). We define \(f := \text{AND} \circ g^q\), that is, \(f(z_1, \ldots, z_q) = 1\) iff \(g(z_i) = 1\) for all \(i \in [q]\). We claim that
\[
2\text{UP}^{dt}(f) \leq \tilde{O}(q^2),
\]
\[
\text{P}^{dt}(f) \geq \tilde{\Omega}(q^3).
\]
For the upper bound, an unambiguous certificate for an input \(z\) will contain unambiguous 1-certificates for \(g(z_i) = 1\) for all \(i \in [\ell - 1]\) where \(\ell\) is the least index such that \(g(z_\ell) = 0\), or \(\ell := q + 1\) if no such index exists. If \(\ell \leq q\) we also include an unambiguous 0-certificate for \(g(z_\ell) = 0\) that just mimics the execution of an optimal decision tree for \(g\) on input \(z_\ell\). In other words, we use the fact that \(\text{coUP}^{dt}(g) \leq \text{P}^{dt}(g)\). The cost is at most \((\ell - 1) \cdot \text{UP}^{dt}(g) + \text{P}^{dt}(g) \leq \tilde{O}(q^2)\). For the lower bound, we have \(\text{P}^{dt}(\text{AND} \circ g^q) = \text{P}^{dt}(\text{AND}) \cdot \text{P}^{dt}(g) = q \cdot \tilde{\Theta}(q^2) = \tilde{\Theta}(q^3)\) by the basic fact (e.g., [Sav02, Lemma 3.2]) that \(\text{P}^{dt}\) behaves multiplicatively with respect to composition.
3 Raz–McKenzie Simulation

The goal of this section is to give a self-contained, streamlined, and somewhat simplified proof of the Simulation Theorem that works without any assumptions on the outer function

\[ f : \{0, 1\}^N \rightarrow \{0, 1\}. \]

(Here we use \( N \) for the input length instead of \( n \), which we reserve for later use.) In fact, \( f \) can be taken to be anything: a partial function, a search problem (a general relation), or have a non-boolean codomain. However, we stick with the boolean function case for concreteness.

The gadget \( g : [m] \times \{0, 1\}^m \rightarrow \{0, 1\} \), where \( m := N^{20} \), is chosen to be the indexing function defined by \( g(x, y) := y_{k} \). Recall that for the composed function \( F := f \circ g^N \), Alice’s input is \( x := (x_1, \ldots, x_N) \in [m]^N \) and Bob’s input is \( y := (y_1, \ldots, y_N) \in ([0, 1]^m)^N \). We denote by \( z_i := g(x_i, y_i) \) the \( i \)-th input bit of \( f \) so that \( F(x, y) := f(z_1, \ldots, z_N) \).

We prove the nontrivial part of the Simulation Theorem, namely the lower bound

\[ p_{cc}(f \circ g^N) \geq p_{dt}(f) \cdot \Omega(\log m). \]

3.1 High-level overview

Once and for all, we fix a deterministic protocol for \( F := f \circ g^N \) of communication cost \( k \leq o(N \cdot \log m) \). The basic strategy is to use the protocol to build a decision tree of cost \( O(k / \log m) \) for evaluating the outer function \( f \) on an unknown input \( z \in \{0, 1\}^N \). The simulation algorithm proceeds in iterations, where in each iteration we either descend one level in the communication protocol tree (by making the protocol send a bit), or descend one level in the decision tree (by querying a bit of \( z \)). To show the simulation is correct, we maintain invariants ensuring that when we reach a leaf in the protocol tree, the value it outputs must be the correct value of \( f(z) \) (hence we can make the current node in the decision tree a leaf). To show the simulation is efficient, we use a potential function argument showing that in each “communication iteration” the potential increases by at most \( O(1) \), and in each “query iteration” the potential decreases by at least \( \Omega(\log m) \), and hence the number of query iterations is at most \( O(k / \log m) \) since there are at most \( k \) communication iterations.

In a little more detail, let \( R_v \) denote the rectangle associated with the current node \( v \) in the communication protocol tree. The simulation maintains a “cleaned up” subrectangle \( A \times B \subseteq R_v \) with the property that the set of all outputs of \( g^N \) over points in \( A \times B \) is exactly the set of all possible \( z \)-s that are consistent with the results of the queries made so far. This ensures the correctness when we reach a leaf. The analysis has two key lemmas: the Thickness Lemma helps us update \( A \times B \) in a communication iteration, and the Projection Lemma helps us update \( A \times B \) in a query iteration.

To determine which type of iteration should be next, we examine, for each unqueried coordinate, how predictable it is (in some sense) from the values of the other unqueried coordinates of \( g^N \) within \( A \times B \). If no coordinate is too predictable, then it is “safe” to have a communication iteration; the protocol partitions the rectangle \( R_v \) into two parts, and we restrict to the part that is “bigger” (from the perspective of the unqueried coordinates), then use the Thickness Lemma to do some further clean-up that restores our invariants. On the other hand, if say the \( i \)-th coordinate is too predictable from the others, then its value (within \( A \times B \)) is in danger of becoming a function of the values of the other coordinates (which would violate our invariants). In this case, we query \( z_i \) while we are still able to accommodate either possible value for it (which might become impossible
that \(\beta\) U rectangle

Lemma 7 (Projection Lemma).

We describe our notation and state the two key lemmas in Section 3.2. Then we describe the simulation algorithm itself in Section 3.3 and analyze it in Section 3.4. Finally, we provide the proofs of the two key lemmas in Section 3.5 and Section 3.6.

