Electronic Colloquium on Computational Complexity, Revision 3 of Report No. 146 (2017)

On Derandomized Composition of Boolean Functions

Or Meir*

July 24, 2018

Abstract

The composition of two Boolean functions $f : \{0,1\}^m \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}$ is the function $f \diamond g$ that takes as inputs m strings $x_1, \ldots, x_m \in \{0,1\}^n$ and computes

$$(f \diamond g)(x_1, \ldots, x_m) = f\left(g(x_1), \ldots, g(x_m)\right).$$

This operation has been used several times in the past for amplifying different hardness measures of f and g. This comes at a cost: the function $f \diamond g$ has input length $m \cdot n$ rather than m or n, which is a bottleneck for some applications.

In this paper, we propose to decrease this cost by "derandomizing" the composition: instead of feeding into $f \diamond g$ independent inputs $x_1, \ldots, x_m$, we generate $x_1, \ldots, x_m$ using a shorter seed. We show that this idea can be realized in the particular setting of the composition of functions and universal relations [KRW95, GMWW17]. To this end, we provide two different techniques for achieving such a derandomization: a technique based on averaging samplers, and a technique based on Reed-Solomon codes.

1 Introduction

Given two Boolean functions $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$, their *(block-)composition* is the function $f \diamond g : (\{0,1\}^n)^m \to \{0,1\}$ that is defined by

$$(f \diamond g)(x_1, \ldots, x_m) = f(g(x_1), \ldots, g(x_m)).$$

This operation has been a useful tool in proving lower bounds on a variety of complexity measures. This is usually done by showing that $f \diamond g$ is harder than f and g in some sense, and then using the composition to construct hard functions with some desired properties. Some important examples of this idea are the following:

- **Depth complexity:** The *depth complexity* of a Boolean function f, denoted $\mathsf{D}(f)$, is the depth of the shallowest circuit that computes it (with fan-in 2). Karchmer, Raz, and Wigderson [KRW95] conjectured that

  $$\mathsf{D}(f \diamond g) \approx \mathsf{D}(f) + \mathsf{D}(g).$$

  Following [GMWW17], we refer to this conjecture as the "KRW conjecture". [KRW95] showed that their conjecture, if true, implies super-logarithmic lower bounds on the depth complexity of an explicit function, thus resolving an outstanding open problem of complexity theory. The latter explicit function is constructed by a repeated application of the composition operation. They also showed that this approach can be used to prove such lower bounds in the settings of monotone circuits.

*Department of Computer Science, University of Haifa, Haifa 3498838, Israel. ormeir@cs.haifa.ac.il. Partially supported by the Israel Science Foundation (grant No. 1445/16).

ISSN 1433-8092

- **Lifting theorems:** Raz and McKenzie [RM97] showed[1] that the composition operation can be used to "lift" lower bounds from query complexity to communication complexity. Suppose that we have a query-complexity lower-bound for some function f, and we wish to "lift" it to communication complexity. [RM97] showed that we can do it by composing f with some specific function g. Specifically, the composed function $f \diamond g$ has communication complexity which is roughly the query complexity of f.
  [RM97] used this theorem to prove that the monotone **NC**-hierarchy does not collapse. Recently, such lifting theorems were proved for other models and other choices of g, and were used to derive a host of new important complexity separations. See [SZ09, She11, GPW15, Göö15, LRS15, GLM$^+$16, CKLM17, GKPW17, KMR17, WYY17] for some examples.

- **Sensitivity measures:** The *sensitivity*, *block complexity* and *certificate complexity* of a Boolean function f are important combinatorial measures, which measure how sensitive f is to changes in its input. Nisan and Wigderson [NW95] used the composition operation to prove a separation between the sensitivity of a function and its polynomial degree. Gilmer, Saks, and Srinivasan [GSS13] used the composition operation to provide an optimal separation between these block complexity and certificate complexity. Kulkarni and Tal [KT16] used the composition operation to separate variant of block complexity, namely, *fractional block complexity*, from the degree that is required to approximate f as a polynomial. A key tool in those results is the fact that the fractional block complexity of $f \diamond g$ is the product of the fractional block complexities of f and g. On a related note, Tal [Tal13] used the composition operation to resolve a question of Kalai on low-degree functions [MOS04].

The cost of the composition operation is that it increases the length of the input: the input length of $f \diamond g$ is $m \cdot n$ rather than m or n. This cost is a serious bottleneck for the applications of this operation. For example, in the application of [KRW95], the composition is used multiple times, and thus the input length grows exponentially with the number of compositions. This leads to weaker lower bounds than those that could have been achieved if the input of $f \diamond g$ was shorter, say, $m+n$. Similarly, in the applications of composition to lifting theorems, the cost of composition sometimes weakens the lower bounds that are obtained via lifting.

In this work, we propose an approach for decreasing this cost by defining a "derandomized" composition. A derandomized composition is a function that takes as input a string z of length ℓ where $\ell \ll m \cdot n$, uses z to generate strings $x_1, \ldots, x_m \in \{0,1\}^n$, and then outputs $(f \diamond g)(x_1, \ldots, x_m)$. We refer to this approach as "derandomized composition" since one can think of this construction as if z is the seed of a "pseudorandom generator" that generates strings $x_1, \ldots, x_m$ that are indistinguishable from independent strings from the point of view of $f \diamond g$. This approach follows[2] the derandomizations of Yao's XOR lemma by Impagliazzo [Imp95], and Impagliazzo and Wigderson [IW97].

If one could prove that this derandomized composition is roughly as hard as $f \diamond g$, then one would be able to use it as a substitute for $f \diamond g$ in the above applications while reducing the cost from $m \cdot n$ to ℓ. Our main results show that this approach can indeed by realized in the setting of the composition of functions and universal relations, which is a variant of the KRW conjecture discussed above. We now describe this setting as well as the relevant background.

[1] Actually, [RM97] proved it for a specific type of functions. Later, Goos, Pitassi and Watson [GPW15] observed that the proof of [RM97] works for every function.

[2] We note that recently, Razborov [Raz16] used similar ideas to prove very strong trade-offs for the resolution proof-system.

1.1 Background

1.1.1 Karchmer-Wigderson relations

Our main tool in studying the KRW conjecture is a framework developed by Karchmer and Wigderson [KW90]. They observed an interesting connection between depth complexity and communication complexity: for every Boolean function f, there exists a corresponding communication problem KW_f, such that the depth complexity of f is equal to the communication complexity[3] of KW_f. The communication problem KW_f is often called the *Karchmer-Wigderson relation* of f, and we will refer to it as a *KW relation* for short.

The communication problem KW_f is defined as follows: Alice gets an input $x \in f^{-1}(0)$, and Bob gets an input $y \in f^{-1}(1)$. Clearly, it holds that $x \neq y$. The goal of Alice and Bob is to find a coordinate i such that $x_i \neq y_i$. Note that there may be more than one possible choice for i, which means that KW_f is a relation rather than a function.

This connection between functions and KW relations allows us to study the depth complexity of functions using techniques from communication complexity. In the past, this approach has proved very fruitful in the setting of *monotone* circuits [KW90, GS91, RW92, KRW95], and in particular [KRW95] used it to separate the monotone versions of $\mathbf{NC}^1$ and $\mathbf{NC}^2$.

1.1.2 KW relations and the KRW conjecture

In order to prove the KRW conjecture, one could study the KW relation that corresponds to the composition $f \diamond g$. Let us describe how the KW relation $KW_{f \diamond g}$ looks like. Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$. Recall that the input to the composition consists of m strings in $\{0,1\}^n$. In order to streamline the presentation, it is more convenient to represent these m strings as an $m \times n$ binary matrix. For every $m \times n$ matrix X, let us denote by $g(X)$ the string in $\{0,1\}^m$ obtained by applying g to each row of X, so $f \diamond g(X) = f(g(X))$. In the KW relation $KW_{f \diamond g}$, Alice and Bob get as inputs $m \times n$ matrices X, Y, respectively, such that $g(X) \in f^{-1}(0)$ and $g(Y) \in f^{-1}(1)$, and their goal is to find an entry (i,j) such that $X_{i,j} \neq Y_{i,j}$.

Let us denote the (deterministic) communication complexity of a problem R by $\mathsf{C}(R)$. Clearly, it holds that

$$\mathsf{C}(KW_{f \diamond g}) \leq \mathsf{C}(KW_f) + \mathsf{C}(KW_g). \tag{1}$$

This upper bound is achieved by the following protocol: for every $i \in [m]$, let X_i denote the i-th row of X, and same for Y. Alice and Bob first use the optimal protocol of f on inputs $g(X)$ and $g(Y)$, and thus find an index $i \in [m]$ such that $g(X_i) \neq g(Y_i)$. Then, they use the optimal protocol of g on inputs X_i and Y_i to find a coordinate j on which the i-th rows differ, thus obtaining an entry (i,j) on which X and Y differ. The KRW conjecture says that the above protocol is essentially optimal.

1.1.3 The universal relation and its composition

Since proving the KRW conjecture seems difficult, [KRW95] suggested studying a simpler problem as a starting point. To describe this simpler problem, we first need to define a communication problem called the *universal relation*, and its composition with itself. The *universal relation* U_n is a communication problem in which Alice and Bob get as inputs $x, y \in \{0,1\}^n$ with the sole guarantee that $x \neq y$, and their goal is to find a coordinate i such that $x_i \neq y_i$. The universal relation U_n is universal in the sense that every KW relation reduces to it, and it is know that $n \leq \mathsf{C}(\mathrm{U}_n) \leq n + 2$ (see [KRW95] for the lower bound and [TZ97] for the upper bonds).

[3] In this paper, we always refer to *deterministic* communication complexity, unless stated explicitly otherwise.

The composition of two universal relations U_m and U_n, denoted $\mathrm{U}_m \diamond \mathrm{U}_n$, is defined as follows. Alice gets as input an $m \times n$ matrix X and a string $a \in \{0,1\}^m$, and Bob gets as input an $m \times n$ matrix Y and a string $b \in \{0,1\}^m$. Their inputs satisfy the following conditions:

1. $a \neq b$.
2. for every $i \in [m]$ such that $a_i \neq b_i$, it holds that $X_i \neq Y_i$.

Their goal, as before, is to find an entry on which X and Y differ. The vectors a and b are analogues of the vectors $g(X)$ and $g(Y)$ in the KW relation $KW_{f \diamond g}$.

To see why $\mathrm{U}_m \diamond \mathrm{U}_n$ is a good way to abstract the KRW conjecture, observe that $\mathrm{U}_m \diamond \mathrm{U}_n$ is a universal version of composition problems $KW_{f \diamond g}$, in the sense that every composition problem $KW_{f \diamond g}$ reduces to $\mathrm{U}_m \diamond \mathrm{U}_n$. Moreover, the protocol described above for $KW_{f \diamond g}$ also works for $\mathrm{U}_m \diamond \mathrm{U}_n$: Alice and Bob first apply the optimal protocol for U_m to a and b to find i, and then apply the optimal protocol for U_n to X_i and Y_i. In particular, it holds that $\mathsf{C}(\mathrm{U}_m \diamond \mathrm{U}_n) \leq \mathsf{C}(\mathrm{U}_m) + \mathsf{C}(\mathrm{U}_n)$. Thus, a natural variant of the KRW conjecture for this composition would be that the latter protocol is optimal for $\mathrm{U}_m \diamond \mathrm{U}_n$. Following this reasoning, [KRW95] suggested to prove that

$$\mathsf{C}(\mathrm{U}_m \diamond \mathrm{U}_n) \approx \mathsf{C}(\mathrm{U}_m) + \mathsf{C}(\mathrm{U}_n) \geq m + n \tag{2}$$

as a first step toward proving the KRW conjecture. This challenge was met[4] by [EIRS01] up to a small additive loss, and an alternative proof was given later[5] in [HW93].

1.1.4 The composition of a function with the universal relation

So far we discussed the composition $KW_{f \diamond g}$, for which no lower bound is known, and the composition $\mathrm{U}_m \diamond \mathrm{U}_n$, for which a lower bound is known. In order to bridge the gap between the two compositions, Gavinsky et. al. [GMWW17] defined a composition $KW_{f \diamond \mathrm{U}_n}$ between a function $f : \{0,1\}^m \to \{0,1\}$ and a universal relation U_n.

The communication problem $KW_{f \diamond \mathrm{U}_n}$ is defined as follows: Alice gets as input an $m \times n$ matrix X and a string $a \in f^{-1}(0)$, and Bob gets as input an $m \times n$ matrix Y and a string $b \in f^{-1}(1)$. Their inputs are guaranteed to satisfy Condition 2 of $\mathrm{U}_m \diamond \mathrm{U}_n$, i.e., for every $i \in [m]$ such that $a_i \neq b_i$, it holds that $X_i \neq Y_i$. Clearly, their inputs also satisfy $a \neq b$, as in Condition 1 of $\mathrm{U}_m \diamond \mathrm{U}_n$. The goal of Alice and Bob, as usual, is to find an entry on which X and Y differ.

