SIAM J. COMPUT.
Vol. 46, No. 1, pp. 114–131

TOWARD BETTER FORMULA LOWER BOUNDS: THE COMPOSITION OF A FUNCTION AND A UNIVERSAL RELATION*

DMITRY GAVINSKY†, OR MEIR‡, OMRI WEINSTEIN§, AND AVI WIGDERSON¶

Abstract. One of the major open problems in complexity theory is proving superlogarithmic lower bounds on the depth of circuits (i.e., $\mathbf{P} \not\subseteq \mathbf{NC}^1$). This problem is interesting for two reasons: first, it is tightly related to understanding the power of parallel computation and of small-space computation; second, it is one of the first milestones toward proving superpolynomial circuit lower bounds. Karchmer, Raz, and Wigderson [*Comput. Complexity*, 5 (1995), pp. 191–204] suggested approaching this problem by proving the following conjecture: given two Boolean functions f and g, the depth complexity of the composed function $g \diamond f$ is roughly the sum of the depth complexities of f and g. They showed that the validity of this conjecture would imply that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. As a starting point for studying the composition of functions, they introduced a relation called "the universal relation" and suggested studying the composition of universal relations. This suggestion proved fruitful, and an analogue of the Karchmer–Raz–Wigderson (KRW) conjecture for the universal relation was proved by Edmonds et al. [*Comput. Complexity*, 10 (2001), pp. 210–246]. An alternative proof was given later by Håstad and Wigderson [in *Advances in Computational Complexity Theory*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 13, AMS, Providence, RI, 1993, pp. 119–134]. However, studying the composition of functions seems more difficult, and the KRW conjecture is still an open question. In this work, we make a natural step in this direction, which lies between what is known and the original conjecture: we show that an analogue of the conjecture holds for the composition of a function with a universal relation.

Key words. formula, Karchmer–Wigderson relations, lower bounds, information complexity, communication complexity, KRW conjecture

AMS subject classification. 68Q17

DOI. 10.1137/15M1018319

1. Introduction.

One of the holy grails of complexity theory is showing that **NP** cannot be computed by polynomial-size circuits, namely, that $\mathbf{NP} \not\subseteq \mathbf{P}/\operatorname{poly}$. Unfortunately, it currently seems that even finding a function in **NP** that cannot be computed by circuits of linear size is beyond our reach. Thus, it makes sense to try to prove lower bounds against weaker models of computation, in the hope that such a study would eventually lead to lower bounds against general circuits.

This paper focuses on (de Morgan) formulas, which are one such weaker model. Intuitively, formulas model computations that cannot store intermediate results. For-

*Received by the editors April 24, 2015; accepted for publication (in revised form) October 10, 2016; published electronically February 16, 2017. A preliminary version of this work was published in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, 2014. An extended version of this work is available as ECCC TR13-190 [GMWW13].

http://www.siam.org/journals/sicomp/46-1/M101831.html

Funding: The first author was partially supported by grant P202/12/G061 of GA ČR and by RVO: 67985840. Part of this work was done while the first author was visiting the CQT at the National University of Singapore, and was partially funded by the Singapore Ministry of Education and the NRF. The second and fourth authors were partially supported by NSF grant CCF-1412958. The third author was supported by a Simons Society junior fellowship.

†Institute of Mathematics, Academy of Sciences, Žitna 25, 11567 Praha 1, Czech Republic (dmitry.gavinsky@gmail.com).

‡Department of Computer Science, University of Haifa, Haifa 31905, Israel (ormeir@cs.haifa.ac.il).

§Department of Computer Science, Columbia University, New York, NY 10027 (omri@cs.columbia.edu).

¶Institute for Advanced Study, Princeton, NJ 08540 (avi@ias.edu).

mally, they are defined as circuits with AND, OR, and NOT gates that have fan-out 1, or, in other words, their underlying graph is a tree.

For our purposes, it is useful to note that formulas are polynomially related to circuits[1] of depth $O(\log n)$: it is easy to show that circuits of depth $O(\log n)$ can be converted into formulas of polynomially related size. On the other hand, every formula of size s can be converted into a formula of depth $O(\log s)$ and size $\mathrm{poly}(s)$ [Spi71, Bre74, BB94]. In particular, the complexity class[2] $\mathbf{NC}^1$ can be defined both as the class of polynomial-size formulas and as the class of polynomial-size circuits of depth $O(\log n)$.

It is a major open problem to find an explicit function that requires formulas of superpolynomial size, that is, to prove that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. In fact, even proving that[2] $\mathbf{NEXP} \not\subseteq \mathbf{NC}^1$ would be a big breakthrough. The state-of-the-art in this direction is the work of Håstad [Hås98], which proved a lower bound of $\tilde{\Omega}(n^3)$ on the formula complexity of an explicit function due to Andreev [And87] (building on earlier work by [Sub61, And87, IN93, PZ93]). Improving over this lower bound is an important challenge.

One strategy for separating $\mathbf{P}$ from $\mathbf{NC}^1$ was suggested by Karchmer, Raz, and Wigderson [KRW95]. They made a conjecture on the depth complexity of composition and showed that this conjecture implies that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. In order to introduce their conjecture, we need some notation.

DEFINITION 1.1 (composition). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be Boolean functions. Their* composition $g \diamond f : (\{0,1\}^n)^m \to \{0,1\}$ *is defined by*

$$(g \diamond f)(x_1, \ldots, x_m) \stackrel{\text{def}}{=} g(f(x_1), \ldots, f(x_m)),$$

where $x_1, \ldots, x_m \in \{0,1\}^n$.