3.2 Notation and lemmas

For a node \(v\) in the communication protocol tree, let \(R_v := X^v \times Y^v\) denote its associated rectangle, let \(X^{v,b} \subseteq X^v\) be the set of Alice’s inputs on which the bit \(b\) would be sent (if Alice sends), and let \(Y^{v,b} \subseteq Y^v\) be the set of Bob’s inputs on which the bit \(b\) would be sent (if Bob sends).

Supposing \(A \subseteq [m]^n\) and \(B \subseteq (\{0, 1\}^m)^n\) for some \(n \leq N\), we make the following definitions.

- **Size of sets:** Let \(\alpha(A)\) be such that \(|A| = 2^{-\alpha(A)} \cdot |[m]^n|\), and let \(\beta(B)\) be such that \(|B| = 2^{-\beta(B)} \cdot |(\{0, 1\}^m)^n|\) (assuming \(|A|, |B| > 0\).

- **Projections:** If \(I \subseteq [n]\) then let \(A_I := \{(x_i)_{i \in I} : (x_1, \ldots, x_n) \in A\text{ for some } (x_j)_{j \in [n] \setminus I}\} \subseteq [m]^{|I|}\) be the projection of \(A\) onto the coordinates in \(I\), and similarly \(B_I := \{(y_i)_{i \in I} : (y_1, \ldots, y_n) \in B\text{ for some } (y_j)_{j \in [n] \setminus I}\} \subseteq (\{0, 1\}^m)^{|I|}\).

- **Pruning:** If \(U \subseteq [m], V \subseteq \{0, 1\}^m\), and \(i \in [n]\), then let \(A^{i,U} := \{x \in A : x_i \in U\}\) and \(B^{i,V} := \{y \in B : y_i \in V\}\).

- **Auxiliary graph:** If \(i \in [n]\) then let Graph\(_i\)(\(A\)) be the bipartite graph defined as follows. The left nodes are \([m]\), the right nodes are \([m]^{n-1}\), and each tuple \(x := (x_1, \ldots, x_n) \in A\) is viewed as an edge between the left node \(x_i\) and the right node \((x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)\). Note that \(A_{[n] \setminus \{i\}}\) is the set of nonzero-degree right nodes.

- **Average/minimum degree:** Let \(\text{AvgDeg}_i(A) := |A|/|A_{[n] \setminus \{i\}}|\) and \(\text{MinDeg}_i(A)\) be, respectively, the average and minimum degrees of a nonzero-degree right node in Graph\(_i\)(\(A\)).

- **Thickness:** We say \(A\) is **thick** iff \(\text{MinDeg}_i(A) \geq m^{17/20}\) for all \(i \in [n]\).

The following lemma is helpful for when we need to let the communication protocol send a bit.

**Lemma 6 (Thickness Lemma).** If \(n \geq 2\) and \(A \subseteq [m]^n\) is such that \(\text{AvgDeg}_i(A) \geq d\) for all \(i \in [n]\), then there exists an \(A' \subseteq A\) such that

1. \(\text{MinDeg}_i(A') \geq d/2n\) for all \(i \in [n]\),
2. \(\alpha(A') \leq \alpha(A) + 1\).

The following lemma is helpful for when we need to have the decision tree query a bit.

**Lemma 7 (Projection Lemma).** Suppose \(n \geq 2\), \(A \subseteq [m]^n\) is thick, and \(B \subseteq (\{0, 1\}^m)^n\) is such that \(\beta(B) \leq m^{2/20}\). Then for every \(i \in [n]\) and every \(b \in \{0, 1\}\) there exists a \(b\)-monochromatic rectangle \(U \times V \subseteq [m] \times \{0, 1\}^m\) in \(g\) such that

1. \(A^{i,U}_{[n] \setminus \{i\}}\) is thick,
2. \(\alpha(A^{i,U}_{[n] \setminus \{i\}}) \leq \alpha(A) - \log m + \log \text{AvgDeg}_i(A)\),
3. \(\beta(B^{i,V}_{[n] \setminus \{i\}}) \leq \beta(B) + 1\).
Algorithm 1: Simulation algorithm for Theorem 3

Input: $z \in \{0,1\}^N$

Output: $f(z)$

1. initialize $v = \text{root}$, $I = [N]$, $A = [m]^N$, $B = (\{0,1\}^m)^N$
2. while $v$ is not a leaf do
   3. if $\text{AvgDeg}_i(A_I) \geq m^{19/20}$ for all $i \in I$ then
      4. let $v_0, v_1$ be the children of $v$
      5. if Alice sends a bit at $v$ then
         6. let $b \in \{0,1\}$ be such that $\alpha((A \cap X^{v,b})_I) \leq \alpha(A_I) + 1$
         7. let $A' \subseteq (A \cap X^{v,b})_I$ be such that
            8. (1) $A'$ is thick
            9. (2) $\alpha(A') \leq \alpha((A \cap X^{v,b})_I) + 1$
         10. update $A = \{x \in A \cap X^{v,b} : (x)_i \in I \in A'\}$ and $v = v_b$ (so now $A_I = A'$)
      6. else if Bob sends a bit at $v$ then
         7. let $b \in \{0,1\}$ be such that $\beta((B \cap Y^{v,b})_I) \leq \beta(B_I) + 1$
         8. update $B = B \cap Y^{v,b}$ and $v = v_b$
     7. else if $\text{AvgDeg}_i(A_I) < m^{19/20}$ for some $i \in I$ then
        8. query $z_i$
        9. let $U \times V \subseteq [m] \times \{0,1\}^m$ be a $z_i$-monochromatic rectangle of $g$ such that
           10. (1) $A_{I \setminus \{i\}}^U$ is thick
           11. (2) $\alpha(A_{I \setminus \{i\}}^U) \leq \alpha(A_I) - (\log m)/20$
           12. (3) $\beta(B_{I \setminus \{i\}}^V) \leq \beta(B_I) + 1$
        13. update $A = A_{I \setminus \{i\}}^U$, $B = B_{i,V}$, and $I = I \setminus \{i\}$
     14. end
   8. end
15. output the same value that $v$ does