Note that $KW_{f \diamond \mathrm{U}_n}$ is universal, in the sense that for any $g : \{0,1\}^n \to \{0,1\}$, the communication problem $KW_{f \diamond g}$ reduces to $KW_{f \diamond \mathrm{U}_n}$. Also, as in the previous variants of the KRW conjecture, it holds that $\mathsf{C}(KW_{f \diamond \mathrm{U}_n}) \leq \mathsf{C}(KW_f) + \mathsf{C}(\mathrm{U}_n)$. Therefore, a natural analogue of the KRW conjecture for $KW_{f \diamond \mathrm{U}_n}$ would be

$$\mathsf{C}(KW_{f \diamond \mathrm{U}_n}) \approx \mathsf{C}(KW_f) + \mathsf{C}(\mathrm{U}_n) \geq \mathsf{C}(KW_f) + n. \tag{3}$$

In order to state the result of [GMWW17], let us denote by $\mathsf{L}(f)$ the *formula complexity* of f (see Section 2.1). The quantity $\log \mathsf{L}(f)$ is closely related to the depth complexity $\mathsf{D}(f)$, and hence to the communication complexity $\mathsf{C}(KW_f)$. Now, [GMWW17] proved the following result.

Theorem 1.1 ([GMWW17])**.** *Let $m, n \in \mathbb{N}$, and let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant function. Then,*

$$\mathsf{C}(KW_{f \diamond \mathrm{U}_n}) \geq \log \mathsf{L}(f) + n - O(1 + \frac{m}{n}) \cdot \log m.$$

[4] In fact, they only consider the case where $m = n$, but their argument also works in the case where $m \neq n$.

[5] We note that we cite here the journal versions of those works, and therefore [KRW95] and [EIRS01] look as if they appeared after [HW93]. However, the conference versions of [KRW95] and [EIRS01] appeared in 1991.

Again, since $\log \mathsf{L}(f)$ is closely related to $\mathsf{C}(KW_f)$, this result is closely related to the conjecture of Equation 3. In a recent work, Koroth and the author [KM18] improved the above bound and obtained the following result.

Theorem 1.2 ([KM18])**.** *Let $m, n \in \mathbb{N}$ be such that $m < 2^{\frac{n}{6}}$., and let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant function. Then,*

$$\mathsf{C}(KW_{f \diamond \mathrm{U}_n}) \geq \log \mathsf{L}(f) + n - O\left(\log^*(m)\right).$$

1.2 Our results

In this work we present two different ways to "derandomize" the composition $KW_{f \diamond \mathrm{U}_n}$. That is, we define two variants of the composition $KW_{f \diamond \mathrm{U}_n}$ that have significantly shorter input length, and prove that their complexity is close to that of $KW_{f \diamond \mathrm{U}_n}$.

In order to motivate our definitions of these variants, let us first define the notion of derandomizing the standard composition $f \diamond g$. We define a *generator* to be a function $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$, where $\ell < m \cdot n$ and $\{0,1\}^{m \times n}$ denotes the set of $m \times n$ binary matrices. Intuitively, we think of ϕ as a "pseudorandom generator" that uses a short seed in order to generate inputs for $f \diamond g$. The "derandomized composition" that corresponds to ϕ will simply be the function $(f \diamond g) \circ \phi$ (where $\circ$ denotes the standard function composition). We think of the generator ϕ as "good" (or "pseudorandom") if the function $(f \diamond g) \circ \phi$ is roughly as hard as the function $f \diamond g$.

The corresponding KW relation $KW_{(f \diamond g) \circ \phi}$ is the following communication problem: Alice and Bob get strings $x, y \in \{0,1\}^\ell$, and they use them to generate matrices $X = \phi(x)$, $Y = \phi(y)$ and strings $a = g(X)$, $b = g(Y)$. The parties are guaranteed that $a \in f^{-1}(0)$ and $b \in f^{-1}(1)$, and their goal is find a coordinate $k \in [\ell]$ such that $x_k \neq y_k$. Observe that, as before, for every $i \in [m]$ such that $a_i \neq b_i$, it holds that $X_i \neq Y_i$.

We now define the derandomized composition of f with the universal relation in an analogous way. Let $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ be a generator. The *derandomized composition* $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$ is defined as follows: Alice and Bob receive as inputs strings $x, y \in \{0,1\}^\ell$ and strings $a \in f^{-1}(0)$, $b \in f^{-1}(1)$. We denote $X = \phi(x)$, $Y = \phi(y)$. Then, Alice and Bob are guaranteed that for every $i \in [n]$ such that $a_i \neq b_i$, it holds that $X_i \neq Y_i$, and their goal is to find a coordinate $k \in [\ell]$ such that $x_k \neq y_k$. It is not hard to see that for any generator ϕ, it holds that

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \leq \mathsf{C}(KW_f) + \mathsf{C}(\mathrm{U}_n),$$

as in the case of the non-derandomized composition. Our goal is to find generators ϕ for which this upper bound is close to be tight.

In this work, we present two specific generators $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$, and show that the corresponding problems $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$ are almost as hard as the non-derandomized problem $KW_{f \diamond \mathrm{U}_n}$. Our first generator is based on sampling: the generator ϕ maps a string x to a matrix X whose rows are projections of x to certain subsets of coordinates, where these subsets sample the set $[\ell]$ well (see Section 2.6 for the definition of sampling). For this generator, we prove the following result.

Theorem 1.3. *There exists a universal constant $q \in \mathbb{N}$ such that the following holds. Let $\varepsilon > 0$ be an arbitrary constant, and let $m, n \in \mathbb{N}$ be such that $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$. Then, for every $f : \{0,1\}^m \to \{0,1\}$ there exists a generator $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ such that $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$ and such that*

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \left(1 - \frac{q}{\varepsilon^4 \cdot n}\right) \cdot \mathsf{C}(KW_f) + (1 - 3\varepsilon) \cdot n - 2 \log m - 1.$$

Remark 1.4. The above theorem states that the generator ϕ depends on the function f. However, our construction of the generator depends only on the complexity $\mathsf{C}(KW_f)$ (since we wish to set $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$). Given the choice of ℓ, the generator ϕ is independent of f.

Remark 1.5. In the above theorem, it may be surprising that the seed length ℓ may be as small as n. However, note that in this case the lower bound holds vacuously, since in this case it must hold that $n \geq \mathsf{C}(KW_f)/\varepsilon^2$ and therefore the loss of $O(\varepsilon \cdot n)$ exceeds $\mathsf{C}(KW_f)$.

Remark 1.6. In the above theorem, the restriction $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$ comes from the limitations of the known explicit samplers. If we used non-explicit samplers, we could get a (non-explicit) generator with a milder restriction of $n \geq q \cdot \frac{\log\log m}{\varepsilon^2}$.

Remark 1.7. Note that in the above theorem the lower bound depends on $\mathsf{C}(KW_f)$ whereas in the previous work of [GMWW17] it depends on $\log \mathsf{L}(f)$. In a sense, this is an improvement, since the KRW conjecture refers to $\mathsf{C}(KW_f)$ rather than $\log \mathsf{L}(f)$. In fact, our techniques could also be used to prove a lower bound on the non-derandomized composition $KW_{f \diamond \mathrm{U}_n}$ that depends on $\mathsf{C}(KW_f)$ rather than $\log \mathsf{L}(f)$.

Our second generator is based on Reed-Solomon codes: the generators ϕ views the string x as a vector over the finite field $\mathbb{F}$ of size 2^n, and the matrix X as a vector in $\mathbb{F}^m$. Taking this view, the matrix X is the encoding of the the string x via a Reed-Solomon code. In other words, the string x is viewed as representing the coefficients of a univariate polynomial p of degree $\ell/n - 1$ over $\mathbb{F}$, and every row of X consists of the evaluation of p at some fixed points in $\mathbb{F}$ (see Section 2 for additional information on Reed-Solomon codes). For this generator, we have the following result.

Theorem 1.8. *Let $m, n \in \mathbb{N}$ be such that $m < 2^n$. Then, there exists a generator $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ such that $\ell \leq 5 \cdot m + n$, and such that for every $f : \{0,1\}^m \to \{0,1\}^n$ it holds that*

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \log \mathsf{L}(f) + n - 4 \cdot (1 + \frac{m}{n}) \cdot \log m.$$

Comparison between the two generators. Roughly speaking, Theorem 1.3 is stronger when m is large compared to n, whereas Theorem 1.8 is stronger when m is close to n or smaller.

1.3 Our techniques

In this section we provide a high-level description of the analysis of our generators (i.e., proofs of Theorems 1.3 and 1.8). In both proofs, we use a slightly different definition of $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$: In the modified definition, we do not promise the parties that the inputs satisfy that $X_i \neq Y_i$ whenever $a_i \neq b_i$. Instead, we allow the inputs to violate this promise, in which case the parties are allowed to reject the inputs (but not to output a wrong answer). It is not hard to show that this version of the problem is not much harder than the original definition we presented (see Section 2.4 for details). In both proofs, the lower bound is based on choosing the inputs such that the parties always reject. This means that the task of the parties becomes the task of certifying that there exists an index $i \in [m]$ such that $a_i \neq b_i$ but $X_i = Y_i$. Our goal in both proofs is to lower bound the amount of communication that is needed to accomplish this task.

1.3.1 Sampling-based generator

We begin by discussing the proof of Theorem 1.3. In order to construct the generator of this theorem, we fix a sequence of sets $\mathcal{S}_1, \ldots, \mathcal{S}_m \subseteq [\ell]$ of size n that constitute a good averaging

sampler of $[\ell]$ (i.e., every subset of $[\ell]$ intersects most of the sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ with roughly the right proportion). The generator ϕ is now defined as follows: given a string $x \in \{0,1\}^\ell$, the generator outputs the $m \times n$ matrix X whose i-th row is $x|_{\mathcal{S}_i}$.

We would like to prove a lower bound of roughly $\mathsf{C}(KW_f)+(1-\varepsilon)\cdot n$ on $KW_{(f\diamond \mathrm{U}_n)\circ\phi}$. Our proof uses an adversary argument: we describe an adversary that takes a protocol that is "too efficient", and constructs a transcript in which the protocol errs. Specifically, the adversary simulates the protocol by choosing messages for the parties, until the protocol ends and outputs some output. Then, the adversary "wins" if it can choose inputs that are consistent with the messages that were sent, but on which the output of the protocol is wrong.

In particular, the adversary will construct a transcript that rejects the inputs. In order to show that this transcript errs, the adversary will construct inputs (x, a) and (y, b) that are consistent with the transcript and should not be rejected. That is, those inputs will satisfy that for every $i \in [m]$, if $a_i \neq b_i$ then $X_i \neq Y_i$. To this end, observe that if, for some $i \in [m]$, the parties transmitted less than n bits of information "about" the i-th row by the end of the protocol, then the adversary can choose inputs for which $X_i \neq Y_i$. Intuitively, the reason is that the communication complexity of equality is n, and therefore the parties have to transmit n bits of information about the i-th row in order to certify that $X_i = Y_i$. The goal of the adversary, therefore, is to make sure that when the protocol ends, for every $i \in [m]$ one of the following holds:

- either $a_i = b_i$,
- or the parties have transmitted less than n bits of information "about" the i-th row.

We now describe the adversary in more detail. Suppose that the adversary is given a protocol that is too efficient, i.e., it transmits significantly less than $\mathsf{C}(KW_f) + (1-\varepsilon)\cdot n$ bits. The adversary partitions the protocol into two stages, where the first stage consists of somewhat less than $\mathsf{C}(KW_f)$ bits and the second stage consists of less than $(1-\varepsilon)\cdot n$ bits. The adversary starts by simulating the first stage of the protocol. After the first stage ends, we partition the rows of the matrices X and Y into two types:

- "Revealed rows": Rows about which at least $\varepsilon \cdot n$ bits of information have been transmitted so far.
- "Unrevealed rows": Rows about which less than $\varepsilon \cdot n$ bits of information have been transmitted so far.

Now, on the unrevealed rows the adversary has already won: the reason is that in the remaining part of the protocol, the players are only going to transmit less than $(1-\varepsilon)\cdot n$ bits. Hence, by the end of the protocol, less than n bits of information are going to have been transmitted on each of the unrevealed rows, which is what the adversary wants.

It remains to deal with the revealed rows. This is done as follows: When the first stage ends, the adversary "discards" some of the inputs that are still legal at this point. The remaining inputs have the property that for every revealed row i it holds that $a_i = b_i$, so the adversary wins on those rows as well. Then, the adversary runs the second stage only on those remaining inputs. When the protocol ends, the adversary wins on every row, since for the revealed rows it holds that $a_i = b_i$, and on each of the unrevealed rows the parties have transmitted less than n bits of information.

We still need to show that it is possible to discard inputs in this manner without discarding too many inputs. The argument here consists of two points:

- The adversary can discard inputs so as to force $a_i = b_i$ on a small number of the rows.

- The number of revealed rows is small.