DEFINITION 1.2 (depth complexity). *Let $f : \{0,1\}^n \to \{0,1\}$. The depth complexity of f, denoted $\mathsf{D}(f)$, is the smallest depth of a circuit of fan-in 2 that computes f using AND, OR, and NOT gates.*

CONJECTURE 1.3 (the Karchmer–Raz–Wigderson (KRW) conjecture [KRW95]). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be nonconstant functions. Then*[3]

$$\mathsf{D}(g \diamond f) \approx \mathsf{D}(g) + \mathsf{D}(f). \tag{1.1}$$

As noted above, [KRW95] showed that this conjecture could be used to prove that $\mathbf{P} \not\subseteq \mathbf{NC}^1$: the basic idea is that one could apply $O(\log n)$ compositions of a random function $f : \{0,1\}^{\log n} \to \{0,1\}$, thus obtaining a new function over n bits that is computable in polynomial time yet requires depth $\tilde{\Omega}(\log^2 n)$. The key point here is that a random function on $\log n$ bits has depth complexity $\log n - o(\log n)$, and can be described explicitly using n bits. An interesting feature of this argument is that it does not seem to fall[4] into the framework of "natural proofs" of [RR97].

[1] All the circuits in this paper are assumed to have constant fan-in.

[2] In this paper, $\mathbf{NC}^1$ always denotes the *nonuniform* version of $\mathbf{NC}^1$, which is sometimes denoted $\mathbf{NC}^1/\mathrm{poly}$.

[3] The meaning of "approximate equality" in (1.1) is left vague, since there are a few variations that could be useful, some of which are considerably weaker than strict equality. In particular, proving either of the following lower bounds would imply that $\mathbf{P} \not\subseteq \mathbf{NC}^1$: (i) $\mathsf{D}(g \diamond f) \geq \varepsilon \cdot \mathsf{D}(g) + \mathsf{D}(f)$, or (ii) $\mathsf{D}(g \diamond f) \geq \mathsf{D}(g) + \varepsilon \cdot \mathsf{D}(f)$. It is also sufficient to prove the first inequality for a *random* g, or the second inequality for a *random* f.

[4] More specifically, it seems that this argument violates the largeness property because it only proves a lower bound for a specific, artificially constructed function, rather than for a random function.

In this paper, we make a natural step toward proving the KRW conjecture. The rest of this introduction is organized as follows: in section 1.1, we review the background relevant to our results. In section 1.2, we describe our main result and our techniques.

1.1. Background.

1.1.1. Karchmer–Wigderson relations. Karchmer and Wigderson [KW90] observed an interesting connection between depth complexity and communication complexity: for every Boolean function f, there exists a corresponding communication problem R_f, such that the depth complexity of f is equal to the deterministic[5] communication complexity of R_f. The communication problem R_f is often called the *Karchmer–Wigderson relation* of f, and we will refer to it as a *KW relation* for short. In fact, a stronger statement is implicit[6] in [KW90].

FACT 1.4 (see [KW90]). *For every formula ϕ that computes f, there exists a deterministic protocol Π_ϕ for R_f, whose underlying tree is exactly the underlying tree of ϕ, and vice versa.*

A corollary of Fact 1.4 that is particularly useful for us is the following: the formula size of f is exactly the minimal number of distinct transcripts in every protocol that solves R_f.

The communication problem R_f is defined as follows: Alice gets as input $x \in f^{-1}(0)$, and Bob gets as 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 R_f is a relation rather than a function.

This connection between functions and KW relations allows us to study the formula and depth complexity of functions using techniques from communication complexity. In the past, this approach has proved very fruitful in the setting of *monotone* formulas [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 $g \diamond f$. Let us describe how the KW relation $R_{g\diamond f}$ looks. Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$. For every $m \times n$ matrix X, let us denote by $f(X)$ the vector in $\{0,1\}^m$ obtained by applying f to each row X_j of X. In the KW relation $R_{g\diamond f}$, Alice and Bob get as inputs $m \times n$ matrices X, Y, respectively, such that $f(X) \in g^{-1}(0)$ and $f(Y) \in g^{-1}(1)$, and their goal is to find an entry (j,i) such that $X_{j,i} \neq Y_{j,i}$.

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

$$\mathsf{C}(R_{g\diamond f}) \leq \mathsf{C}(R_g) + \mathsf{C}(R_f). \tag{1.2}$$

This upper bound is achieved by the following protocol: for every $j \in [m]$, let X_j denote the jth row of X, and the same for Y. Alice and Bob first use the optimal protocol of g on inputs $f(X)$ and $f(Y)$ and thus find an index $j \in [m]$ such that $f(X_j) \neq f(Y_j)$. Then, they use the optimal protocol of f on inputs X_j and Y_j to find

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

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

a coordinate i on which the jth rows differ, thus obtaining an entry (j, i) on which X and Y differ.

The KRW conjecture says that the above protocol is essentially optimal. One intuition for that conjecture is the following: the best way for Alice and Bob to solve $R_{g \diamond f}$ is to solve R_f on some row j such that $f(X_j) \neq f(Y_j)$, since otherwise they are not using the guarantee they have on X and Y. However, to do that, they must find such a row j, and to this end they have to solve R_g. Thus, they have to transmit $\mathsf{C}(R_g)$ bits in order to find j, and another $\mathsf{C}(R_f)$ bits to solve f on the jth row.

This intuition was made rigorous for the monotone version of the KRW conjecture in [KRW95]. In the monotone setting, they showed a reduction from the direct sum of R_f and R_g to $R_{g \diamond f}$, which means that any protocol that solves $R_{g \diamond f}$ must solve both R_f and R_g. They used this reduction to separate $\mathbf{NC}^1$ from $\mathbf{NC}^2$. A similar intuition underlies our argument, as well as the works [EIRS01, HW93] that are to be discussed later.