3.3 Description of the simulation algorithm

The Simulation Theorem is witnessed by Algorithm 1, which is a decision tree for $f$ that employs the hypothesized communication protocol for $F$. Algorithm 1 uses the following variables: $v$ is a node in the communication protocol tree, $I \subseteq [N]$ is the set of unqueried coordinates, $A \subseteq [m]^N$ is a set of inputs to Alice, and $B \subseteq (\{0,1\}^m)^N$ is a set of inputs to Bob. We now expost what Algorithm 1 is doing, with reference to the high-level overview in Section 3.1.

On input $z \in \{0,1\}^N$, the node variable $v$ traces a root-to-leaf path (of length at most $k$) in the protocol tree, which is determined which $z_i$ bits to query, and when. The set $A \times B$ is the “cleaned up” subrectangle of $R_v$ (so we maintain $A \subseteq X^v$ and $B \subseteq Y^v$). We maintain the invariant that every $(x,y) \in A \times B$ is consistent with the results of the queries made so far (i.e., $g^N(x,y)$ agrees with $z$ on queried coordinates), or in other words, $A_{\{i\}} \times B_{\{i\}}$ is $z_i$-monochromatic in $g$ for $i \in [N] \setminus I$. Thus we never need to worry about any coordinate that has previously been queried. The interesting structure in the sets $A$ and $B$ is what they look like on the unqueried coordinates, i.e., the projections $A_I$ and $B_I$. Since all $2^{|I|}$ settings of the unqueried bits of $z$ remain possible, we must maintain that all these settings are indeed possible outcomes of $g^{|I|}$ on points
in $A_I \times B_I$. In fact we maintain a stronger property that turns out to entail this, namely that $A_I$ is thick ($\text{MinDeg}_i(A_I) \geq m^{17/20}$ for every $i \in I$) and $B_I$ is “large” (as measured by $\beta(B_I)$). The potential function is $\alpha(A_I)$; i.e., we look at the set of all projections of elements of $A$ to the unqueried coordinates, and we consider how large this set is compared to its domain $[m]$\textsuperscript{I}. Smaller potential corresponds to a larger set.

We caution that the sets $A$ and $B$ in the statements of the Thickness Lemma and Projection Lemma will not be the $A \subseteq [m]^N$ and $B \subseteq \{(0,1)^m\}^N$ maintained by the algorithm, but rather will be subsets of the projected spaces $([m]^N)_I = [m]^n$ and $((\{0,1\}^m)_I = \{(0,1)^m\}^n$, where $n$ is the size of $I$.

Lines 2–23 are the main loop, with each iteration being either a “communication iteration” (if line 3 holds) in which we update $v, A, B$, or a “query iteration” (if line 15 holds) in which we update $I, A, B$. The type of iteration is determined by $\text{MinDeg}_i(A_I)$, which is our measure of how much the values of the unqueried coordinates are unpredictable from each other within $A \times B$.

In a communication iteration, there are two subcases depending on whether it is Alice’s turn (line 5) or Bob’s turn (line 11) to communicate. In either subcase, the bit of communication partitions $R_v$ (and hence $A \times B$) into two parts, and we restrict our attention to the “bigger” part (lines 6 and 12) by having the communication protocol “send” the corresponding bit. Here, “bigger” is actually in terms of the projections $A_I$ and $B_I$. This ensures the potential does not increase too much if Alice sends, and $B_I$ stays large enough if Bob sends. However, if Alice sends, then the restriction to the bigger part may destroy the thickness invariant, and the Thickness Lemma is used (lines 7–9) to repair this.

In a query iteration, we have the decision tree query a bit $z_i$ for which $\text{AvgDeg}_i(A_I)$ is too small (line 16). Then we can use the Projection Lemma (lines 17–20) to restrict $A \times B$ to a subrectangle on which the $i$-th output bit of $g^N$ is fixed to $z_i$ (for either possible value of $z_i \in \{0,1\}$); this exploits the fact that $\text{MinDeg}_i(A_I)$ is large by the thickness invariant. Furthermore, the fact that $\text{AvgDeg}_i(A_I)$ is small allows us to ensure a $\Omega(\log m)$ decrease in potential (i.e., the density of $A_I$ increases). (Although the absolute size of $A_I$ decreases, recall that the measure $\alpha(A_I)$ is relative to the current set $I$; by fixing the $i$-th coordinate, $I$ becomes $I \setminus \{i\}$, and since we fixed a coordinate of small average degree, the density projected to $I \setminus \{i\}$ will increase a lot.)

### 3.4 Analysis of the simulation algorithm

We now formally argue that Algorithm 1 witnesses the Simulation Theorem (assuming the Thickness Lemma and the Projection Lemma). Assuming lines 7–9 and 17–20 always succeed (which we argue below), in each iteration one of the following three cases occurs.

- If lines 3 and 5 hold, then $\alpha(A_I)$ increases by $\leq 2$ and $\beta(B_I)$ stays the same.
- If lines 3 and 11 hold, then $\alpha(A_I)$ stays the same and $\beta(B_I)$ increases by $\leq 1$.
- If line 15 holds, then $\alpha(A_I)$ decreases by $\geq (\log m)/20$ and $\beta(B_I)$ increases by $\leq 1$.