Together, these two items imply that adversary can force the equality $a_i = b_i$ on all the revealed rows. We conclude the overview by explaining why each of the foregoing items is correct:

- The reason that the adversary can force $a_i = b_i$ on a small number of rows is that during the first stage, the parties transmitted somewhat less than $\mathsf{C}(KW_f)$ bits. Hence, they have not finished solving KW_f, and they do not know any index i on which $a_i \neq b_i$. Therefore, there are many possible inputs that satisfy $a_i = b_i$ on any small set of rows.

- To see why the number of revealed rows is small, recall that the parties transmitted less than $\mathsf{C}(KW_f)$ bits in the first stage, and that $\mathsf{C}(KW_f) \leq \varepsilon \cdot \ell$ (by the choice of ℓ in the theorem). This means that the parties transmitted less than $\varepsilon \cdot \ell$ bits of information about their inputs, and therefore they transmitted less than $\varepsilon \cdot n$ bits about the *average* row. Now, since $\mathcal{S}_1, \ldots, \mathcal{S}_m$ constitute a good sampler, this means that the players have transmitted about $\varepsilon \cdot n$ bits of information about almost all the rows.

This concludes the proof. The strategy of this proof was originally developed by Edmonds et. al. [EIRS01] in order to prove a lower bound on $\mathrm{U}_m \diamond \mathrm{U}_n$ (in which case $\ell = m \cdot n$ and the sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are disjoint), and was extended recently by Dinur and Meir [DM16] to the composition of certain functions. Our contribution is observing that this argument can be extended to the case where the sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are not disjoint but form a good sampler. This requires, among other things, using a somewhat stronger information-theoretic tools compared to [EIRS01] (see Lemma 2.22).

1.3.2 Reed-Solomon-based generator

We turn to discuss the proof of Theorem 1.8. The generator ϕ of this theorem takes as an input a string $x \in \{0,1\}^\ell$ and interprets it as a vector in $\mathbb{F}^{\ell/n}$ where $\mathbb{F}$ is the finite field of size 2^n. The generator ϕ maps x to the matrix $X \in \{0,1\}^{m \times n}$ that corresponds to the encoding of x via the Reed-Solomon code when viewed as a vector in $\mathbb{F}^m$.

The crux of the proof is the following "agreement property" of Reed-Solomon codes, which we prove in Section 4.1: Consider the communication problem in which each of Alice and Bob gets codeword of Reed-Solomon of degree d, and their goal is to verify that their codewords agree on at least h coordinates where $h \leq d + 1$. Then, the communication complexity of this problem is at least $h \cdot \log |\mathbb{F}|$.

The proof of the lower bound proceeds as follows. Given a protocol for $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$, we invoke it on inputs (x, a) and (y, b) where $x = y$. Obviously, on such inputs the protocol must reject, and therefore the players have to certify the existence of a violation, i.e., the existence of an index $i \in [m]$ such that $a_i \neq b_i$ but $X_i = Y_i$. We now consider two cases:

- **The protocol does not solve KW_f on (a, b):** In this case, when the protocol ends, the parties do not know any specific index i for which $a_i \neq b_i$. At best, they can confine the search for the index i to at most t rows (where t is a parameter that should be optimized). This means that in order to become convinced that $X_i = Y_i$ and $a_i \neq b_i$ for some index i, they have to certify that X and Y agree on at least t rows. We now use the foregoing agreement property with $h = t$ to argue that in order to perform the latter task, the parties must transmit at least $t \cdot n$ bits. This is larger than our desired lower bound for an appropriate choice of t.

- **The protocol solves KW_f on (a, b):** In this case, when the protocol ends, the players know some index i for which $a_i \neq b_i$, but they must have transmitted at least $\log \mathsf{L}(f) \approx \mathsf{C}(KW_f)$ bits in order to find it. In addition, we know that in order to detect a violation, the players must certify that $X_{i'} = Y_{i'}$ for some index i' (which may or may not be equal to i). We show that certifying this requires the parties to transmit at least n bits by using the foregoing agreement property (with $h = 1$).
  In short, the players must transmit $\log \mathsf{L}(f)$ bits for performing the first task, and n bits for the latter task. Using a double-counting argument, we show that the two quantities ($\log \mathsf{L}(f)$ and n) add up, and thus we get a lower bound of $\approx \log \mathsf{L}(f) + n$ bits, as required.

Actually, a protocol does not need to belong to one of the foregoing cases, but may consist of some combination of them: the protocol may solve KW_f for some seeds x and avoid solving it on others. Thus, in our actual proof, we partition the seeds x according to the case to which they belong, and combine the lower bounds from both cases. The details of this argument are provided in Section 4.

We note that the above proof strategy were developed by [GMWW17] for the (non-derandomized) composition $KW_{f \diamond \mathrm{U}_n}$. In particular, [GMWW17] used the fact that the set of all $m \times n$ matrices satisfies the foregoing agreement property. Our contribution is observing that this property is also satisfied by the set of matrices that correspond to Reed-Solomon codewords, and extending the argument to this setting.

Remark 1.9. The aforementioned "agreement property" of Reed-Solomon codes is inspired by the work of Dinur et. al. [DHSV15], who proved a robust variant of this property for Reed-Muller codes. They view this property as a derandomization of graph products.

2 Preliminaries

We reserve bold letters for random variables, and calligraphic letters for sets. We use $[n]$ to denote the set $\{1, \ldots, n\}$. We denote the set of $m \times n$ binary matrices by $\{0,1\}^{m \times n}$. For every binary $m \times n$ matrix X, we denote by $X_i \in \{0,1\}^n$ the i-th row of X.

Reed-Solomon codes. Let $\mathbb{F}$ be an arbitrary finite field, and let $\alpha_1, \ldots, \alpha_m$ be arbitrary distinct elements of $\mathbb{F}$ (where $m \leq |\mathbb{F}|$). For every $d \in \mathbb{N}$ such that $d \leq m$, the *Reed-Solomon code $\mathcal{C}$ of degree d is the subset of* $\mathbb{F}^m$ is defined as follows: A vector $\bar{c} \in \mathbb{F}^m$ belongs to $\mathcal{C}$ if and only if there exists a degree-d polynomial $p(x)$ such that $\bar{c} = (p(\alpha_1), \ldots, p(\alpha_m)) \in \mathbb{F}^m$. The elements of $\mathcal{C}$ are called *codeword*s. It is well-known that $\mathcal{C}$ is a linear subspace of dimension $d + 1$, and that every non-zero codeword has at most d coordinates that are equal to 0. See [MS78] for more details.

2.1 Formulas

Definition 2.1. A *(de-Morgan) formula* ϕ is a binary tree, whose leaves are identified with literals of the forms x_i and $\neg x_i$, and whose internal vertices are labeled as AND ($\wedge$) or OR ($\vee$) gates. A formula ϕ computes a Boolean function $f : \{0,1\}^n \to \{0,1\}$ in the natural way. The *size* of a formula is the number of its leaves (which is the same as the number of its wires up to a factor of 2). We note that a single input coordinate x_i can be associated with many leaves.

Definition 2.2. The *formula complexity* of a Boolean function $f : \{0,1\}^n \to \{0,1\}$, denoted $\mathsf{L}(f)$, is the size of the smallest formula that computes f. The *depth complexity* of f, denoted $\mathsf{D}(f)$, is the smallest depth of a formula that computes f.

The following definition generalizes the above definitions from functions to promise problems, which will be useful when we discuss Karchmer-Wigderson relations.

Definition 2.3. Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be disjoint sets. We say that a formula ϕ *separates* $\mathcal{X}$ and $\mathcal{Y}$ if $\phi(\mathcal{X}) = 0$ and $\phi(\mathcal{Y}) = 1$. The *formula complexity of the rectangle* $\mathcal{X} \times \mathcal{Y}$, denoted $\mathsf{L}(\mathcal{X} \times \mathcal{Y})$, is the size of the smallest formula that separates $\mathcal{X}$ and $\mathcal{Y}$. The *depth complexity of the rectangle* $\mathcal{X} \times \mathcal{Y}$, denoted $\mathsf{D}(\mathcal{X} \times \mathcal{Y})$, is the smallest depth of a formula that separates $\mathcal{X}$ and $\mathcal{Y}$.

Remark 2.4. Note that we define here the depth complexity of a function as the depth of a *formula* that computes f, while in the introduction we defined it as the depth of a *circuit with fan-in* 2 that computes f. However, for our purposes, this distinction does not matter, since every circuit with fan-in 2 can be converted into a formula with the same depth.

2.2 Communication complexity

Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be sets, and let $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ be a relation. The communication problem [Yao79] that corresponds to R is the following: two players, Alice and Bob, get inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, respectively. They would like to communicate and find $z \in \mathcal{Z}$ such that $(x, y, z) \in R$. At each round, one of the players sends a bit that depends on her/his input and on the previous messages, until they find z. The *communication complexity* of R is the minimal number of bits that is transmitted by a protocol that solves R. More formally, we define a protocol as a binary tree, in which every vertex represents a possible state of the protocol, and every edge represents a message that moves the protocol from one state to another:

Definition 2.5. A *(deterministic) protocol* that solves a relation $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ is a rooted binary tree with the following structure:

- Every node of the tree is labeled by a rectangle $\mathcal{X}_v \times \mathcal{Y}_v$ where $\mathcal{X}_v \subseteq \mathcal{X}$ and $\mathcal{Y}_v \subseteq \mathcal{Y}$. The root is labeled by the rectangle $\mathcal{X} \times \mathcal{Y}$. Intuitively, the rectangle $\mathcal{X}_v \times \mathcal{Y}_v$ is the set of pairs of inputs that lead the players to the node v.
- Each internal node v is *owned* by Alice or by Bob. Intuitively, v is owned by Alice if at state v, it is Alice's turn to speak, and same for Bob.
- Every edge of the tree is labeled by either 0 or 1.
- For every internal node v that is owned by Alice, the following holds: let v_0 and v_1 be the children of v associated with the out-going edges labeled with 0 and 1, respectively. Then,
  - $\mathcal{X}_v = \mathcal{X}_{v_0} \cup \mathcal{X}_{v_1}$, and $\mathcal{X}_{v_0} \cap \mathcal{X}_{v_1} = \emptyset$.
  - $\mathcal{Y}_v = \mathcal{Y}_{v_0} = \mathcal{Y}_{v_1}$.

  Intuitively, when the players are at the vertex v, Alice transmits 0 if her input is in $\mathcal{X}_{v_0}$ and 1 if her input is in $\mathcal{X}_{v_1}$. An analogous property holds for nodes owned by Bob, while changing the roles of $\mathcal{X}$ and $\mathcal{Y}$.
- For every leaf ℓ, there exists a value z such that $\mathcal{X}_\ell \times \mathcal{Y}_\ell \times \{z\} \subseteq R$. Intuitively, z is the output of the protocol at ℓ.

Definition 2.6. The *communication complexity* of a protocol Π, denoted $\mathsf{C}(\Pi)$, is the the depth of the protocol tree. In other words, it is the maximum number of bits that can be transmitted in an invocation of the protocol on any pair of inputs (x, y). For a relation R, we denote by $\mathsf{C}(R)$ the minimal communication complexity of a (deterministic) protocol that solves R.

Definition 2.7. Given a protocol Π, the *transcript* $\Pi(x, y)$ is the string that consists of the messages of Alice and Bob in the protocol when they get the inputs x and y, respectively. More formally, observe that for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, there is a unique leaf ℓ such that $(x, y) \in \mathcal{X}_\ell \times \mathcal{Y}_\ell$. The transcript $\Pi(x, y)$ is the string that is obtained by concatenating the labels of the edges on the path from the root to the leaf ℓ . We will sometimes identify $\Pi(x, y)$ with the leaf ℓ itself.

We now define a notion of protocol size that is analogous to the notion of formula size.

Definition 2.8. We define the *size* of a protocol Π to be its number of leaves. Note that this is also the number of distinct transcripts of the protocol. We define the *protocol size* of a relation R, denoted $\mathsf{L}(R)$, as the size of the smallest protocol that solves it (this is also known as the *protocol partition number* of R).

2.3 Karchmer-Wigderson relations

In this section, we define KW relations formally, and state the correspondence between KW relations and formulas. We start by defining KW relations for general rectangles, and then specialize the definition to functions.

Definition 2.9. Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be two disjoint sets. The *KW relation* $KW_{\mathcal{X}\times\mathcal{Y}} \subseteq \mathcal{X} \times \mathcal{Y} \times [n]$ is defined by

$$KW_{\mathcal{X}\times\mathcal{Y}} \stackrel{\text{def}}{=} \{(x, y, i) : x_i \neq y_i\}$$

Intuitively, $KW_{\mathcal{X}\times\mathcal{Y}}$ corresponds to the communication problem in which Alice gets $x \in \mathcal{X}$, Bob gets $y \in \mathcal{Y}$, and they would like to find a coordinate $i \in [n]$ such that $x_i \neq y_i$ (note that $x \neq y$ since $\mathcal{X} \cap \mathcal{Y} = \emptyset$).

Definition 2.10. Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant function. The *KW relation of* f, denoted KW_f, is defined by $KW_f \stackrel{\text{def}}{=} KW_{f^{-1}(0)\times f^{-1}(1)}$.