1.1.3. The universal relation and its composition. Since proving the KRW conjecture seems difficult, the authors of [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* R_{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 R_{U_n} is universal in the sense that every KW relation reduces to it, and indeed, it is not hard to prove that $\mathsf{C}(R_{\mathrm{U}_n}) \geq n$.

The composition of two universal relations R_{U_m} and R_{U_n}, denoted $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$, 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 $j \in [n]$ such that $a_j \neq b_j$, it holds that $X_j \neq Y_j$.

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 $f(X)$ and $f(Y)$ in the KW relation $R_{g \diamond f}$.

To see why $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$ is a good way to abstract the KRW conjecture, observe that $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$ is a universal version of composition problems $R_{g \diamond f}$, in the sense that every composition problem $R_{g \diamond f}$ reduces to $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$. Moreover, the protocol described above for $R_{g \diamond f}$ also works for $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$: Alice and Bob first apply the optimal protocol for R_{U_m} to a and b to find j, and then apply the optimal protocol for R_{U_n} to X_j and Y_j. Thus, a natural variant of the KRW conjecture for this protocol would be that this protocol is optimal for $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$. Following this reasoning, the authors of [KRW95] suggested proving that

$$\mathsf{C}(R_{\mathrm{U}_m \diamond \mathrm{U}_n}) \approx \mathsf{C}(R_{\mathrm{U}_m}) + \mathsf{C}(R_{\mathrm{U}_n}) \geq m + n \tag{1.3}$$

as a first step toward proving the KRW conjecture. This challenge was met[7] by [EIRS01] up to a small additive loss, and an alternative proof was given later in [HW93]. Since then, there has been no further progress on the KRW conjecture for about two decades.

[7] In fact, they only consider the case where $m = n$, but their argument should generalize to the case where $m \neq n$.

1.2. Our main result: The composition of a function with the universal relation. Summing up, the KRW conjecture is about the composition of two functions $R_{g \diamond f}$, but it was only known how to prove it for the composition of two universal relations $R_{U_m \diamond U_n}$. In this work we go a step further: we prove an analogue of the KRW conjecture for relations of the form $R_{g \diamond \mathrm{U}_n}$, where $g \in \{0,1\}^m \to \{0,1\}$ is an arbitrary function, and where $R_{g \diamond \mathrm{U}_n}$ is a problem that can be naturally viewed as the composition of g with the universal relation.

We define the communication problem $R_{g \diamond \mathrm{U}_n}$ as follows. Alice gets as input an $m \times n$ matrix X and a string $a \in g^{-1}(0)$, and Bob gets as input an $m \times n$ matrix Y and a string $b \in g^{-1}(1)$. Their inputs are guaranteed to satisfy condition 2 of $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$, i.e., for every $j \in [n]$ such that $a_j \neq b_j$, it holds that $X_j \neq Y_j$. Clearly, their inputs also satisfy $a \neq b$, as in condition 1 of $R_{\mathrm{U}_m \diamond \mathrm{U}_n}$. The goal of Alice and Bob, as usual, is to find an entry on which X and Y differ.

Note that $R_{g \diamond U_n}$ is universal, in the sense that for any $f : \{0,1\}^n \to \{0,1\}$, the communication problem $R_{g \diamond f}$ reduces to $R_{g \diamond \mathrm{U}_n}$. An analogue of the KRW conjecture for $R_{g \diamond U_n}$ would be

$$\mathsf{C}(R_{g \diamond U_n}) \approx \mathsf{C}(R_g) + \mathsf{C}(R_{\mathrm{U}_n}) \geq \mathsf{C}(R_g) + n. \tag{1.4}$$

We prove the following closely related result.

THEOREM 1.5. *Let $m, n \in \mathbb{N}$, and let $g : \{0,1\}^m \to \{0,1\}$ be a nonconstant function. Then,*

$$\mathsf{C}(R_{g \diamond \mathrm{U}_n}) \geq \Omega\left(\mathsf{C}(R_g)\right) + n - O\Big(1 + \frac{m}{n}\Big) \cdot \log m.$$

In fact, we obtain Theorem 1.5 as a corollary of the following theorem, which gives a tighter bound in terms of formula complexity. Let $\mathsf{L}(g)$ denote the formula complexity of g, and recall that $\log \mathsf{L}(g) \geq \Omega\left(\mathsf{C}(R_g)\right)$ due to the correspondence between formula size and circuit depth. We have the following result.

THEOREM 1.6 (main theorem). *Let $m, n \in \mathbb{N}$, and let $g : \{0,1\}^m \to \{0,1\}$ be a nonconstant function. Then,*

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

Moreover, the same lower bound applies to the logarithm of the number of leaves of any protocol for $R_{g \diamond \mathrm{U}_n}$ (which is the "formula complexity" of $R_{g \diamond \mathrm{U}_n}$).

There is a good reason why the formula complexity $\mathsf{L}(g)$ appears in Theorem 1.6, as will be made clear in the following discussion on our techniques.

Remark 1.7. In the target application of the KRW conjecture, namely the proof that $\mathbf{P} \not\subseteq \mathbf{NC}^1$, the parameters can be chosen such that $m \ll n$, so the loss of $O(1 + \frac{m}{n}) \cdot \log m$ in Theorem 1.6 is not very important.

Remark 1.8. We note that Theorem 1.6 also implies a lower bound on the composition $R_{U_m \diamond U_n}$ of two universal relations, thus giving yet another proof for the results of [EIRS01, HW93]. In fact, our techniques can be used to give a simpler proof for those results.