Since there are at most $k$ iterations in which line 3 holds, and since $\alpha(A_I)$ is initially 0 and always nonnegative, it follows that there are at most $40k/\log m$ iterations in which line 15 holds, and hence the decision tree makes at most $40k/\log m$ queries. Moreover, since there are at most $k + 40k/\log m \leq m^{2/20}$ iterations, and $\beta(B_I)$ is initially 0, at all times we have $\beta(B_I) \leq m^{2/20}$.

**Claim 8.** Lines 7–9 and 17–20 always succeed, and the following loop invariants are maintained.

(i) $A_I$ is thick.
(ii) $A \times B \subseteq R_v$. 

9
(iii) \( g(x_i, y_i) = z_i \) for all \((x, y) \in A \times B\) and all \(i \in [N] \setminus I\).

**Proof.** The invariants trivially hold initially. Now assume they hold at the beginning of an iteration.

Suppose lines 3 and 5 hold. For all \(i \in I\), we have \( \text{AvgDeg}_i((A \cap X^{v,b})_I) = \left| (A \cap X^{v,b})_I \right| / \left| (A \cap X^{v,b})_{I \setminus \{i\}} \right| \geq (|A_I|/2) / |A_{I \setminus \{i\}}| = \text{AvgDeg}_i(A_I)/2 \geq m^{19/20}/2\). Thus we may apply the Thickness Lemma with \((A \cap X^{v,b})_I\) (in place of \(A\) in the lemma), \(I\) identified with \([n]\), and \(d := m^{19/20}/2\) (noting that \(d/2n \geq m^{19/20}/4m^{1/20} \geq m^{17/20}\)) to conclude that lines 7–9 succeed, and hence (i) is maintained. Also, (ii) is maintained by line 10. If lines 3 and 11 hold, then (i) is trivially maintained and (ii) is maintained by line 13. Supposing line 3 holds, in either case (iii) is maintained since the new \(A \times B\) is a subset of the old \(A \times B\) and \(I\) is unchanged.

Now suppose line 15 holds. Since (i) holds and \(\beta(B_I) \leq m^{2/20}\),\(^1\) we may apply the Projection Lemma with \(A_I\) and \(B_I\) (in place of \(A\) and \(B\) in the lemma), \(I\) identified with \([n]\), and \(b := z_i\) (noting that \(-\log m + \log \text{AvgDeg}_i(A_I) \leq -(\log m)/20\)) to conclude that lines 17–20 succeed, and hence (i) is maintained. The new \(A \times B\) is a subset of the old \(A \times B\); therefore, (ii) is maintained since \(v\) is unchanged, and (iii) is maintained since \(U \times V\) is \(z_i\)-monochromatic in \(g\).

Let \(v\) be the leaf reached at termination. We claim that there exists an \((x, y) \in R_v\) such that \(g^N(x, y) = z\), and hence the algorithm indeed outputs \(f(z) = F(x, y)\). Imagine that instead of terminating, the algorithm continues by executing lines 16–21 repeatedly, once for each remaining coordinate \(i \in I\) in arbitrary order until only one coordinate remains unqueried—except that we ignore condition (2) (line 19). In this “extended” execution there are a total of \(k + N - 1 \leq m^{2/20}\) iterations, so we have \(\beta(B_I) \leq m^{2/20}\) at all times, and thus as in the proof of Claim 8, the application of the Projection Lemma always succeeds and invariants (i), (ii), (iii) are maintained.

Consider the state (i.e., \(v, I, A, B\)) at the end of this extended execution. Then \(I\) is a singleton, say \(\{1\}\), and \(|A_{\{1\}}| = \text{MinDeg}_{1}(A_{\{1\}}) \geq m^{17/20}\) by (i), and \(|B_{\{1\}}| \geq 2^{-m^{2/20}} \cdot 2^m = 2^m - m^{m/20}\). Hence \(A_{\{1\}} \times B_{\{1\}}\) is not monochromatic in \(g\), since the largest monochromatic rectangle with rows \(A_{\{1\}}\) has at most \(2^{m-|A_{\{1\}}|} < |B_{\{1\}}|\) columns. Pick an \((x_1, y_1) \in A_{\{1\}} \times B_{\{1\}}\) such that \(g(x_1, y_1) = z_1\), and pick an \((x, y) \in A \times B\) with this value of \((x_1, y_1)\). By (ii) we have \((x, y) \in R_v\), and by (iii) we also have \(g(x_i, y_i) = z_i\) for all \(i \in [N] \setminus \{1\}\), and thus \(g^N(x, y) = z\). The correctness is established.

### 3.5 Proof of the Thickness Lemma

The Thickness Lemma is witnessed by Algorithm 2, which constructs a sequence \(A = A^0 \supseteq A^1 \supseteq A^2 \supseteq \cdots\) that converges to the desired set \(A'\).