We are now ready to state the connection between formulas and KW relations. We state the connection for general rectangles, and the specialization to functions is straightforward.

Theorem 2.11 (Implicit in [KW90][6])**.** *Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be two disjoint sets. Then, for every formula ϕ that separates $\mathcal{X}$ and $\mathcal{Y}$, there exists a protocol Π_ϕ that solves $KW_{\mathcal{X}\times\mathcal{Y}}$, whose underlying tree is the same as the underlying tree of ϕ. In the other direction, for every protocol Π that solves $KW_{\mathcal{X}\times\mathcal{Y}}$ there exists a formula ϕ_Π that separates $\mathcal{X}$ and $\mathcal{Y}$, whose underlying tree is the same as the underlying tree of Π.*

Corollary 2.12 ([KW90])**.** *For every two disjoint sets $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ it holds that $\mathsf{D}(\mathcal{X} \times \mathcal{Y}) = \mathsf{C}(KW_{\mathcal{X}\times\mathcal{Y}})$, and $\mathsf{L}(\mathcal{X} \times \mathcal{Y}) = \mathsf{L}(KW_{\mathcal{X}\times\mathcal{Y}})$. In particular, for every non-constant $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathsf{D}(f) = \mathsf{C}(KW_f)$, and $\mathsf{L}(f) = \mathsf{L}(KW_f)$.*

2.3.1 Relaxed Karchmer-Wigderson problems

In this section, we review the notion of "relaxed KW problems", defined by [GMWW17]. Intuitively, these are KW relations that only require that the players *"almost"* find a coordinate i such that $x_i \neq y_i$. This relaxation turns out to be useful at a certain point in our proof of Theorem 1.8, where we want to argue that the players have to "almost" solve a KW relation.

More formally, given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a number $t \in \mathbb{N}$, the relaxed KW problem $KW_f(t)$ is a communication problem in which Alice wants to find a set $\mathcal{I}$ of size less than t such that $x|_{\mathcal{I}} \neq y|_{\mathcal{I}}$. This relaxes the definition of KW relations in two ways:

[6] This fact is discussed explicitly in [Raz90, KKN95, GMWW17].

1. Unlike the standard KW relation, Alice is not required to know a particular coordinate i such that $x_i \neq y_i$. Instead, she only needs to isolate it to a "small" set $\mathcal{I}$. The parameter t measures the amount of Alice's uncertainty about the coordinate i.

2. Moreover, unlike a standard KW relation, we do not require that at the end of the protocol, both players know the set $\mathcal{I}$. Instead, we only require that Alice knows the set $\mathcal{I}$.

The second relaxation above implies that a "relaxed KW problem" cannot be defined as a relation, in the same way we defined communication problems until this point. This leads us to the following definition of the relaxed KW problem.

Definition 2.13. Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant function and let $t \in \mathbb{N}$. Let Π be a protocol whose root is labeled by the rectangle $f^{-1}(0) \times f^{-1}(1)$. We say that Π solves the relaxed KW problem $KW_f(t)$ if it satisfies the following requirement:

- For every leaf ℓ of Π that is labeled by a rectangle $\mathcal{X}_\ell \times \mathcal{Y}_\ell$, and for every $x \in \mathcal{X}_\ell$, there exists a set $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| < t$, such that for every $y \in \mathcal{Y}_\ell$ it holds that $x|_{\mathcal{I}} \neq y|_{\mathcal{I}}$.

Remark 2.14. Note that in Definition 2.13, the fact that $\mathcal{I}$ is determined by both ℓ and x means that Alice knows the set $\mathcal{I}$, but Bob does not necessarily know it.

The following proposition, due to [GMWW17], says that the relaxed KW problem $KW_f(t)$ is not much easier than the original KW relation KW_f.

Proposition 2.15 ([GMWW17])**.** *Let $f : \{0,1\}^n \to \{0,1\}$, and let $t \in \mathbb{N}$. Then,*

$$\begin{aligned} \mathsf{C}(KW_f(t)) &\geq \mathsf{C}(KW_f) - t \cdot (\log n + 2) \\ \mathsf{L}(KW_f(t)) &\geq 2^{-t\cdot(\log n+2)} \cdot \mathsf{L}(KW_f). \end{aligned}$$

2.4 The universal relation and the derandomized composition

In this section, we define the universal relation and its derandomized composition formally. We use a slightly different definition than the one given in the introduction: In the definition given in the introduction, the players were promised that $x \neq y$. On the other hand, in the following definition, they are not given this promise, but are allowed to reject if the promise does not hold. This variant, which was suggested by [HW93], is usually more convenient to work with. It is also not hard to see that this modification does not change the complexity of the universal by much.

Definition 2.16. The *universal relation* U_n is defined as follows:

$$\mathrm{U}_n \stackrel{\text{def}}{=} \{(x,y,i) : x \neq y \in \{0,1\}^n, i \in [n], x_i \neq y_i\} \cup \{(x,x,\perp) : x \in \{0,1\}^n\}.$$

This corresponds to the communication problem in which Alice and Bob get strings x and y, respectively, and are required to output a coordinate i on which x and y differ, or the special rejection symbol $\perp$ if $x = y$.

Following [GMWW17], we also define the derandomized composition $KW_{(f\diamond \mathrm{U}_n)\circ\phi}$ using this "rejection" variant. Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant Boolean function, and let $\phi : \{0,1\}^\ell \to \{0,1\}^{m\times n}$. The relation $KW_{(f\diamond \mathrm{U}_n)\circ\phi}$ corresponds to the following communication problem: Alice gets as input strings $x \in \{0,1\}^\ell, a \in f^{-1}(0)$, Bob gets strings $y \in \{0,1\}^\ell, b \in f^{-1}(1)$. We define the matrices $X = \phi(x)$ and $Y = \phi(y)$. The goal of Alice and Bob is to find a coordinate k on which x and y differ, but they are allowed to reject if there exists an index $i \in [m]$ such that $a_i \neq b_i$ but $X_i = Y_i$. Formally,

Definition 2.17. Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant function, and let $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$. The relation $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$ is defined by

$$\begin{aligned} KW_{(f \diamond \mathrm{U}_n) \circ \phi} \stackrel{\text{def}}{=} & \left\{ ((x,a),(y,b),k) : x,y \in \{0,1\}^\ell, a \in f^{-1}(0), b \in f^{-1}(1), x_k \neq y_k \right\} \\ & \cup \left\{ ((x,a),(y,b),\perp) : x,y \in \{0,1\}^\ell, a \in f^{-1}(0), b \in f^{-1}(1), \exists i : a_i \neq b_i, \phi(x)_i = \phi(y)_i \right\}. \end{aligned}$$

Remark 2.18. We now explain why allowing inputs that violate the promise does not increase the complexity of the problem by much. Suppose that we have a protocol that solves the problem only on inputs that satisfy the promise. Then, we can construct a protocol that solves the problem on all inputs as follows: Alice and Bob run the original protocol on their inputs, thus obtaining a coordinate k. They then send each other the values x_k, y_k. If $x_k \neq y_k$, then k is a valid answer and they are done. If $x_k = y_k$, then this means that the original protocol failed, and therefore their inputs must violate the promise. Hence, the parties may safely reject.

The new protocol only sends two more bits than the original protocol. Hence, the complexity of the problem over all inputs is larger by at most two bits than the complexity of the promise problem.

2.5 Min-entropy

Given a random variable $\mathbf{x}$ that takes values from a finite set $\mathcal{X}$, its min-entropy $H_\infty(\mathbf{x})$ is defined as the largest number h such that $\Pr[\mathbf{x} = x] \leq 2^{-h}$ for every $x \in \mathcal{X}$. In other words,

$$H_\infty(\mathbf{x}) = \min_{x \in \mathcal{X}} \left\{ \log \frac{1}{\Pr[\mathbf{x} = x]} \right\}.$$

The following simple facts are useful.

Fact 2.19. *Let $\mathbf{x}$ be a random variable that takes values from a finite set $\mathcal{X}$. Then $H_\infty(\mathbf{x}) \leq \log |\mathcal{X}|$.*

Fact 2.20. *Let $\mathbf{x}$ be a random variable, and let $\mathcal{E}$ be an event. Then $H_\infty(\mathbf{x}|\mathcal{E}) \geq H_\infty(\mathbf{x}) - \log \frac{1}{\Pr[\mathcal{E}]}$.*

Proof. For every value x it holds that

$$\Pr[\mathbf{x} = x | \mathcal{E}] = \frac{\Pr[\mathbf{x} = x \wedge \mathcal{E}]}{\Pr[\mathcal{E}]} \leq \frac{\Pr[\mathbf{x} = x]}{\Pr[\mathcal{E}]} \leq 2^{-H_\infty(\mathbf{x}) + \log \frac{1}{\Pr[\mathcal{E}]}}.$$

It therefore follows that $H_\infty(\mathbf{x}|\mathcal{E}) \geq H_\infty(\mathbf{x}) - \log \frac{1}{\Pr[\mathcal{E}]}$, as required. ■

Fact 2.21. *Let $\mathbf{x}, \mathbf{x}'$ be independent identically distributed random variables. Then $\Pr[\mathbf{x} = \mathbf{x}'] \leq 2^{-H_\infty(\mathbf{x})}$.*

Proof. Let us denote the support of $\mathbf{x}, \mathbf{x}'$ by $\mathcal{X}$. Then

$$\begin{aligned} \Pr[\mathbf{x} = \mathbf{x}'] &= \sum_{x \in \mathcal{X}} \Pr[\mathbf{x} = \mathbf{x}' = x] \\ &= \sum_{x \in \mathcal{X}} \Pr[\mathbf{x} = x]^2 \\ &\leq \max_{x \in \mathcal{X}} \{\Pr[\mathbf{x} = x]\} \cdot \sum_{x \in \mathcal{X}} \Pr[\mathbf{x} = x] \\ &= \max_{x \in \mathcal{X}} \{\Pr[\mathbf{x} = x]\} \\ &= 2^{-H_\infty(\mathbf{x})}, \end{aligned}$$

as required. ■

The following lemma says that if a random string $\mathbf{x} \in \{0,1\}^n$ has close-to-full min-entropy, then it is possible to fix a few "bad" bits of $\mathbf{x}$ such that all projections of the good bits have close-to-full min-entropy. Similar lemmas were proved in [GLM+16, DM16, KMR17].

Lemma 2.22. *Let $\mathbf{x}$ be a random variable taking values from $\{0,1\}^n$ such that $H_\infty(\mathbf{x}) \geq n - t$. Then, for every $b \in \mathbb{N}$, there exists a set $\mathcal{B} \subseteq [n]$ of "bad coordinates" of size at most b and an event $\mathcal{E}$ such that for every $\mathcal{S} \subseteq [n] - \mathcal{B}$ it holds that*

$$H_\infty(\mathbf{x}|_{\mathcal{S}} \,|\, \mathcal{E}) \geq \left(1 - \frac{t}{b}\right) |\mathcal{S}| \,. \tag{4}$$

Proof. Let $\mathcal{B}$ be a a maximal set of coordinates that violates Inequality 4, i.e., a maximal set such that

$$H_\infty(\mathbf{x}|_{\mathcal{B}}) < \left(1 - \frac{t}{b}\right) |\mathcal{B}|$$

Then, there exists some value $x_{\mathcal{B}} \in \{0,1\}^{|\mathcal{B}|}$ such that

$$\Pr\left[\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}}\right] > 2^{-\left(1-\frac{t}{b}\right)|\mathcal{B}|}.$$

Let $\mathcal{E}$ be the event that $\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}}$. We show that these choices of $\mathcal{B}, \mathcal{E}$ satisfy the requirements of the lemma. Let $\mathcal{S} \subseteq [n] - \mathcal{B}$. We show that $\mathcal{S}$ satisfies Inequality 4. Suppose otherwise. Then, there exists a value $x_{\mathcal{S}} \in \{0,1\}^{|\mathcal{S}|}$ such that

$$\Pr\left[\mathbf{x}|_{\mathcal{S}} = x_{\mathcal{S}} | \mathcal{E}\right] > 2^{-\left(1-\frac{t}{b}\right)|\mathcal{S}|},$$

and therefore

$$\Pr\left[\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}} \wedge \mathbf{x}|_{\mathcal{S}} = x_{\mathcal{S}}\right] = \Pr\left[\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}}\right] \cdot \Pr\left[\mathbf{x}|_{\mathcal{S}} = x_{\mathcal{S}} | \mathcal{E}\right] > 2^{-\left(1-\frac{t}{b}\right)(|\mathcal{B}|+|\mathcal{S}|)}.$$

It follows that

$$H_\infty(\mathbf{x}|_{\mathcal{B}\cup\mathcal{S}}) < \left(1 - \frac{t}{b}\right) |\mathcal{B} \cup \mathcal{S}| \,,$$

contradicting the minimality of $\mathcal{B}$. Thus, $\mathcal{S}$ satisfies Inequality 4.