Remark 1.9. The key obstacle that makes the KRW conjecture much harder to prove than the above results of [EIRS01, HW93] is the following: in the universal relation, Alice and Bob are "symmetric"; that is, their sets of legal inputs are the same.

This property makes it much easier to prove lower bounds for universal relations, and it was instrumental for the results of [EIRS01, HW93]. On the other hand, KW relations do not lend themselves to this property, which makes them more difficult to analyze. The latter can be viewed as an artifact of the following conceptual difference between the universal relation and KW relations: while the KW relation R_f is a *total* relation (for any function f), the universal relation is not. This means, for example, that even the *nondeterministic* communication complexity of solving the universal relation is $\approx n$, while the nondeterministic communication complexity of solving R_f is only $\approx \log n$, suggesting that the latter problem might be much harder to analyze.

In our work, we get halfway to bypassing this obstacle: we get rid of the latter symmetry property in one part of the proof, but retain it in the other. More specifically, the works [EIRS01, HW93], which analyze compositions of the form $\mathrm{U}_m \diamond \mathrm{U}_n$, use the symmetry property both for analyzing the "U_m part" and for analyzing the "U_n part." We, on the other hand, replace U_m with a function g, and hence we manage to get rid of the use of the symmetry property in the analysis of U_m. However, we retain the use of the symmetry property in our analysis of U_n.

1.2.1. Our techniques. Our proof uses a combinatorial counting argument, which is inspired by ideas from the information-complexity literature. Our starting point is the observation that (the logarithm of) the size of a formula ϕ for any function f can be reinterpreted as the information that is transmitted by protocols that solve R_f.

To see why this is helpful, consider the KW relation $R_{g \diamond \mathrm{U}_n}$. Intuitively, we would like to argue that in order to solve $R_{g \diamond \mathrm{U}_n}$, Alice and Bob must solve R_g (incurring a cost of $\mathsf{C}(R_g)$) and also solve the universal relation on one of the rows of their matrices (incurring a cost of n). Such an argument requires decomposing the communication of Alice and Bob into communication "about" R_g and communication "about" R_{U_n}. However, it is not clear how to do that, because Alice and Bob may "talk" simultaneously about R_g and R_{U_n} (e.g., by sending the XOR of a bit of a and a bit of X).

On the other hand, when considering the *information* transmitted by Alice and Bob, such a decomposition comes up naturally: the information that Alice and Bob transmit can be decomposed, using the chain rule, into the information they transmit on the strings a, b (which are inputs of R_g) and the information they transmit on the matrices X and Y (which consist of inputs of R_{U_n}). We now derive the required lower bound

$$\mathsf{C}(R_{g \diamond \mathrm{U}_n}) \geq \log \mathsf{L}(g) + n - O\Big(1 + \frac{m}{n}\Big) \cdot \log m,$$

as follows: the information about a and b contributes $\log \mathsf{L}(g)$ (which is the information cost of R_g), and the information about X and Y contributes n (which is the information cost of R_{U_n}). Of course, implementing this argument is far from trivial, and in particular, we do not know how to extend this argument to the full KRW conjecture, i.e., KW relations of the form $R_{g \diamond f}$.

This is reminiscent of a similar phenomenon in the literature about the direct sum problem in communication complexity (e.g., [BBCR13]): the direct sum problem asks whether solving k independent instances of a function is k times harder than solving a single instance. The reason that information complexity is useful for studying this question is that there, too, the information transmitted by the protocol can be decomposed, using the chain rule, into the information about each of the independent instances.

This suggests that information complexity may be the "right" tool for studying the KRW conjecture. In particular, since in the setting of KW relations, the information cost is analogous to the formula size, the "correct" way to state the KRW conjecture may be using formula size:

$$\mathsf{L}(g \diamond f) \approx \mathsf{L}(g) \cdot \mathsf{L}(f).$$

Interestingly, the KRW conjecture is supported by the works [And87, Hås98], which prove that

$$\mathsf{L}(g \diamond \oplus_n) = \mathsf{L}(g) \cdot n^2 / \operatorname{poly}\log m = \mathsf{L}(g) \cdot \mathsf{L}(\oplus_n) / \operatorname{poly}\log(m),$$

where $\oplus_n$ is the parity function of n bits and g is an arbitrary function over m bits, and where the second equality follows from [Khr72].

In the extended version of the current paper [GMWW13], we develop a general framework for using information-complexity arguments to analyze KW relations, and we present the proof of our main result in this framework.

Organization of this paper. In section 2, we review the required preliminaries. Then, in section 3, we provide the proof of our main result.

2. Preliminaries. We reserve bold letters for random variables, and calligraphic letters for sets. We use $[n]$ to denote the set $\{1, \ldots, n\}$. For a function $f : \mathbb{N} \to \mathbb{N}$, we denote

$$\tilde{O}(f) \stackrel{\text{def}}{=} O(f \cdot \log^{O(1)} f),$$
$$\tilde{\Omega}(f) \stackrel{\text{def}}{=} \Omega(f / \log^{O(1)} f).$$

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 $X_j \in \{0,1\}^n$ the jth row of X. Throughout the paper, we denote by $\oplus_m$ the parity function over m bits.

2.1. Formulas.

DEFINITION 2.1. *A* 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 theorem establishes a tight connection between the formula complexity and the depth complexity of a function.

THEOREM 2.3 (see [BB94], following along the lines of [Spi71, Bre74]). *For every* $\alpha > 1$ *the following holds: for every formula* ϕ *of size* s*, there exists an equivalent formula* ϕ' *of depth* $O(\log s)$ *and size* s^α*. The constant in the big-O notation depends on* α*.*

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* that computes f. However, for our purposes, this distinction does not matter, since every circuit of depth $O(\log n)$ can be transformed into a formula of the same depth and of polynomial size.