<table>
<thead>
<tr>
<th>Algorithm 2: Algorithm for Lemma 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 let (A^0 := A)</td>
</tr>
<tr>
<td>2 for (j = 0, 1, 2, \ldots) do</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4 let (i) such that (\text{MinDeg}_i(A^j) &lt; d/2n), and assume (i = 1) for convenience of notation</td>
</tr>
<tr>
<td>5 let ((x_2^<em>, \ldots, x_n^</em>)) be a nonzero-degree right node in (\text{Graph}_1(A^j)) with degree (&lt; d/2n)</td>
</tr>
<tr>
<td>6 let (A^{j+1} := A^j \setminus {(x_1, x_2^<em>, \ldots, x_n^</em>) : x_1 \in [m]})</td>
</tr>
</tbody>
</table>

\(^1\)There is no circular reasoning here; in showing that \(\beta(B_I) \leq m^{2/20}\) we just needed that lines 7–9 and 17–20 succeeded in all iterations before this one.
If the algorithm terminates, then \( A' \) satisfies (1). We just need to argue that it does terminate, moreover with \(|A'| \geq |A|/2 \) (which is equivalent to (2)). In an iteration, it obtains \( \text{Graph}_i(A^j) \) from \( \text{Graph}_i(A^j) \) by removing all edges incident to some right node in \( A^j \). Hence \( |A^j+1| = |A^j| - 1 \), and for every \( i' \neq i \), \( |A^j| = |A^j+1| \). Therefore, the total number of iterations is at most \( \sum_{i=1}^{n} |A^j| = n \cdot |A|/d \). Since \( |A^j+1| > |A^j| - d/2n \) in each iteration, in total at most \((n \cdot |A|/d) \cdot (d/2n) = |A|/2 \) elements of \( A \) can be removed throughout the execution. Thus the algorithm must terminate with \(|A'| \geq |A|/2 \).

3.6 Proof of the Projection Lemma

Assume \( i = n \) for convenience of notation, so \( A_{[n]}^n = A_{[n-1]}^n = \{(x_1, \ldots, x_{n-1}) : (x_1, \ldots, x_n) \in A \text{ for some } x_n \in U \} \) (which is the set of right nodes in \( \text{Graph}_{n+1}(A) \)) and \( B_{[n]}^n = B_{[n-1]}^n = \{(y_1, \ldots, y_{n-1}) : (y_1, \ldots, y_n) \in B \text{ for some } y_n \in V \} \).

We claim that if we take a uniformly random \( A \subseteq [m] \) of size \( m^{17/20} \) and let \( V := \{ w \in \{0,1\}^m : w_j = b \text{ for all } j \in U \} \), then

\[
\begin{align*}
(0) & \quad A_{[n-1]}^n = A_{[n-1]}^n \text{ with probability greater than } 1 - 2^{-m^{3/20}}, \\
(1) & \quad A_{[n-1]}^n \text{ is thick}, \\
(2) & \quad \alpha(A_{[n-1]}^n) \leq \alpha(A) - \log m + \log \text{AvgDeg}_n(A), \\
(3) & \quad \beta(B_{[n]}^n) \leq \beta(B) + 1 \text{ with probability greater than } 2^{-m^{3/20}}.
\end{align*}
\]

The Projection Lemma then follows by a union bound. (We mention that our argument for property (3) is substantially different from and simpler than the corresponding part of the proof in [RM99].)

Property (0). For every nonzero-degree right node \((x_1, \ldots, x_{n-1}) \in A_{[n-1]}^n \) of \( \text{Graph}_{n+1}(A) \), let \( L_{x_1, \ldots, x_{n-1}} := \{ x_n \in [m] : (x_1, \ldots, x_{n-1}, x_n) \in A \} \) denote the set of all left nodes adjacent to it. We have \( |L_{x_1, \ldots, x_{n-1}}| \geq \text{MinDeg}_n(A) \geq m^{17/20} \), and \((x_1, \ldots, x_{n-1}) \in A_{[n-1]}^n \) iff \( U \) intersects \( L_{x_1, \ldots, x_{n-1}} \). Since \( U \) has size \( m^{17/20} \), the probability \( U \) does not intersect \( L_{x_1, \ldots, x_{n-1}} \) is at most

\[
(1 - m^{17/20} / m)^{m^{17/20}} \leq e^{-m^{3/20}}.
\]

Since the number of elements \((x_1, \ldots, x_{n-1}) \in A_{[n-1]}^n \) is at most \( m^{n-1} \leq 2m^{1/20} \), by a union bound the probability that one of them is not in \( A_{[n]}^n \) is at most \( 2m^{1/20} \cdot e^{-m^{3/20}} < 2^{-m^{3/20}} \).

Property (1). For this it suffices to show that \( \text{MinDeg}_j(A_{[n-1]}^n) \geq \text{MinDeg}_j(A) \) for all \( j \in [n-1] \).

Assume \( j = n - 1 \) for convenience of notation. For every nonzero-degree right node \((x_1, \ldots, x_{n-2}) \in \text{Graph}_{n-1}(A_{[n-1]}^n) \), there exists \( x_{n-1} \) such that \((x_1, \ldots, x_{n-2}, x_{n-1}) \in A_{[n-1]}^n \). Thus by the definition of \( A_{[n-1]}^n \) there exists \( x_n \) such that \((x_1, \ldots, x_{n-2}, x_{n-1}, x_n) \in A \). Therefore, by the definition of \( \text{MinDeg}_{n-1}(A) \) applied to the nonzero-degree right node \((x_1, \ldots, x_{n-2}, x_{n}) \) of \( \text{Graph}_{n-1}(A) \), we have that \((x_1, \ldots, x_{n-2}, x_{n}'_{n-1}, x_n) \in A \) holds for at least \( \text{MinDeg}_{n-1}(A) \) different elements \( x_{n}'_{n-1} \). All these elements satisfy \((x_1, \ldots, x_{n-2}, x_{n}'_{n-1}) \in A_{[n-1]}^n \). Hence, the degree of the right node \((x_1, \ldots, x_{n-2}) \) in \( \text{Graph}_{n-1}(A_{[n-1]}^n) \) is at least \( \text{MinDeg}_{n-1}(A) \).