We turn to show that $|\mathcal{B}| \leq b$. Suppose otherwise. Observe that $H_\infty(\mathbf{x}|_{[n]-\mathcal{B}}|\mathcal{E}) \leq n - |\mathcal{B}|$ and hence there exists a value $x_{[n]-\mathcal{B}}$ such that

$$\Pr\left[\mathbf{x}|_{[n]-\mathcal{B}} = x_{[n]-\mathcal{B}} | \mathcal{E}\right] \geq 2^{-(n-|\mathcal{B}|)}.$$

Then, it follows that

$$\begin{aligned}
\Pr\left[\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}} \wedge \mathbf{x}|_{[n]-|\mathcal{B}|} = x_{[n]-|\mathcal{B}|}\right] &= \Pr\left[\mathbf{x}|_{\mathcal{B}} = x_{\mathcal{B}}\right] \cdot \Pr\left[\mathbf{x}|_{[n]-|\mathcal{B}|} = x_{[n]-|\mathcal{B}|} | \mathcal{E}\right] \\
&> 2^{-\left[(n-|\mathcal{B}|)+\left(1-\frac{t}{b}\right)|\mathcal{B}|\right]} \\
&= 2^{-\left(n-\frac{t}{b}\cdot|\mathcal{B}|\right)} \\
\text{(Since we assumed } |\mathcal{B}| > b) &> 2^{-(n-t)},
\end{aligned}$$

contradicting the assumption that $H_\infty(\mathbf{x}) \geq n - t$. Thus, $|\mathcal{B}| \leq b$, as required. ■

2.6 Averaging Samplers

Intuitively, an averaging sampler [Zuc97] is a family of subsets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ of some universe $[\ell]$, such that for every function $f : [\ell] \to [0,1]$, the average value of f on $[\ell]$ is well-approximated by the average value of f on a typical set $\mathcal{S}_i$. In the following definition, we restrict ourselves to Boolean functions $f : [\ell] \to \{0,1\}$, since this is all we need in this paper. We view such a function f as the indicator function of some set $\mathcal{R} \subseteq [\ell]$.

Definition 2.23. Let $\ell \in \mathbb{N}$, and let $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ be a family of subsets of $[\ell]$ of size n. We say that $\mathcal{F}$ is an (α, ε)-*averaging sampler of* $[\ell]$ if for every set $\mathcal{R} \subseteq [\ell]$ of density $\delta \stackrel{\text{def}}{=} \frac{|\mathcal{R}|}{\ell}$, it holds for all but α fraction of the sets $\mathcal{S}_i$ in $\mathcal{F}$ that

$$\delta - \varepsilon \leq \frac{|\mathcal{R} \cap \mathcal{S}_i|}{n} \leq \delta + \varepsilon.$$

Averaging samplers are equivalent to (seeded) randomness extractors [NZ96, Zuc97]. Using the known explicit constructions of extractors, we obtain the following construction of averaging samplers.

Theorem 2.24 (Follows from [Zuc97, RRV02])**.** *There exists a universal constant q such that the following holds: there is an algorithm that when given as input ℓ, m, n, ε such that $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$, runs in time $\mathrm{poly}(\ell, m, n, 1/\varepsilon)$ and outputs an (α, ε)-averaging sampler $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ of $[\ell]$ where the sets are of size n and where*

$$\alpha = \frac{q}{\varepsilon^2} \cdot \frac{\ell}{m \cdot n}.$$

We derive Theorem 2.24 from known constructions of extractors in Appendix A.

3 Derandomizing Composition via Samplers

In this section we prove Theorem 1.3, restated next.

Theorem 1.3. *There exists a universal constant $q \in \mathbb{N}$ such that the following holds. Let $\varepsilon > 0$ be an arbitrary constant, and let $m, n \in \mathbb{N}$ be such that $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$. Then, for every $f : \{0,1\}^m \to \{0,1\}$ there exists a generator $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ such that $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$ and such that*

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \left(1 - \frac{q}{\varepsilon^4 \cdot n}\right) \cdot \mathsf{C}(KW_f) + (1 - 3\varepsilon) \cdot n - 2\log m - 1.$$

To this end, we prove the following theorem, which implies Theorem 1.3 as a corollary:

Theorem 3.1. *Let $f : \{0,1\}^m \to \{0,1\}$ be a function, let $n \in \mathbb{N}$, let $\varepsilon > 0$, and let $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$. Let $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ be an (α, ε)-averaging sampler of $[\ell]$ where the sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are of size n, and let $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ be the generator that maps a string $x \in \{0,1\}^\ell$ to the $m \times n$ matrix X whose i-th row is $x|_{S_i}$ for every $i \in [m]$. Then*

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \mathsf{C}(KW_f) - 2\alpha \cdot m + (1 - 3\varepsilon) \cdot n - 2\log m - 1$$

Proof of Theorem 1.3 from Theorem 3.1. Let q' be the universal constant from the construction of averaging samplers in Theorem 2.24, and let $q \stackrel{\text{def}}{=} 2 \cdot q'$. Let $m, n \in \mathbb{N}$, $\varepsilon > 0$ be such that $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$. Let $f : \{0,1\}^m \to \{0,1\}$, let $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$, and let $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ be an (α, ε)-averaging sampler of $[\ell]$ where the sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are of size n and where

$$\alpha = \frac{q'}{\varepsilon^2} \cdot \frac{\ell}{m \cdot n}$$

(such an averaging sampler exists by Theorem 2.24). Finally, let $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ be the generator defined in Theorem 3.1 with respect to $\mathcal{F}$.

We now consider two cases: either $\ell = \mathsf{C}(KW_f)/\varepsilon^2$ or $\ell = n$. If $\ell = \mathsf{C}(KW_f)/\varepsilon^2$, then Theorem 3.1 implies that

$$\begin{aligned}
\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) &\geq \mathsf{C}(KW_f) - 2\alpha \cdot m + (1 - 3\varepsilon) \cdot n - 2\log m - 1 \\
&= \mathsf{C}(KW_f) - \frac{2q'}{\varepsilon^2} \cdot \frac{\ell}{m \cdot n} \cdot m + (1 - 3\varepsilon) \cdot n - 2\log m - 1 \\
&= \mathsf{C}(KW_f) - \frac{q}{\varepsilon^2} \cdot \frac{\mathsf{C}(KW_f)/\varepsilon^2}{m \cdot n} \cdot m + (1 - 3\varepsilon) \cdot n - 2\log m - 1 \\
&= \left(1 - \frac{q}{\varepsilon^4 \cdot n}\right) \cdot \mathsf{C}(KW_f) + (1 - 3\varepsilon) \cdot n - 2\log m - 1.
\end{aligned}$$

as required.

On the other hand, if $\ell = n$, then in this case the lower bound we wish to prove is at most n. We prove this lower bound by direct reduction from the universal relation: Given a protocol for $\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi})$, we can solve the universal relation U_n as follows. We fix some $a \in f^{-1}(0), b \in f^{-1}(1)$ that disagree on exactly one coordinate, and denote this coordinate by i. Given inputs $x, y \in \{0,1\}^n$ for U_n, the parties construct inputs $x', y' \in \{0,1\}^\ell$ for ϕ by setting the i-th rows X_i, Y_i to x, y respectively. The parties now invoke the protocol for $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$ on inputs (x', a) and (y', b), thus obtaining either a coordinate on which x and y differ or a rejection (in which case they reject). Note that the complexity of this protocol is exactly the complexity of the original protocol for $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$.. ■

In the rest of this section we focus on proving Theorem 3.1. Let $f : \{0,1\}^m \to \{0,1\}$ be a function, let $n \in \mathbb{N}$, let $\varepsilon > 0$, and let $\ell = \max\left\{\mathsf{C}(KW_f)/\varepsilon^2, n\right\}$. Let $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ be an (α, ε)-sampler and let $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ be a generator as in Theorem 3.1. Suppose that there is a protocol Π that solves $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$ and is "too efficient", i.e.,

$$\mathsf{C}(\Pi) < \mathsf{C}(KW_f) - 2 \cdot \alpha \cdot m + (1 - 3 \cdot \varepsilon) \cdot n - 2 \cdot \log m - 1.$$

We show how to find a transcript π and inputs (x, a), (y, b) on which the protocol errs. To this end, we will invoke the protocol on inputs of the form $(x, a), (x, b)$, i.e., the seeds of the parties are identical. As described in Section 1.3.1, we partition the protocol to two stages, where the first stage ends when the players transmitted $\mathsf{C}(KW_f) - \alpha \cdot m - \log m - 1$ bits. The first step of the proof is to show that when the first stage ends, the parties have not solved KW_f yet, and also that they have not transmitted more than $\varepsilon^2 \cdot \ell$ bits of information about x. Then, we perform a "clean-up" step which restricts the set of possible inputs such that it satisfies the following properties:

- On all but α fraction of the rows of $X = \phi(x)$, the parties transmitted at most $3 \cdot \varepsilon \cdot n + \alpha \cdot m$ bits of information.
- On each of the remaining rows, it holds that $a_i = b_i$.

The last step in the proof is to use the latter two properties to show that when the second stage ends, we can find inputs on which the protocol errs. The rest of this section is organized as follows: The analysis of the first stage is provided in Section 3.1, the clean-up step is described in Section 3.2, and the second stage is analyzed in Section 3.3.

3.1 The first stage

Our goal in the analysis of the first stage is to show that there is a transcript of this stage in which the parties do not solve KW_f and do not transmit too much information about their inputs. In order to formalize this notion, we introduce some notation. For every partial transcript π of the protocol, recall that we denote by $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ the rectangle that consists of the inputs $(x,a),(y,b)$ that are consistent with π. Given a string $x \in \{0,1\}^\ell$, we denote by $\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}$ the rectangle of inputs to KW_f that is obtained by fixing x to be the seed of both parties. Formally,

$$\mathcal{A}_{\pi,x} \stackrel{\text{def}}{=} \left\{ a \in f^{-1}(0) \middle| (x,a) \in \mathcal{X}_\pi \right\}$$
$$\mathcal{B}_{\pi,x} \stackrel{\text{def}}{=} \left\{ b \in f^{-1}(1) \middle| (x,b) \in \mathcal{Y}_\pi \right\}.$$

We denote by $\mathsf{C}(\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x})$ the communication complexity of the residual KW relation on $\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}$, namely, $KW_{\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}}$. The following definition captures the properties that we require from the transcript of the first stage.

Definition 3.2. We say that a partial transcript of the protocol is *alive* if for at least $2^{-\varepsilon^2 \cdot \ell}$ fraction of the strings $x \in \{0,1\}^\ell$ it holds that

$$\mathsf{C}(\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}) \geq \alpha \cdot m + \log m + 1. \tag{5}$$

We now prove that the adversary can find a live transcript for the first stage.

Lemma 3.3. *There exists a live transcript π^1 of length* $\min\left\{\mathsf{C}(\Pi), \mathsf{C}(KW_f) - \alpha \cdot m - \log m - 1\right\}$.

Proof. Let $c = \min\left\{\mathsf{C}(\Pi), \mathsf{C}(KW_f) - \alpha \cdot m - \log m - 1\right\}$. For every $x \in \{0,1\}^\ell$, we denote by Π_x the protocol that is obtained by hard-wiring x as the seed of both parties in the protocol Π. We will prove that for every $x \in \{0,1\}^\ell$, there is a transcript π_x of Π_x of length c that satisfies Inequality 5. Since there are at most 2^c transcripts of length c, we will conclude that at least one of the transcripts π_x is "good" for at least 2^{-c} fraction of the seeds. Details follow.

We first prove that every seed $x \in \{0,1\}^\ell$ there is a corresponding transcript π_x. Let $x \in \{0,1\}^\ell$. Suppose for the sake of contradiction that all the partial transcripts of Π_x of length c do not satisfy Inequality 5. Then, we can obtain from Π_x a protocol that solves KW_f using less than $\mathsf{C}(KW_f)$ bits: the new protocol would simulate Π_x for c bits, thus reaching some partial transcript π, and then proceed with the optimal protocol for the rectangle $\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}$. Therefore there must exist at least one transcript of Π_x that satisfies Inequality 5, and we denote it by π_x.

Next, since all these transcripts π_x are of length c, there are at most 2^c distinct transcripts π_x. This implies that there exists some partial transcript π^1 such that $\pi^1 = \pi_x$ for at least 2^{-c} fraction of the seeds. Since $2^{-c} \geq 2^{-\varepsilon^2 \cdot \ell}$ (by the choice of ℓ), it follows that π^1 is alive and of length c, as required. ■

3.2 The clean-up

Let π^1 be the live transcript from Lemma 3.3, and let $\mathcal{T}_1$ be the set of strings $x \in \{0,1\}^\ell$ for which the rectangle $\mathcal{A}_{\pi^1,x} \times \mathcal{B}_{\pi^1,x}$ satisfies Inequality 5. Our goal in the clean-up step is to construct a subset of $\mathcal{T}_1$ and corresponding inputs (a,b) to KW_f that satisfy the following properties:

- On all but α fraction of the rows of $X = \phi(x)$, the parties transmitted at most $3\varepsilon \cdot n + \alpha \cdot m$ bits of information.
- On each of the remaining rows, it holds that $a_i = b_i$.