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 the 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 outgoing edges labeled with* 0 *and* 1*, respectively. Then, the following hold:*
  - *$\mathcal{X}_v = \mathcal{X}_{v_0} \cup \mathcal{X}_{v_1}$, 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 notes owned by Bob, while changing the roles of $\mathcal{X}$ and $\mathcal{Y}$.*
- *For each 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 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.*

We will sometimes invoke a protocol Π on inputs that are random variables $\mathbf{x}, \mathbf{y}$. In such a case, the transcript is a random variable as well. With some abuse of

notation, we will use $\Pi \stackrel{\text{def}}{=} \Pi(\mathbf{x}, \mathbf{y})$ to denote this random transcript.

2.3. KW relations. In the remainder of this section, we provide some background, including the formal definitions and extensions which are required to state and understand our main result. The familiar reader may want to skip the overview below and jump to the actual proof in section 3.

We start by defining KW relations formally, and give a sketch of the correspondence between KW relations and formulas. In addition, in section 2.3.1, we introduce a useful generalization of KW relations, which we call "relaxed KW problems."

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

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

Intuitively, $R_{\mathcal{X},\mathcal{Y}}$ 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 nonconstant function. The* KW relation of f*, denoted R_f, is defined by $R_f \stackrel{\text{def}}{=} R_{f^{-1}(0),f^{-1}(1)}$.*

DEFINITION 2.11. *Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be two disjoint sets. We say that a formula ϕ* separates *$\mathcal{X}$ and $\mathcal{Y}$ if $\phi(\mathcal{X}) = 0$ and $\phi(\mathcal{Y}) = 1$.*

THEOREM 2.12 (implicit in [KW90]). *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 $R_{\mathcal{X},\mathcal{Y}}$, whose underlying tree is the same as the underlying tree of ϕ. In the other direction, for every protocol Π that solves $R_{\mathcal{X},\mathcal{Y}}$, 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.*

Proof. For the first direction, let ϕ be a formula that separates $\mathcal{X}$ and $\mathcal{Y}$. We construct Π_ϕ by induction: if ϕ is of size 1, then ϕ is a single literal of the form x_i or $\neg x_i$. This implies that all the strings in $\mathcal{X}$ differ from all the strings in $\mathcal{Y}$ on the coordinate i. Therefore, we define Π_ϕ as the protocol in which the players do not interact, and always output i. Note that the protocol tree Π_ϕ indeed has the same structure as the tree of ϕ.

Next, assume that $\phi = \phi_0 \wedge \phi_1$ (if $\phi = \phi_0 \vee \phi_1$, the construction is analogous). Let us denote by $\mathcal{X}_0$ and $\mathcal{X}_1$ the sets of strings x such that $\phi_0(x) = 0$ and $\phi_1(x) = 0$, respectively, and observe that $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$. Moreover, observe that $\phi_0(\mathcal{Y}) = \phi_1(\mathcal{Y}) = 1$. We now define Π_ϕ as follows: Alice sends Bob a bit b such that her input belongs to $\mathcal{X}_b$, and then they execute the protocol Π_{ϕ_b}. It is easy to see that Π_ϕ indeed solves $R_{\mathcal{X},\mathcal{Y}}$, and that the protocol tree of Π_ϕ has the same structure as the tree of ϕ. This concludes the first direction.

For the second direction, let Π be a protocol that solves $R_{\mathcal{X},\mathcal{Y}}$. Again, we construct ϕ_Π by induction: if Π is of size 1, then it consists of a single leaf that is labeled with some coordinate i. This implies that all the strings in $\mathcal{X}$ differ from all the strings in $\mathcal{Y}$ on the coordinate i. If for all $x \in \mathcal{X}$ it holds that $x_i = 0$, we define ϕ_Π to be the literal x_i, and otherwise we define it to be the literal $\neg x_i$. Note that the tree of ϕ_Π indeed has the same structure as the tree of Π.

Next, assume that Alice speaks first at Π (if Bob speaks first, the construction is analogous). Let us denote by $\mathcal{X}_0$ and $\mathcal{X}_1$ the sets of strings x on which Alice sends the bits 0 and 1, respectively, as her first message. Let Π_0 and Π_1 be the

residual protocols obtained from Π by conditioning on Alice's message, and note that by induction there exist formulas ϕ_{Π_0} and ϕ_{Π_1} such that ϕ_{Π_b} separates $\mathcal{X}_b$ and $\mathcal{Y}$. We now define $\phi_\Pi \stackrel{\text{def}}{=} \phi_{\Pi_0} \wedge \phi_{\Pi_1}$. It is easy to see that ϕ_Π indeed separates $\mathcal{X}$ and $\mathcal{Y}$, and to see that the tree of ϕ_Π has the same structure as the tree of Π. This concludes the second direction. □

COROLLARY 2.13. *Let $f : \{0,1\}^n \to \{0,1\}$. Then, for every formula ϕ for f, there exists a protocol Π_ϕ that solves R_f, whose underlying tree is the same as the underlying tree of ϕ. In the other direction, for every protocol Π that solves R_f there exists a formula ϕ_Π for f, whose underlying tree is the same as the underlying tree of Π.*

COROLLARY 2.14. *For every nonconstant $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathsf{D}(f) = \mathsf{C}(R_f)$, and $\mathsf{L}(f) = \mathsf{L}(R_f)$.*

2.3.1. Relaxed KW problems. In this section, we introduce the notion of "relaxed KW problems." Intuitively, these are KW relations that require only 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, 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* $R_f(t)$ 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:

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 the standard KW relation, we do not require that at the end of the protocol, both players know the set $\mathcal{I}$. Instead, we require only 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 up to this point. This leads us to the following definition of the relaxed KW problem.