Property (2). We have \(|A_{[n-1]}| = |A| / \text{AvgDeg}_n(A) \) and \(|m|^{n-1} = |m|^{n}/m \), and hence

\[
\alpha(A_{[n-1]}^n) = \log \left( |m|^{n-1} / |A_{[n-1]}^n| \right) = \log \left( |m|^{n} / |A| \right) - \log (m / \text{AvgDeg}_n(A)) = \alpha(A) - \log m + \log \text{AvgDeg}_n(A).
\]

(Thus (2) holds with equality, but we only needed the inequality.)
Property (3). We first state a claim, whose proof we give later.

Claim 9. For every $W \subseteq \{0,1\}^m$ with $\beta(W) \leq m^{11/20}$, we have $\Pr_U[V \cap W \neq \emptyset] \geq 3/4$.

In particular, for every $W \subseteq \{0,1\}^m$ we have $\Pr_U[V \cap W \neq \emptyset] \geq \frac{3}{4} |W|/2^m - 2^{-m^{11/20}}$. For every $(y_1, \ldots, y_{n-1}) \in (\{0,1\}^m)^{n-1}$, let $W_{y_1,\ldots,y_{n-1}} := \{ y_n \in \{0,1\}^m : (y_1, \ldots, y_{n-1}, y_n) \in B \}$. Letting $(y_1, \ldots, y_{n-1})$ be uniformly random in $(\{0,1\}^m)^{n-1}$, we have

$$\begin{align*}
\mathbb{E}_U[|B_{[n-1]}^{n,V}|/2^{m(n-1)}] &= \mathbb{E}_{y_1,\ldots,y_{n-1}} \Pr_U[(y_1, \ldots, y_{n-1}) \in B_{[n-1]}^{n,V}] \\
&= \mathbb{E}_{y_1,\ldots,y_{n-1}} \Pr_U[V \cap W_{y_1,\ldots,y_{n-1}} \neq \emptyset] \\
&\geq \mathbb{E}_{y_1,\ldots,y_{n-1}} \left( \frac{3}{4} |W_{y_1,\ldots,y_{n-1}}|/2^{m-2^{-m^{11/20}}} \right) \\
&= \frac{3}{4} \cdot |B|/2^{mn} - 2^{-m^{11/20}} \\
&\geq \frac{5}{6} \cdot |B|/2^{mn}
\end{align*}$$

where the last line follows since $|B|/2^{mn} = 2^{-\beta(B)} \geq 2^{-m^{2/10}}$. It follows that with probability at least $\frac{1}{8} \cdot |B|/2^{mn} > 2^{-m^{3/20}}$ over $U$, we have $|B_{[n-1]}^{n,V}|/2^{m(n-1)} \geq \frac{1}{2} \cdot |B|/2^{mn}$, which is equivalent to (3). This finishes the proof of the Projection Lemma, except for the proof of Claim 9.

Recall that $b \in \{0,1\}$ is fixed. For $W \subseteq \{0,1\}^m$ and $j \in [m]$, define $W^j := \{ w \in W : w_j = b \}$ and $\text{Bad}(W) := \{ j \in [m] : |W^j| < |W|/4 \}$.

Claim 10. For every $W \subseteq \{0,1\}^m$, $|\text{Bad}(W)| \leq 6\beta(W)$.

Proof of Claim 10. Let $w$ be a random variable uniformly distributed over $W$, and let $H(\cdot)$ denote Shannon entropy. There are at most $6\beta(W)$ coordinates $j$ such that $\Pr[w_j = b] < 1/4$, since otherwise $H(w) \leq \sum_{j=1}^m H(w_j) < 6\beta(W) \cdot H(1/4) + (m - 6\beta(W)) \cdot 1 \leq m - 6\beta(W) \cdot (1 - 0.82) \leq m - \beta(W)$, contradicting the fact that $H(w) = \log |W| = m - \beta(W)$. $\square$

Proof of Claim 9. Suppose we sample $U := \{ j_1, \ldots, j_{m^{7/20}} \}$ by iteratively picking each $j_{i+1} \in [m] \setminus \{ j_1, \ldots, j_i \}$ uniformly at random. We write $V$ as $V_U$, as a reminder that it depends on $U$. For $i \in \{0,1,\ldots,m^{7/20}\}$, define $W_i := \{ w \in W : w_{j_1} = w_{j_2} = \cdots = w_{j_i} = b \}$, and note that $W_0 = W$, $W_{i+1} = W_{i}^{j_{i+1}}$, and $W_{m^{7/20}} = V_U \cap W$. Let $E_{i+1}$ denote the event that $j_{i+1} \not\in \text{Bad}(W_i)$, and note that if $E_{i+1}$ occurs then $\beta(W_{i+1}) \leq \beta(W_i) + 2$. Thus if $E_1 \cap \cdots \cap E_{m^{7/20}}$ occurs then $\beta(V_U \cap W) \leq \beta(W) + 2m^{7/20} < \infty$ and hence $V_U \cap W \neq \emptyset$. Conditioned on any particular outcome of $j_1, \ldots, j_i$ for which $E_1 \cap \cdots \cap E_i$ occurs, by Claim 10 we have $|\text{Bad}(W_i)| \leq 6\beta(W_i) \leq 6(\beta(W) + 2i)$ and thus