We start by constructing a subset $\mathcal{T}_2 \subseteq \mathcal{T}_1$ that satisfies a slightly stronger version of the first property by proving the following result.

Lemma 3.4. *There exists a subset $\mathcal{T}_2 \subseteq \mathcal{T}_1$ and a set of rows $\mathcal{R} \subseteq [m]$ of size at most $\alpha \cdot m$ such that the following holds: Let $\mathbf{x}$ be a random variable that is uniformly distributed over $\mathcal{T}_2$, and let $\mathbf{X} = \phi(\mathbf{x})$. Then, for every $i \in [m] - \mathcal{R}$ it holds that $H_\infty(\mathbf{X}_i) \geq n - 3\varepsilon \cdot n$.*

Proof. Let $\mathbf{x}^1$ be a random variable that is uniformly distributed over $\mathcal{T}_1$. Observe that

$$H_\infty(\mathbf{x}^1) = \log |\mathcal{T}_1| \geq \ell - \varepsilon^2 \cdot \ell.$$

By applying Lemma 2.22 with $b = \varepsilon \cdot \ell$, it follows that there is a set $\mathcal{B}$ of "bad coordinates" of size at most $\varepsilon \cdot \ell$ and an event $\mathcal{T}_2 \subseteq \mathcal{T}_1$ such that for every $\mathcal{S} \subseteq [\ell] - \mathcal{B}$ it holds that

$$H_\infty(\mathbf{x}^1|_{\mathcal{S}} \big| \, \mathbf{x}^1 \in \mathcal{T}_2) \geq (1 - \frac{\varepsilon^2 \cdot \ell}{\varepsilon \cdot \ell}) \cdot |\mathcal{S}| = (1 - \varepsilon) \cdot |\mathcal{S}| \,.$$

Let $\mathbf{x}$ be a random variable that is uniformly distributed over $\mathcal{T}_2$, so for every $\mathcal{S} \subseteq [\ell] - \mathcal{B}$ it holds that $H_\infty(\mathbf{x}|_{\mathcal{S}}) \geq (1 - \varepsilon) \cdot |\mathcal{S}|$. Recall that we denote by $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ the (α, ε)-sampler that is used to construct the generator ϕ. Then, by the property of the sampler it follows that for at least $(1 - \alpha)$ fraction of the indices $i \in [m]$ it holds that

$$|\mathcal{S}_i \cap \mathcal{B}| \leq (\frac{|\mathcal{B}|}{\ell} + \varepsilon) \cdot n \leq 2 \cdot \varepsilon \cdot n.$$

Let $\mathcal{R} \subseteq [m]$ be the set of indices that violate the latter equality. The set $\mathcal{R}$ correspond to the "revealed rows" of $\mathbf{X}$, i.e., the rows on which too much information was revealed. Then, for every $i \in [m] - \mathcal{R}$ it holds that

$$\begin{aligned} H_\infty(\mathbf{X}_i) &= H_\infty(\mathbf{x}|_{\mathcal{S}_i}) \\ &\geq H_\infty(\mathbf{x}|_{\mathcal{S}_i - \mathcal{B}}) \\ &\geq (1 - \varepsilon) \cdot |\mathcal{S}_i - \mathcal{B}| \\ &\geq (1 - \varepsilon) \cdot |\mathcal{S}_i| - |\mathcal{S}_i \cap B| \\ &\geq (1 - \varepsilon) \cdot n - 2 \cdot \varepsilon \cdot n \\ &= n - 3 \cdot \varepsilon \cdot n, \end{aligned}$$

as required. ■

We now restrict the inputs further in order to make sure that a and b agree on the coordinates in $\mathcal{R}$.

Lemma 3.5. *There exist a set $\mathcal{T}_3 \subseteq \mathcal{T}_2$ and strings $a_x \in f^{-1}(0)$, $b_x \in f^{-1}(1)$ for every $x \in \mathcal{T}_3$ such that the following properties hold:*

- *For every $x \in \mathcal{T}_3$, the inputs (x, a_x), (x, b_x) are consistent with the transcript π^1 (i.e., $a_x \in \mathcal{A}_{\pi^1, x}$ and $b_x \in \mathcal{B}_{\pi^1, x}$).*

- *For every $x, y \in \mathcal{T}_3$ it holds that $a_x|_{\mathcal{R}} = b_y|_{\mathcal{R}}$.*
- *If $\mathbf{x}$ is a random variable that is uniformly distributed over $\mathcal{T}_3$, and $\mathbf{X} = \phi(\mathbf{x})$, then for every $i \in [m] - \mathcal{R}$ it holds that*
$$H_\infty(\mathbf{X}_i) \geq n - 3 \cdot \varepsilon \cdot n - \alpha \cdot m.$$

Proof. We first claim that for every $x \in \mathcal{T}_2$, there exist strings $a_x \in \mathcal{A}_{\pi^1,x}$, $b_x \in \mathcal{B}_{\pi^1,x}$ such that $a_x|_{\mathcal{R}} = b_x|_{\mathcal{R}}$: To see why, recall that by the definition of $\mathcal{T}_1$ it holds that

$$\mathsf{C}(\mathcal{A}_{\pi^1,x} \times \mathcal{B}_{\pi^1,x}) \geq \alpha \cdot m + \log m + 1. \tag{6}$$

On the other hand, if such strings a_x, b_x did not exist, then it would mean that for every $a \in \mathcal{A}_{\pi^1,x}, b \in \mathcal{B}_{\pi^1,x}$ it holds that $a|_{\mathcal{R}} \neq b|_{\mathcal{R}}$. In such case, Alice and Bob could solve $KW_{\mathcal{A}_{\pi^1,x} \times \mathcal{B}_{\pi^1,x}}$ using $\alpha \cdot m + \log m$ bits as follows: On inputs a, b, Alice would send $a|_{\mathcal{R}}$ to Bob, and then Bob would reply with a coordinate $i \in \mathcal{R}$ such that $a_i \neq b_i$. This would imply that

$$\mathsf{C}(\mathcal{A}_{\pi^1,x} \times \mathcal{B}_{\pi^1,x}) \leq \alpha \cdot m + \log m,$$

contradicting Inequality 6.

Let us fix some choice of the latter strings a_x, b_x for every $x \in \mathcal{T}_2$. Now, let us label every $x \in \mathcal{T}_2$ with the label $a_x|_{\mathcal{R}}$, and let $r \in \{0,1\}^{|\mathcal{R}|}$ denote the most popular label. Let $\mathcal{T}_3$ be the set of seeds $x \in \mathcal{T}_2$ that are labeled with r, so $a_x|_{\mathcal{R}} = b_x|_{\mathcal{R}} = r$. Then, for every $x, y \in \mathcal{T}_3$ it holds that $a_x|_{\mathcal{R}} = b_y|_{\mathcal{R}}$, as required by the lemma. Furthermore, it holds that

$$|\mathcal{T}_3| \geq 2^{-|\mathcal{R}|} \cdot |\mathcal{T}_2| \geq 2^{-\alpha \cdot m} \cdot |\mathcal{T}_2|.$$

Now, let $\mathbf{x}$ be a random variable that is uniformly distributed over $\mathcal{T}_3$, and let $\mathbf{x}^2$ be a random variable that is uniformly distributed over $\mathcal{T}_2$. Observe that the distribution of $\mathbf{x}$ is exactly the distribution of $\mathbf{x}^2|\mathbf{x}^2 \in \mathcal{T}_3$. Let $\mathbf{X} = \phi(\mathbf{x})$ and $\mathbf{X}^2 = \phi(\mathbf{x}^2)$. Now, by Lemma 3.4, it holds for every $i \in [m] - \mathcal{R}$ that

$$H_\infty(\mathbf{X}^2_i) \geq n - 3 \cdot \varepsilon \cdot n.$$

Moreover, it holds that $\Pr\left[\mathbf{x}^2 \in \mathcal{T}_3\right] \geq 2^{-\alpha \cdot m}$. Hence, by Fact 2.20, for every $i \in [m] - \mathcal{R}$ it holds that

$$\begin{aligned} H_\infty(\mathbf{X}_i) &= H_\infty(\mathbf{X}^2_i \big| \mathbf{x}^2 \in \mathcal{T}_2) \\ &\geq H_\infty(\mathbf{X}^2_i) - \log \frac{1}{\Pr\left[\mathbf{x}^2 \in \mathcal{T}_3\right]} \\ &\geq n - 3 \cdot \varepsilon \cdot n - \alpha \cdot m, \end{aligned}$$

as required. ■

3.3 The second stage

Our goal in the analysis of the second stage is to construct a suffix π^2 such that $\pi \stackrel{\text{def}}{=} \pi^1 \circ \pi^2$ is a full transcript of the protocol that errs on some inputs. To this end, let $\mathcal{T}_3$ and $\mathcal{R}$ be as in Lemma 3.5 and let $\mathbf{x}$ be a uniformly distributed seed in $\mathcal{T}_3$. We invoke the protocol on inputs $(\mathbf{x}, a_{\mathbf{x}})$ and $(\mathbf{x}, b_{\mathbf{x}})$ starting at the end of the first stage (i.e., after the parties transmitted π_1), and run it until the protocol ends. Let π^2 be the most likely suffix, i.e., π^2 is the strings of bits that has the highest probability to be transmitted after π^1.

Observe that $\pi \stackrel{\text{def}}{=} \pi^1 \circ \pi^2$ is a transcript in which the parties reject the input (i.e., they output $\perp$). The reason is that this transcript was obtained by giving the parties identical seeds x, y, and therefore they are unable to find a coordinate on which their seeds differ. We would like to show that this transcript errs on some inputs. To this end, we need to show that there exist inputs (x, a_x) and (y, b_y) that are consistent with π but must not be rejected. Namely, those inputs should satisfy the property that for every $i \in [m]$, if $(a_x)_i \neq (b_y)_i$ then $X_i \neq Y_i$. Since it always holds that $a_x|_{\mathcal{R}} = b_y|_{\mathcal{R}}$, it suffices to show that there exist inputs (x, a_x) and (y, b_y) such that $X_i \neq Y_i$ for every $i \in [m] - \mathcal{R}$. We now show that such inputs exist.

Let $\mathcal{T}_4 \subseteq \mathcal{T}_3$ be the event in which π^2 is transmitted, i.e., $\mathcal{T}_4$ is the set of seeds $x \in \mathcal{T}_3$ such that the protocol transmits π^2 on inputs (x, a_x), (x, b_x). We prove a lower bound on the probability of $\mathcal{T}_4$ and use it to show that the protocol errs. Recall that we assumed that the protocol Π transmits less than

$$\mathsf{C}(KW_f) - 2 \cdot \alpha \cdot m + (1 - 3 \cdot \varepsilon) \cdot n - 2 \cdot \log m - 1$$

bits. Moreover, in the first stage the parties transmitted

$$\min \{\mathsf{C}(\Pi), \mathsf{C}(KW_f) - \alpha \cdot m - \log m - 1\}$$

bits. Therefore, in the second stage the parties transmit less than

$$(1 - 3 \cdot \varepsilon) \cdot n - \alpha \cdot m - \log m$$

bits. Hence,

$$\Pr[\mathbf{x} \in \mathcal{T}_4] > 2^{-[(1-3\cdot\varepsilon)\cdot n - \alpha \cdot m - \log m]}.$$

Let $\mathbf{X} = \phi(\mathbf{x})$. By Lemma 3.5 and Fact 2.20, it follows that for every $i \in [m] - \mathcal{R}$ it holds that

$$\begin{aligned} H_\infty(\mathbf{X}_i | \mathbf{x} \in \mathcal{T}_4) &\geq H_\infty(\mathbf{X}_i) - \log \frac{1}{\Pr[\mathbf{x} \in \mathcal{T}_4]} \\ &> n - 3 \cdot \varepsilon \cdot n - \alpha \cdot m \\ &\quad - [(1 - 3 \cdot \varepsilon) \cdot n - \alpha \cdot m - \log m] \\ &\geq \log m. \end{aligned}$$

Now, let $\boldsymbol{y}$ be a random variable that is independent and identically distributed to $\mathbf{x}$, and let $\boldsymbol{Y} = \phi(\boldsymbol{y})$. Then, for every $i \in [m] - \mathcal{R}$, the random variables

$$\mathbf{X}_i | \mathbf{x} \in \mathcal{T}_4, \boldsymbol{Y}_i | \boldsymbol{y} \in \mathcal{T}_4$$

are independent and identically distributed, and therefore the probability that they are equal is at most

$$2^{-H_\infty(\mathbf{X}_i | \mathbf{x} \in \mathcal{T}_4)} < \frac{1}{m}$$

by Fact 2.21. By the union bound, it follows that with non-zero probability, it holds that $\mathbf{X}_i \neq \boldsymbol{Y}_i$ for every $i \in [m] - \mathcal{R}$ conditioned on $\mathbf{x}, \boldsymbol{y} \in \mathcal{T}_4$, and therefore there exist particular choices $x, y \in \mathcal{T}_4$ of such inputs. Finally, observe that the inputs (x, a_x), (y, b_y) satisfy that for every $i \in [m]$, if $(a_x)_i \neq (b_y)_i$ then $X_i \neq Y_i$, and therefore they should not be rejected. However, the protocol Π rejects those inputs, and therefore the protocol errs.