DEFINITION 2.15. *Let $f : \{0,1\}^n \to \{0,1\}$ be a nonconstant 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 $R_f(t)$ *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 2.16. Note that in Definition 2.15, 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.

Remark 2.17. It is tempting to guess that $R_f(1)$ is the same as R_f, but it is not: in the communication problem R_f, Bob is required to know i at the end of the protocol, while in $R_f(1)$, he is not. In particular, if f is the AND function, then $\mathsf{C}(R_f) = \log n$, while $\mathsf{C}(R_f(1)) = 0$.

Remark 2.18. Definition 2.15 is inspired by the definition of *k-limit* in [HJP95, Definition 2.1].

We now prove the following easy proposition, which says that the relaxed KW problem $R_f(t)$ is not much easier than the original KW relation R_f.

PROPOSITION 2.19. *Let $f : \{0,1\}^n \to \{0,1\}$, and let $t \in \mathbb{N}$. Then,*

$$\mathsf{C}(R_f(t)) \geq \mathsf{C}(R_f) - t \cdot (\log n + 2),$$
$$\mathsf{L}(R_f(t)) \geq 2^{-t\cdot(\log n+2)} \cdot \mathsf{L}(R_f).$$

Proof. We prove the proposition by reducing R_f to $R_f(t)$. Let Π be a protocol for $R_f(t)$. We show that there exists a protocol Π' for R_f such that

$$\mathsf{C}(\Pi') \leq \mathsf{C}(R_f) + t \cdot (\log n + 2),$$
$$\mathsf{L}(R_f(t)) \leq 2^{t\cdot(\log n+2)} \cdot \mathsf{L}(R_f).$$

The protocol Π' for R_f is defined as follows: when Alice and Bob get inputs x and y, respectively, they invoke the protocol Π on their inputs, thus reaching a leaf ℓ. By Definition 2.15, there exists a set $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| < t$, such that $x|_{\mathcal{I}} \neq y'|_{\mathcal{I}}$ for every y' that is supported by ℓ. Alice now sends the set $\mathcal{I}$ and the string $x|_{\mathcal{I}}$ to Bob, and Bob replies with $y|_{\mathcal{I}}$. At this point, they both know a coordinate on which x and y differ, and the protocol ends.

The correctness of the protocol Π' is easy to verify. To analyze its communication complexity and size, observe that after reaching the leaf ℓ, Alice and Bob transmit at most

$$|\mathcal{I}| \cdot \log n + 2 \cdot |\mathcal{I}| < t \cdot (\log n + 2)$$

bits: $|\mathcal{I}| \cdot \log n$ bits for transmitting the set $\mathcal{I}$ itself, and another $2 \cdot |\mathcal{I}|$ bits for transmitting $a|_{\mathcal{I}}$ and $b|_{\mathcal{I}}$. This implies that the protocol tree of Π' can be obtained from the protocol tree of Π by replacing each leaf of Π with a binary tree that has at most $2^{t\cdot(\log n+2)}$ leaves and is of depth at most $t \cdot (\log n + 2)$. The required upper bounds on $\mathsf{C}(\Pi')$ and $\mathsf{L}(\Pi')$ follow. ∎

2.4. The universal relation and its compositions. In this section, we formally define the universal relation and its compositions. *We caution that the following definitions are slightly different from those given in the introduction:* for example, 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 modification was suggested by [HW93].

DEFINITION 2.20. *The* universal relation R_{U_n} *is defined as follows:*

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

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 output the special rejection symbol $\perp$ if $x = y$.

We use Definition 2.20 rather than the definition in the introduction because it is more convenient to work with. For example, using Definition 2.20, it is trivial to prove a lower bound on the communication complexity of this relation: the easiest way to see it is to note that the task of checking whether two strings are equal reduces to R_{U_n}, and the communication complexity of this task is well known to be at least n.

We note, however, that the difference between Definition 2.20 and the definition in the introduction does not substantially change the communication complexity of R_{U_n}.

To see this, suppose that there is a protocol Π that solves R_{U_n} under the promise that $x \neq y$. Then, there is a protocol Π' that solves R_{U_n} without this promise using two more bits: given inputs x and y which may be equal, the players invoke the protocol Π. Suppose Π outputs a coordinate i. Now, the players check whether $x_i \neq y_i$ by exchanging two more bits. If they find that $x_i = y_i$, they reject, and otherwise they output i.

2.5. A combinatorial lemma. In this section, we state and prove a combinatorial lemma that will be used in the proof of our main result. The motivation for this lemma comes from the following question in communication complexity, which will be encountered in the forthcoming sections: suppose Alice and Bob get as inputs $x, y \in \Sigma^m$ for some finite alphabet Σ. They would like to verify that their inputs agree on at least h coordinates. We wish to prove that Alice and Bob must transmit at least $h \cdot \log|\Sigma|$ bits.

This communication problem motivates the definition of the following property of sets of strings.

DEFINITION 2.21. *Let Σ be a finite alphabet, let $h, m \in \mathbb{N}$, and let $\mathcal{S} \subseteq \Sigma^m$. We say that $\mathcal{S}$ satisfies the* h-agreement property *if every two strings in $\mathcal{S}$ agree on at least h coordinates.*

Now, in order to prove the lower bound on the above communication problem, we need an upper bound on the size of sets that satisfy the h-agreement property.

The most straightforward way to construct a set that satisfies the h-agreement property is to fix a set of coordinates $\mathcal{I} \subseteq [m]$ of size h, and take all the strings whose restriction to $\mathcal{I}$ is some fixed string. A set $\mathcal{S}$ constructed this way will be of size $|\Sigma|^{m-h}$. The following theorem says that this is the optimal way of constructing such a set.