$$\Pr[E_{i+1} | j_1, \ldots, j_i] \geq 1 - \frac{|\text{Bad}(W_i)|}{m - i} \geq 1 - \frac{6(\beta(W) + 2i)}{(6/7)m} \geq e^{-14(\beta(W) + 2i)/m}$$

where the last inequality uses the fact that $1 - x \geq e^{-2x}$ if $x \in [0,1/2]$, applied to $x := 6(\beta(W) + 2i)/(6/7)m \leq 7(m^{11/20} + 2m^{7/20})/m \leq 1/2$. We conclude that

$$\Pr[V_U \cap W \neq \emptyset] \geq \Pr[E_1 \cap \cdots \cap E_{m^{7/20}}] \geq \prod_{i=0}^{m^{7/20}-1} \Pr[E_{i+1} | E_1 \cap \cdots \cap E_i] \geq \prod_{i=0}^{m^{7/20}-1} e^{-14(\beta(W) + 2i)/m} = \exp\left( -\sum_{i=0}^{m^{7/20}-1} 14(\beta(W) + 2i)/m \right)$$

12
The upper bound \( \text{UP} \) is essentially tight) in \([\text{AKK16}]\) BPP thereby quantitatively improving the main result from analogues of these results (the BPP than deterministic lower bounds (using different functions). In particular, the query complexity between quantum query complexity and approximate polynomial degree.

Our techniques, \([\text{MS15}]\) that our function from Theorem 5 already disproves the Saks–Wigderson conjecture. Inspired by one-sided randomized query complexities, as well as a fourth power separation between deterministic and zero-sided randomized query complexities and between zero-sided and \(\text{SW86}\) in \([\text{Subsequent developments.}]\) our techniques have been adapted in several subsequent works.

There is not much room for improvement in Theorem 2, since an \(\text{UPcc}(F)\) upper bound is known. A closer inspection of our proof shows that \(\text{Pcc}(F) \geq \Omega(\text{UPcc}(F)^2 / \log^3 \text{UPcc}(F))\). The lower bound can be improved to \(\Omega(\text{UPcc}(F)^2 / \log^2 \text{UPcc}(F))\) by letting \(h: \{0,1\}^{O(\log n)} \to \Sigma\) (as in (3) at the top of Section 2, where \(|\Sigma| = \text{poly}(n)|\) be the decoder of any asymptotically good error-correcting code (such as the Justesen code). For such an \(h\), any adversary strategy has the property that unless at least some small constant fraction of the input bits to \(h\) have been queried, every element of \(\Sigma\) remains a possible output of \(h\). Thus an adversary strategy for \(h\) composes with the adversary strategy for \(f\) (from Section 2.1) to give \(\text{Pdt}(f \circ h^n) \geq \Omega(n \log n)\). The upper bound \(\text{UPdt}(f \circ h^n) \leq O(\sqrt{n} \log n)\) is unchanged. The Simulation Theorem introduces another \(\log n\) factor.

Grolmusz and Tardos \([\text{GT03}]\) (building on \([\text{KNSW94}]\)) generalized the fact that for all \(F, \text{Pcc}(F) \leq O(\text{UPcc}(F)^2)\) by showing that for all \(F\) and \(\ell \geq 1\), \(\text{Pcc}(F)\) is at most \(O(\ell)\) times the square of the \(\ell\)-ambiguous nondeterministic communication complexity of \(F\), which is the logarithm of the least number of rectangles so that each 1-input is covered between 1 and \(\ell\) times, and no 0-input is covered. This can be shown to be tight by applying the Simulation Theorem to the corresponding query complexity separation. The latter is witnessed by taking the OR of \(\ell\) independent copies of the function witnessing Theorem 5 (rather than taking the AND as in Theorem 4): the deterministic query complexity goes up by a factor of \(\ell\), but the \(\ell\)-ambiguous query complexity of the new function is at most the unambiguous query complexity of the original function.

Subsequent developments. Our techniques have been adapted in several subsequent works. In \([\text{SW86}]\), Saks and Wigderson conjectured that the largest separation between deterministic and zero-sided randomized query complexities should be a power roughly 1.326 (witnessed by the recursive AND–OR tree). The paper \([\text{ABB}^{+}16a]\) used variants of our function from Theorem 5 to disprove the Saks–Wigderson conjecture and, in fact, to give optimal (quadratic) separations between deterministic and zero-sided randomized query complexities and between zero-sided and one-sided randomized query complexities, as well as a fourth power separation between deterministic and quantum query complexities, among other separations. The paper \([\text{MS15}]\) independently showed that our function from Theorem 5 already disproves the Saks–Wigderson conjecture. Inspired by our techniques, \([\text{ABK16}]\) exhibited a function witnessing a power 2.5 separation between two-sided randomized and quantum query complexities, as well as a function witnessing a power 4 separation between quantum query complexity and approximate polynomial degree.

The paper \([\text{GJPW17}]\) strengthened Theorem 1 and Theorem 2 to have randomized (BPPcc) rather than deterministic lower bounds (using different functions). In particular, the query complexity analogues of these results (the BPPdt analogues of Theorem 4 and Theorem 5) were also shown, thereby quantitatively improving the main result from \([\text{KRS15}]\) (which was the first superlinear BPPdt vs. \(2\text{UPdt}\) separation).