We reached a contradiction, and therefore a "too efficient" protocol does not exist. It follows that

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \mathsf{C}(KW_f) - 2\alpha \cdot m + (1 - 3\varepsilon) \cdot n - 2 \log m - 1,$$

as required.

4 Derandomizing Composition via Reed-Solomon Codes

In this section we prove Theorem 1.8, restated next.

Theorem 1.8. *Let $m, n \in \mathbb{N}$ be such that $m < 2^n$. Then, there exists a generator $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ such that $\ell \leq 5 \cdot m + n$, and such that for every $f : \{0,1\}^m \to \{0,1\}^n$ it holds that*

$$\mathsf{C}(KW_{(f \diamond \mathrm{U}_n) \circ \phi}) \geq \log \mathsf{L}(f) + n - 4 \cdot (1 + \frac{m}{n}) \cdot \log m.$$

As described in the introduction, the generator $\phi : \{0,1\}^\ell \to \{0,1\}^{m \times n}$ of the theorem is defined as follows. Let $m, n \in \mathbb{N}$ be such that $m < 2^n$, and let $d \stackrel{\text{def}}{=} \lceil 3 \cdot \frac{m}{n} \rceil$. Let $\mathbb{F}$ be the finite field of size 2^n, and let $\alpha_1, \ldots, \alpha_m \in \mathbb{F}$ be distinct elements. Let $\mathcal{C} \subseteq \mathbb{F}^m$ be the corresponding Reed-Solomon of degree d, and recall that $\mathcal{C}$ is a linear space of dimension $d+1$ so $|\mathcal{C}| = |\mathbb{F}|^{d+1} = 2^{(d+1) \cdot n}$. We choose $\ell \stackrel{\text{def}}{=} (d+1) \cdot n$, and choose the generator to be some bijection from $\{0,1\}^\ell$ to $\mathcal{C}$ — for example, one can choose the Reed-Solomon encoding described in Section 1.2.

Fix a protocol Π for $KW_{(f \diamond \mathrm{U}_n) \circ \phi}$. We prove that

$$\log \mathsf{L}(\Pi) \geq \log \mathsf{L}(f) + n - 4 \cdot (1 + \frac{m}{n}) \cdot \log m,$$

and since it always holds that $\mathsf{C}(\Pi)] \geq \log \mathsf{L}(\Pi)$ the required result will follow. To this end, we analyze the behavior of the protocol when Alice and Bob get identical seeds as inputs. The following definition is useful.

Definition 4.1. Let π be a (full) transcript of Π, and let $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ be the corresponding rectangle.

- We say that the transcript π *supports* a seed $x \in \{0,1\}^\ell$ if x can be given as an input to both parties at π. Formally, π supports x if there exist $a, b \in \{0,1\}^m$ such that $(x, a) \in \mathcal{X}_\pi$ and $(x, b) \in \mathcal{Y}_\pi$. We also say that X is *supported by π and a*, or *by π and b*. Note that the transcript π must be a transcript that outputs $\perp$, since the parties cannot find a coordinate on which their seeds differ.
- We say that the transcript π *supports* $a \in f^{-1}(0)$ if a can be given as input to Alice at π. Formally, π supports a if there exists a seed $x \in \{0,1\}^\ell$ such that $(x, a) \in \mathcal{X}_\ell$. A similar definition applies to strings $b \in f^{-1}(1)$.

In order to prove lower bound on $\mathsf{L}(\Pi)$, we double-count the number of pairs (π, x), where π is a transcript that outputs $\perp$, and $x \in \{0,1\}^\ell$ is a seed that is supported by π. Specifically, we prove the following lemmas.

Lemma 4.2. *The number of pairs (π, x) is at most $\mathsf{L}(\Pi) \cdot |\mathbb{F}|^d$.*

Lemma 4.3. *The number of pairs (π, x) is at least $2^{-4 \cdot (1 + \frac{m}{n}) \cdot \log m} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f)$.*

By combining the two lemmas, we get:

$$\begin{aligned} \mathsf{L}(\Pi) \cdot |\mathbb{F}|^d &\geq 2^{-4 \cdot (1 + \frac{m}{n}) \cdot \log m} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f) \\ \mathsf{L}(\Pi) &\geq 2^{-4 \cdot (1 + \frac{m}{n}) \cdot \log m} \cdot |\mathbb{F}| \cdot \mathsf{L}(f) \\ \log \mathsf{L}(\Pi) &\geq \log \mathsf{L}(f) + n - 4 \cdot (1 + \frac{m}{n}) \cdot \log m, \end{aligned}$$

as required. The rest of this section is organized as follows: We start by proving a useful combinatorial lemma about Reed-Solomon codes in Section 4.1. We then prove Lemmas 4.2 and 4.3 in Sections 4.2 and 4.3 respectively.

4.1 A combinatorial lemma

In this section we prove a combinatorial lemma about Reed-Solomon codes. In order to motivate this lemma, consider the communication problem in which each of Alice and Bob gets codeword of Reed-Solomon of degree d, and their goal is to verify that their codewords agree on at least h coordinates where $h \leq d+1$. Then, our lemma implies that the communication complexity of this problem is at least $h \cdot \log|\mathbb{F}|$. In order to formulate the lemma, we define the following agreement property.

Definition 4.4. Let $\mathbb{F}$ be a finite field, let $m \in \mathbb{N}$, and let $\mathcal{T} \subseteq \mathbb{F}^m$. We say that $\mathcal{T}$ satisfies the h-agreement property if every two vectors in $\mathcal{T}$ agree on at least h coordinates.

What is the largest set of codewords of $\mathcal{C}$ that has the h-agreement property? If $h \leq d+1$, then there is an easy way to construct such a set: we fix the first h coordinates of a codeword arbitrarily, and take all the codewords that agree with this fixing. It is not hard to show that this construction yields as set of size $|\mathbb{F}|^{d+1-h}$. Our lemma says that this construction is optimal.

Lemma 4.5. *Let $h \leq d+1$, and let Let $\mathcal{T} \subseteq \mathcal{C}$ be a set with the h-agreement property. Then $|\mathcal{T}| \leq |\mathbb{F}|^{d+1-h}$.*

Proof. Suppose for the sake of contradiction that $|\mathcal{T}| > |\mathbb{F}|^{d+1-h}$. Let $\mathcal{R}$ be the Reed-Solomon code of degree $h-1$. Recall that $\mathcal{C}$ is a linear space of dimension $d+1$, and that $\mathcal{R}$ is a linear subspace of $\mathcal{C}$ of dimension h. Therefore, there are $d+1-h$ cosets of the form $\overline{c}+\mathcal{R}$ in $\mathcal{C}/\mathcal{R}$. By the pigeonhole principle, there are two distinct codewords $\overline{c}_1, \overline{c}_2 \in \mathcal{T}$ that belong to the same coset, i.e., $\overline{c}_1+\mathcal{R} = \overline{c}_2+\mathcal{R}$. This implies that $\overline{c}_1-\overline{c}_2 \in \mathcal{R}$, so $\overline{c}_1-\overline{c}_2$ contains at most $h-1$ zeroes. On the other hand, since $\overline{c}_1, \overline{c}_2 \in \mathcal{T}$, it holds that $\overline{c}_1$ and $\overline{c}_2$ agree on at least h coordinates, and therefore $\overline{c}_1 - \overline{c}_2$ contains at least h zeroes. We have reached a contradiction, and therefore $|\mathcal{T}| \leq |\mathbb{F}|^{d+1-t}$. ■

Lemma 4.5 can be viewed as a type of an Erdős-Ko-Rado theorem for sets of Reed-Solomon codewords. A similar lemma for sets of arbitrary tuples was proved by [FT99], and our proof generalizes a proof of [GMWW17] for sets of arbitrary vectors.

4.2 Proof of Lemma 4.2

In this section we prove Lemma 4.2, that is, we would like to prove that the number of pairs (π, x) where π supports x is at most $\mathsf{L}(\Pi) \cdot |\mathbb{F}|^d$. To this end, we prove that every rejecting transcript π can support at most $|\mathbb{F}|^d$ seeds x. Fix a rejecting transcript π, and let $\mathcal{T}$ be the set of matrices that correspond to seeds that are supported by x. We prove that $|\mathcal{T}| \leq |\mathbb{F}|^d$. Intuitively, the reason for this upper bound is that since π is a rejecting transcript, Alice and Bob must be convinced that their matrices agree on at least one row and by Lemma 4.5, this requires them to communicate $\log|\mathbb{F}|$ bits. This intuition is formalized as follows.

Claim 4.6 ([GMWW17])**.** *Every two matrices in $\mathcal{T}$ agree on at least one row.*

Proof. We use a standard "fooling set" argument. Let $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ denote the rectangle that corresponds to π. Suppose, for the sake of contradiction, that there exist two seeds x, y whose corresponding matrices X, Y do not agree on any row. By the definition of $\mathcal{T}$, it follows that there exist $a \in f^{-1}(0)$ and $b \in f^{-1}(1)$ such that $(x,a) \in \mathcal{X}_\pi$ and $(y,b) \in \mathcal{Y}_\pi$. In particular, this means that if we give to Alice and Bob the inputs (x,a) and (y,b), respectively, the protocol will transmit the transcript π.

However, this is a contradiction: on the one hand, π is a rejecting transcript. On the other hand, the players are not allowed to output $\perp$ on inputs (x,a), (y,b), since X and Y differ on all their rows, and in particular differ on the all the rows i for which $a_i \neq b_i$. The claim follows. ■

Claim 4.6 says that $\mathcal{T}$ satisfies the 1-agreement property, and therefore Lemma 4.5 implies that $|\mathcal{T}| \leq |\mathbb{F}|^d$, as required.

4.3 Proof of Lemma 4.3

In this section we prove Lemma 4.3, that is, we would like to prove that the number of pairs (π, x) where π supports x is at least $2^{-5\cdot(1+\frac{m}{n})\cdot\log m} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f)$. To this end, we partition the set of seeds x into "good seeds" and "bad seeds", where good seeds are seeds on which the protocol almost solves KW_f. We will show that every good seed x contributes almost $\mathsf{L}(f)$ pairs (π, x), and that the number of bad seeds of very small, which implies that the number of good seeds is almost $|\mathbb{F}|^{d+1}$. Together, the two bounds will imply the lemma. In order to define the notion of "the protocol almost solves KW_f", we use the following notation, that was also used in Section 3.

Definition 4.7. Let $x \in \{0,1\}^\ell$ be a seed. The protocol Π_x is the protocol that is obtained from Π by fixing the seed of both players to be x. More specifically, the protocol Π_x is the protocol that is obtained from Π be replacing, for each partial transcript π, the rectangle $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ with the rectangle $\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}$ defined by

$$\mathcal{A}_{\pi,x} \stackrel{\text{def}}{=} \left\{ a \in f^{-1}(0) \middle| (x,a) \in \mathcal{X}_\pi \right\}$$
$$\mathcal{B}_{\pi,x} \stackrel{\text{def}}{=} \left\{ b \in f^{-1}(1) \middle| (x,b) \in \mathcal{X}_\pi \right\}.$$

We now define the notion of good and bad seeds as follows.

Definition 4.8. Let $t \stackrel{\text{def}}{=} \left\lceil \frac{3\cdot m}{n} \right\rceil + 1$. A seed $x \in \{0,1\}^\ell$ is called *good* if Π_x is a protocol that solves the relaxed KW problem $KW_f(t)$ (see Section 2.3.1). Otherwise, we say that x is *bad*.

The following proposition says that good seeds contribute almost $\mathsf{L}(f)$ pairs (π, x), and follows immediately from the fact that $KW_f(t)$ is not much easier than the KW relation KW_f.

Proposition 4.9. *For every good seed x there are at least $2^{-t\cdot(\log m+2)} \cdot \mathsf{L}(f)$ pairs (π, x).*

Proof. Let x be a good seed. The protocol Π_x solves $KW_f(t)$, and therefore by Proposition 2.15 has at least

$$\mathsf{L}(KW_f(t)) \geq 2^{-t\cdot(\log n+2)} \cdot \mathsf{L}(KW_f) = 2^{-t\cdot(\log n+2)} \cdot \mathsf{L}(f)$$

distinct full transcripts. Each of these transcripts participates in a pair with x, so the result follows. ■

The following proposition, which is proved in Section 4.3.1 below, says that the number of bad seeds is very small.