THEOREM 2.22 (see [FT99, Corollary 1]). *Let $\mathcal{S} \subseteq \Sigma^m$ be a set that satisfies the h-agreement property, and suppose that $|\Sigma| \geq h + 1$. Then $|\mathcal{S}| \leq |\Sigma|^{m-h}$.*

The proof of [FT99] is quite nontrivial. For completeness, we provide a simple proof of the following two weaker lemmas, which are sufficient to prove our main result. The following lemma and its proof are due to [ADFS04, Claim 4.1] (following along the lines of [GL74, Theorem 1]).

LEMMA 2.23. *Let Σ be a finite set of size at least m, and let $\mathcal{S} \subseteq \Sigma^m$ be a set that satisfies the* 1*-agreement property. Then $|\mathcal{S}| \leq |\Sigma|^{m-1}$.*

Proof. Let us view Σ as some additive group of size $|\Sigma|$, say $\mathbb{Z}_{|\Sigma|}$. Consider the following subgroup of Σ^m:

$$C = \left\{ (\underbrace{\sigma, \sigma, \ldots, \sigma}_{m}) \;\middle|\; \sigma \in \Sigma \right\}.$$

Observe that $|C| = |\Sigma|$, so the number of distinct cosets $x + C$ is $|\Sigma|^{m-1}$.

Now, assume for the sake of contradiction that $|\mathcal{S}| > |\Sigma|^{m-1}$. By the pigeonhole principle, there exist two distinct strings $x, y \in \mathcal{S}$ such that $x + C = y + C$. Equivalently, it holds that $x - y \in C$, that is,

$$x - y = (\underbrace{\sigma, \sigma, \ldots, \sigma}_{m})$$

for some nonzero $\sigma \in \Sigma$. But this means that x and y differ on all their coordinates, contradicting the assumption that $x, y \in \mathcal{S}$ (since $\mathcal{S}$ satisfies the 1-agreement property). ∎

The following lemma generalizes the above argument of [ADFS04] to h-agreement for any value of h, but holds only when the alphabet is a finite field of size at least m.

LEMMA 2.24. *Let $\mathbb{F}$ be a finite field, let $m \in \mathbb{N}$ be such that $m \leq |\mathbb{F}|$, and let $\mathcal{S} \subseteq \mathbb{F}^m$ be a set that satisfies the h-agreement property. Then $|\mathcal{S}| \leq |\mathbb{F}|^{m-h}$.*

Proof. We start with some notation. Let $H \subseteq \mathbb{F}$ be an arbitrary set of size m, and let us identify strings in $\mathbb{F}^m$ with functions $f : H \to \mathbb{F}$. Furthermore, let C be the set of such functions that are univariate polynomials of degree at most $h-1$. Observe that $|C| = |\mathbb{F}|^h$, so the number of distinct cosets $x + C$ is $|\mathbb{F}|^{m-h}$.

Now, assume for the sake of contradiction that $|\mathcal{S}| > |\mathbb{F}|^{m-h}$. By the pigeonhole principle, there exist two distinct strings $x, y \in \mathcal{S}$ such that $x + C = y + C$. Equivalently, it holds that $x - y \in C$, that is, $x - y$ is a nonzero univariate polynomial of degree at most $h - 1$. But, such a polynomial has at most $h - 1$ roots, and therefore x and y may agree on at most $h - 1$ coordinates, contradicting the assumption that $x, y \in \mathcal{S}$ (since $\mathcal{S}$ satisfies the h-agreement property). ∎

3. Our main result. In this section, we provide the proof of our main result. Let $g : \{0,1\}^m \to \{0,1\}$. We consider the relation $R_{g \diamond \mathrm{U}_n}$, which corresponds to the following communication problem: Alice gets as input a matrix $X \in \{0,1\}^{m \times n}$ and a string $a \in g^{-1}(0)$. Bob gets a matrix $Y \in \{0,1\}^{m \times n}$ and a vector $b \in g^{-1}(1)$. Their goal is to find an entry (j, i) on which X and Y differ, but they are allowed to reject if there exists an index $j \in [m]$ such that $a_j \neq b_j$ but $X_j = Y_j$. Formally, we have the following definition.

DEFINITION 3.1. *Let $g : \{0,1\}^m \to \{0,1\}$, and let $n \in \mathbb{N}$. The relation $R_{g \diamond \mathrm{U}_n}$ is defined by*

$$\begin{aligned} R_{g \diamond \mathrm{U}_n} \stackrel{\text{def}}{=} & \Big\{ ((X,a),(Y,b),(j,i)) : X, Y \in \{0,1\}^{m \times n},\ a \in g^{-1}(0),\ b \in g^{-1}(1),\ X_{j,i} \neq Y_{j,i} \Big\} \\ & \cup \Big\{ ((X,a),(Y,b),\perp) : X, Y \in \{0,1\}^{m \times n},\ a \in g^{-1}(0),\ b \in g^{-1}(1), \\ & \qquad \exists j :\ a_j \neq b_j,\ X_j = Y_j \Big\}. \end{aligned}$$

THEOREM 1.6 (main theorem). *Let $m, n \in \mathbb{N}$, and let $g : \{0,1\}^m \to \{0,1\}$ be a nonconstant function. Then,*

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

In the rest of this section, we prove Theorem 1.6. We note that only the second inequality requires a proof, whereas the first inequality is trivial since a binary tree of depth c has at most 2^c leaves. Let $m, n \in \mathbb{N}$, let $g : \{0,1\}^m \to \{0,1\}$, and let Π be a protocol for $R_{g \diamond \mathrm{U}_n}$. We would like to prove that Π has at least $\mathsf{L}(g) \cdot 2^{n - O(1 + \frac{m}{n}) \cdot \log m.}$ leaves.