The exponent in our Theorem 1 and Theorem 4 has been improved from 1.5 to \(2 - o(1)\) (which is essentially tight) in \([\text{AKK16}]\). Furthermore, several other query complexity separations were

\[
\begin{align*}
= \exp\left(-\frac{14}{m}\left(\beta(W)m^{7/20} + (m^{7/20} - 1)m^{7/20}\right)\right) \\
\geq \exp\left(-14(m^{-2/20} + m^{-6/20})\right) \\
\geq 3/4. \quad \Box
\end{align*}
\]

4 Conclusion

Further observations. There is not much room for improvement in Theorem 2, since an \(O(\text{UPcc}(F)^2)\) upper bound is known. A closer inspection of our proof shows that \(\text{Pcc}(F) \geq \Omega(\text{UPcc}(F)^2 / \log^3 \text{UPcc}(F))\). The lower bound can be improved to \(\Omega(\text{UPcc}(F)^2 / \log^2 \text{UPcc}(F))\) by letting \(h: \{0,1\}^{O(\log n)} \to \Sigma\) (as in (3) at the top of Section 2, where \(|\Sigma| = \text{poly}(n)|\) be the decoder of any asymptotically good error-correcting code (such as the Justesen code). For such an \(h\), any adversary strategy has the property that unless at least some small constant fraction of the input bits to \(h\) have been queried, every element of \(\Sigma\) remains a possible output of \(h\). Thus an adversary strategy for \(h\) composes with the adversary strategy for \(f\) (from Section 2.1) to give \(\text{Pdt}(f \circ h^n) \geq \Omega(n \log n)\). The upper bound \(\text{UPdt}(f \circ h^n) \leq O(\sqrt{n} \log n)\) is unchanged. The Simulation Theorem introduces another \(\log n\) factor.

Grolmusz and Tardos \([\text{GT03}]\) (building on \([\text{KNSW94}]\)) generalized the fact that for all \(F, \text{Pcc}(F) \leq O(\text{UPcc}(F)^2)\) by showing that for all \(F\) and \(\ell \geq 1\), \(\text{Pcc}(F)\) is at most \(O(\ell)\) times the square of the \(\ell\)-ambiguous nondeterministic communication complexity of \(F\), which is the logarithm of the least number of rectangles so that each 1-input is covered between 1 and \(\ell\) times, and no 0-input is covered. This can be shown to be tight by applying the Simulation Theorem to the corresponding query complexity separation. The latter is witnessed by taking the OR of \(\ell\) independent copies of the function witnessing Theorem 5 (rather than taking the AND as in Theorem 4): the deterministic query complexity goes up by a factor of \(\ell\), but the \(\ell\)-ambiguous query complexity of the new function is at most the unambiguous query complexity of the original function.

Subsequent developments. Our techniques have been adapted in several subsequent works. In \([\text{SW86}]\), Saks and Wigderson conjectured that the largest separation between deterministic and zero-sided randomized query complexities should be a power roughly 1.326 (witnessed by the recursive AND–OR tree). The paper \([\text{ABB}^{+}16a]\) used variants of our function from Theorem 5 to disprove the Saks–Wigderson conjecture and, in fact, to give optimal (quadratic) separations between deterministic and zero-sided randomized query complexities and between zero-sided and one-sided randomized query complexities, as well as a fourth power separation between deterministic and quantum query complexities, among other separations. The paper \([\text{MS15}]\) independently showed that our function from Theorem 5 already disproves the Saks–Wigderson conjecture. Inspired by our techniques, \([\text{ABK16}]\) exhibited a function witnessing a power 2.5 separation between two-sided randomized and quantum query complexities, as well as a function witnessing a power 4 separation between quantum query complexity and approximate polynomial degree.

The paper \([\text{GJPW17}]\) strengthened Theorem 1 and Theorem 2 to have randomized (BPPcc) rather than deterministic lower bounds (using different functions). In particular, the query complexity analogues of these results (the BPPdt analogues of Theorem 4 and Theorem 5) were also shown, thereby quantitatively improving the main result from \([\text{KRS15}]\) (which was the first superlinear BPPdt vs. \(2\text{UPdt}\) separation).

The exponent in our Theorem 1 and Theorem 4 has been improved from 1.5 to \(2 - o(1)\) (which is essentially tight) in \([\text{AKK16}]\). Furthermore, several other query complexity separations were

\[
\begin{align*}
= \exp\left(-\frac{14}{m}\left(\beta(W)m^{7/20} + (m^{7/20} - 1)m^{7/20}\right)\right) \\
\geq \exp\left(-14(m^{-2/20} + m^{-6/20})\right) \\
\geq 3/4. \quad \Box
\end{align*}
\]
proven in [AKK16]: a power 2 separation between $\text{BPP}^{\text{dt}}$ and $2\text{UP}^{\text{dt}}$ (which has since been lifted to communication complexity [ABB+16b, GPW17]), a power 1.5 separation between $\text{BQP}^{\text{dt}}$ and $2\text{UP}^{\text{dt}}$, and a power 2 separation between $\text{BQP}^{\text{dt}}$ and $\text{UP}^{\text{dt}}$.

The papers [RW16, BR17] adapted and applied the Simulation Theorem to prove lower bounds on the communication complexity of finding Nash equilibria.

The Simulation Theorem itself has been improved in various ways: [dNV16] proved a version where rounds of communication in the protocol correspond directly to rounds of adaptivity in the decision tree (and the paper gave applications to proof complexity and circuit complexity), and [CKLM17, WYY17] improved the gadget to be the inner-product function on $O(\log n)$ bits (as was used in [GLM+16]).

Finally, our techniques have inspired the development of simulation theorems for other models: bounded-error randomized (BPP-type) protocols/decision trees [GPW17], as well as $\text{P}^{\text{NP}}$-type protocols/decision trees [GKPW17].

Acknowledgements

We thank Raghu Meka and Ran Raz for very helpful discussions.

References