Proposition 4.10. *The number of bad seeds is at most $2^{-m} \cdot |\mathbb{F}|^{d+1}$. Thus, the number of good seeds is at least $(1 - 2^{-m}) \cdot |\mathbb{F}|^{d+1}$.*

By combining Propositions 4.9 and 4.10, we get that the number of pairs is at least

$$\begin{aligned}
& (1 - 2^{-m}) \cdot |\mathbb{F}|^{d+1} \cdot 2^{-t\cdot(\log m+2)} \cdot \mathsf{L}(f) \\
\geq\ & 2^{-t\cdot(\log m+2)-1} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f) \\
=\ & 2^{-(\left\lceil \frac{6\cdot m}{n} \right\rceil+1)\cdot(\log m+2)-1} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f) \\
\geq\ & 2^{-8\cdot(1+\frac{m}{n})\cdot\log m} \cdot |\mathbb{F}|^{d+1} \cdot \mathsf{L}(f),
\end{aligned}$$

as required.

4.3.1 Proof of Proposition 4.10

The intuition for the proof is the following: recall that Alice and Bob reject, and this means that they have to be convinced that their matrices agree on some row i for which $a_i \neq b_i$. However, when x is a bad seed, Alice and Bob do not know an index i such that $a_i \neq b_i$ at the end of the protocol. This means that they have to be convinced that they agree on many rows, as otherwise they run the risk of rejecting a legal pair of inputs. But verifying that they agree on many rows is very costly, so they can only afford it for few seeds. Details follow.

First, recall that a seed x is bad if and only if Π_x does not solve the relaxed KW problem $KW_f(t)$. This implies that there exists some transcript π of Π_x, which is labeled with a rectangle $\mathcal{A}_{\pi,x} \times \mathcal{B}_{\pi,x}$, and a string $a \in \mathcal{A}_{\pi,x}$, such that the following holds:

- For every $\mathcal{I} \subseteq [m]$ such that $|\mathcal{I}| < t$, there exists $b \in \mathcal{B}_{\pi,x}$ such that $a|_{\mathcal{I}} = b|_{\mathcal{I}}$.

Viewing π as a transcript of the original protocol Π, it follows that there is a string $a \in f^{-1}(0)$, such that the following hold:

- $(x, a) \in \mathcal{X}_\pi$.
- For every $\mathcal{I} \subseteq [m]$ such that $|\mathcal{I}| < t$, there exists $b \in f^{-1}(1)$ such that $a|_{\mathcal{I}} = b|_{\mathcal{I}}$ and $(x, b) \in \mathcal{Y}_\pi$.

Now, without loss of generality, we may assume that

$$\mathsf{L}(\Pi) \leq \mathsf{L}(f) \cdot 2^n \leq 2^{m+n},$$

since otherwise Theorem 1.8 would follow immediately. Therefore, it suffices to prove that every transcript π and string a are "responsible" for at most $2^{-(3 \cdot m+n)} \cdot |\mathbb{F}|^{d+1}$ bad seeds. This would imply that there are at most $2^{-m} \cdot |\mathbb{F}|^{d+1}$ bad d+1, by summing over all transcripts of Π (at most 2^{m+n}) and all strings a (at most 2^m).

To this end, fix a transcript π of Π and a string $a \in f^{-1}(0)$. Let $\mathcal{T} \subseteq \mathcal{C}$ be the set of matrices corresponding to bad seeds that are supported by π and a. We prove that $|\mathcal{T}| \leq 2^{-(3 \cdot m+n)} \cdot |\mathbb{F}|^{d+1}$. The key idea is that since Alice does not know any small set $\mathcal{I}$ such that $a|_{\mathcal{I}} \neq b|_{\mathcal{I}}$, Alice and Bob must be convinced that their matrices agree on at least t rows. This intuition is made rigorous in the following statement.

Claim 4.11 ([GMWW17])**.** *The set $\mathcal{T}$ satisfies the t-agreement property.*

Proof. We need to show that every two matrices $X, Y \in \mathcal{T}$ agree on at least t rows. Let $X, Y \in \mathcal{T}$, let x, y be the corresponding seeds, and and let $\mathcal{I}$ be the set of rows on which X and Y agree. By definition of $\mathcal{T}$, it holds that $(x, a), (y, a) \in \mathcal{X}_\pi$. Suppose that $|\mathcal{I}| < t$. Then, by the assumption on π and a, there exists $b \in f^{-1}(1)$ such that $(y, b) \in \mathcal{Y}_\pi$ and $a|_{\mathcal{I}} = b|_{\mathcal{I}}$.

Next, observe that if we give the input (x, a) to Alice and the input (y, b) to Bob, the protocol will reach the transcript π. Now, π is a rejecting transcript, and therefore there must exist some index $i \in [m]$ such that $a_i \neq b_i$ but $X_i = Y_i$. However, we know that $a|_{\mathcal{I}} = b|_{\mathcal{I}}$, and therefore $i \notin \mathcal{I}$. It follows that X and Y agree on a row outside $\mathcal{I}$, thus contradicting the definition of $\mathcal{I}$. ■

Finally, by combining Claim 4.11 and Lemma 4.5, we conclude that $|\mathcal{T}| \leq |\mathbb{F}|^{d+1-t}$. Wrapping up, it follows that

$$\begin{aligned} |\mathcal{T}| &\leq |\mathbb{F}|^{d+1-t} \\ &= 2^{-t \cdot n} \cdot |\mathbb{F}|^{d+1} \\ &\leq 2^{-(\frac{3 \cdot m}{n}+1) \cdot n} \cdot |\mathbb{F}|^{d+1} \\ &= \frac{1}{2^{3 \cdot m+n}} \cdot |\mathbb{F}|^{d+1}, \end{aligned}$$

as required.

Acknowledgement. We are grateful to Ronen Shaltiel for explaining to us his paper on derandomized parallel repetition [Sha10] which served as an inspiration to this work, for pointers to the extractors' literature, and for numerous valuable discussions.

References

[CKLM17] Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudorandom properties. *CoRR*, abs/1704.06807, 2017.

[DHSV15] Irit Dinur, Prahladh Harsha, Srikanth Srinivasan, and Girish Varma. Derandomized graph product results using the low degree long code. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 275–287, 2015.

[DM16] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 3:1–3:51, 2016.

[EIRS01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiri Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001.

[FT99] Peter Frankl and Norihide Tokushige. The Erdős-Ko-Rado theorem for integer sequences. *Combinatorica*, 19(1):55–63, 1999.

[GKPW17] Mika Göös, Pritish Kamath, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for pˆnp. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 12:1–12:16, 2017.

[GLM+16] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016.

[GMWW17] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: The composition of a function and a universal relation. *SIAM J. Comput.*, 46(1):114–131, 2017.

[Göö15] Mika Göös. Lower bounds for clique vs. independent set. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1066–1076, 2015.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1077–1088, 2015.

[GS91] Michelangelo Grigni and Michael Sipser. Monotone separation of Logspace from NC. In *Structure in Complexity Theory Conference*, pages 294–298, 1991.

[GSS13] Justin Gilmer, Michael E. Saks, and Srikanth Srinivasan. Composition limits and separating examples for some boolean function complexity measures. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 185–196, 2013.

[HW93] Johan Håstad and Avi Wigderson. Composition of the universal relation. In *Advances in computational complexity theory, AMS-DIMACS*, 1993.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 538–545, 1995.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229, 1997.

[KKN95] Mauricio Karchmer, Eyal Kushilevitz, and Noam Nisan. Fractional covers and communication complexity. *SIAM J. Discrete Math.*, 8(1):76–92, 1995.

[KM18] Sajin Koroth and Or Meir. Improved composition theorems for functions and relations. In *RANDOM*, 2018.

[KMR17] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 590–603, 2017.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995.

[KT16] Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chicago J. Theor. Comput. Sci.*, 2016, 2016.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990.

[LRS15] James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 567–576, 2015.

[MOS04] Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning functions of k relevant variables. *J. Comput. Syst. Sci.*, 69(3):421–434, 2004.

[MS78] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*. North-holland Publishing Company, 2nd edition, 1978.

[NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

[Raz90] Alexander A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990.

[Raz16] Alexander A. Razborov. A new kind of tradeoffs in propositional proof complexity. *J. ACM*, 63(2):16:1–16:14, 2016.

[RM97] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 234–243, 1997.

[RRV02] Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002.

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992.

[Sha10] Ronen Shaltiel. Derandomized parallel repetition theorems for free games. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010*, pages 28–37, 2010.

[She11] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011.

[SZ09] Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Information & Computation*, 9(5):444–460, 2009.

[Tal13] Avishay Tal. Properties and applications of boolean function composition. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 441–454, 2013.

[TZ97] Gábor Tardos and Uri Zwick. The communication complexity of the universal relation. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 247–259, 1997.

[WYY17] Xiaodi Wu, Penghui Yao, and Henry S. Yuen. Raz-mckenzie simulation with the inner product gadget. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:10, 2017.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213, 1979.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Struct. Algorithms*, 11(4):345–367, 1997.

A Construction of Samplers

In this appendix, we prove Theorem 2.24, restated next.

Theorem 2.24. *There exists a universal constant q such that the following holds: there is an algorithm that when given as input ℓ, m, n, ε such that $\log n \geq q \cdot \log^3\left(\frac{\log m}{\varepsilon}\right)$, runs in time $\mathrm{poly}(\ell, m, n, 1/\varepsilon)$ and outputs an (α, ε)-averaging sampler $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ of $[\ell]$ where the sets are of size n and where*

$$\alpha = \frac{q}{\varepsilon^2} \cdot \frac{\ell}{m \cdot n}.$$

In order to prove the theorem, we use randomness extractors. An (seeded) extractor is an algorithm that transforms a distribution with high min-entropy into one that is close to the uniform distribution using a few additional uniformly distributed bits. It is defined as follows.

Definition A.1. A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)-*extractor* if for every random variable $\mathbf{x} \in \{0,1\}^n$ such that $H_\infty(\mathbf{x}) \geq k$ it holds that $E(\mathbf{x}, \boldsymbol{y})$ is ε-close to the uniform distribution, where $\boldsymbol{y}$ is a random variable that is uniformly distributed over $\{0,1\}^d$ and is independent of $\mathbf{x}$ (the input $\mathbf{x}$ is called the *source* and the input $\boldsymbol{y}$ is called the *seed*).
We say that E is a *strong* extractor if, when given input (x, y), its output begins with the string y.
We say that E is *explicit* if it can be computed in polynomial time.

Zuckerman *[Zuc97]* observed that every extractor yields an averaging sampler:

Lemma A.2 (*[Zuc97]*)**.** *Let $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a strong (k, ε)-strong extractor. We identify $\{0,1\}^m$ with $[2^m]$, and consider the following family $\mathcal{F}$ of subsets of $[2^m]$: for every string $x \in \{0,1\}^n$, we have a subset in the family that consists of all the possible outputs of $E(x, \cdot)$. Then, $\mathcal{F}$ is a $(2^{-(n-k)+1}, \varepsilon)$-averaging sampler.*

Remark A.3. In Lemma A.2, we require E to be *strong* only in order to guarantee that all the possible outputs of $E(x, \cdot)$ are *distinct*. Without this requirement, the family $\mathcal{F}$ is still a sampler when viewed as a family of *multisets*.

We use the following theorem due to [RRV02].

Theorem A.4 ([RRV02, Theorem 4])**.** *For every $n, k \in \mathbb{N}$ and $\varepsilon > 0$ such that $k \leq n$ there are (k, ε)-explicit strong extractors $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with*

$$\begin{aligned} d &= O\left(\log^2(\frac{n}{\varepsilon}) \cdot \log k\right) \\ m &= k + d - 2\log(1/\varepsilon) - O(1) \end{aligned} \tag{7}$$

We note that one can also construct extractors as in Theorem A.4 for larger values of d: If d is larger than is required by Theorem A.4, the extractor can output all the bits of the seed except for the last $O\left(\log^2(\frac{n}{\varepsilon}) \cdot \log k\right)$ bits, and then run the extractor of A.4 using the remaining bits of the seed.

We turn to prove Theorem 2.24. Given values ℓ, m, n as in the theorem, the algorithm takes the extractor of Theorem A.4, and turns into a sampler using Lemma A.2. The algorithm runs in polynomial time since the extractor is explicit. The parameters of the extractor are chosen as follows:

$$\begin{aligned} n &= \log m \\ d &= \log n \\ m &= \log \ell \\ \varepsilon &= \varepsilon, \end{aligned}$$

and the choice of k is derived from these choices using Equation 7. We choose the universal constant q of Theorem 2.24 such that $d = \log n$ is sufficiently large to for Theorem A.4 to hold, and such that the $O(1)$ term in the choice of m in Theorem A.4 is at most $\log q - 1$, so

$$k = \log \ell - \log n + 2\log(1/\varepsilon) + \log q + 1.$$

By Lemma A.2, the resulting sampler that our algorithm outputs is an (α, ε)-averaging sampler for

$$\begin{aligned}\alpha &= 2^{-(\log m - k)+1} \\ &= 2^{-\log m + \log \ell - \log n + 2\log(1/\varepsilon) + \log q} \\ &= \frac{q}{\varepsilon^2} \cdot \frac{\ell}{m \cdot n},\end{aligned}$$

as required.

Acknowledgement. We would like to thank anonymous referees for comments that improved the presentation of this work.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il