The basic idea for the proof is the following. We lower-bound the number of leaves that output the rejection symbol $\perp$. For each such leaf ℓ, Alice and Bob must be convinced that there exists some $j \in [m]$ such that $a_j \neq b_j$, but $X_j = Y_j$. In particular, the following must hold:

1. They must be convinced that X and Y agree on at least one row. This is where we gain the factor of 2^n in the number of leaves.

2. Either they find an index $j \in [m]$ such that $a_j \neq b_j$, or they do not:
   (a) If they do find such a j, then they must solve R_g. This gains a factor of $\mathsf{L}(g)$ in the number of leaves.
   (b) If they do not find such a specific index j, then they must be convinced that X and Y agree on many rows. However, this forces them to reveal a lot of information about the matrices X and Y, and they cannot afford to do it for most matrices.

We turn to the formal proof. The following technical definition is useful.

DEFINITION 3.2 (supporting leaf). *Let ℓ be a leaf of Π, and let $\mathcal{X}_\ell \times \mathcal{Y}_\ell$ be its corresponding rectangle.*

- *We say that the leaf ℓ* supports *a matrix $X \in \{0,1\}^{m \times n}$ if X can be given as an input to both players at ℓ. Formally, ℓ supports X if there exist $a, b \in \{0,1\}^m$ such that $(X, a) \in \mathcal{X}_\ell$ and $(X, b) \in \mathcal{Y}_\ell$. We also say that X is* supported by ℓ and a, *or by ℓ* and b. *Note that the leaf ℓ must be a leaf that outputs $\perp$.*
- *We say that the leaf ℓ* supports *$a \in g^{-1}(0)$ if a can be given as input to Alice at ℓ. Formally, ℓ supports a if there exists a matrix $X \in \{0,1\}^{m \times n}$ such that $(X, a) \in \mathcal{X}_\ell$. A similar definition applies to strings $b \in g^{-1}(1)$.*

In order to prove a lower bound on $\mathsf{L}(\Pi)$, we double-count the number of pairs (ℓ, X), where ℓ is a leaf of Π that outputs $\perp$, and X is a matrix that is supported by L. Specifically, in the next two subsections, we prove the following lemmas, which together imply Theorem 1.6.

LEMMA 3.3. *The number of pairs (ℓ, X) is at most $\mathsf{L}(\Pi) \cdot 2^{(m-1)\cdot n}$.*

LEMMA 3.4. *The number of pairs (ℓ, X) is at least $2^{mn - O(1+\frac{m}{n})\cdot \log m} \cdot \mathsf{L}(g)$.*

3.1. Proof of Lemma 3.3. We would like to prove that the number of pairs (ℓ, X) is at most $\mathsf{L}(\Pi) \cdot 2^{(m-1)\cdot n}$. To this end, we prove that every leaf can support at most $2^{(m-1)\cdot n}$ matrices. Fix a leaf ℓ, and let $\mathcal{T}$ be the set of matrices supported by ℓ. We prove that $|\mathcal{T}| \leq 2^{(m-1)\cdot n}$.

Intuitively, the reason for this upper bound is that at ℓ, Alice and Bob must be convinced that their matrices agree on at least one row. This intuition is formalized as follows.

CLAIM 3.5. *Every two matrices X, X' in $\mathcal{T}$ agree on at least one row.*

Proof. We use a standard "fooling set" argument. Let $\mathcal{X}_\ell \times \mathcal{Y}_\ell$ denote the rectangle that corresponds to ℓ. Suppose, for the sake of contradiction, that there exist $X, X' \in \mathcal{T}$ that do not agree on any row. By definition of $\mathcal{T}$, it follows that there exist $a \in g^{-1}(0)$ and $b \in g^{-1}(1)$ such that $(X, a) \in \mathcal{X}_\ell$ and $(X', b) \in \mathcal{Y}_\ell$. In particular, this means that if we give to Alice and Bob the inputs (X, a) and (X', b), respectively, the protocol will reach the leaf ℓ.

However, this is a contradiction: on the one hand, ℓ is a leaf on which the protocol outputs $\perp$. On the other hand, the players are not allowed to output $\perp$ on inputs (X, a), (X', b), since X and X' differ on all their rows, and in particular differ on the all the rows j for which $a_j \neq b_j$. The claim follows. ∎

Finally, we observe that Claim 3.5 is just another way of saying that $\mathcal{T}$ satisfies the 1-agreement property (Definition 2.21), when viewed as a set of strings in Σ^m over the alphabet $\Sigma = \{0,1\}^n$. Therefore, Lemma 2.23 implies that $|\mathcal{T}| \leq 2^{(m-1)\cdot n}$, as required.

3.2. Proof of Lemma 3.4. We would like to prove that the number of pairs (ℓ, X) is at least $2^{mn-1} \cdot 2^{-O(\frac{m \log m}{n})} \cdot \mathsf{L}(g)$. We start with the following auxiliary definition of the protocol Π_X, which can be thought of as the protocol obtained from Π by fixing the players' matrices to be X. The following definition will be useful for formalizing what it means that Π solves R_g.

DEFINITION 3.6 (protocol subtrees). *Let $X \in \{0,1\}^{m \times n}$. Let Π_X be the protocol that is obtained from Π as follows: in the protocol tree of Π, we replace each rectangle $\mathcal{X}_v \times \mathcal{Y}_v$ with the rectangle $\mathcal{X}'_v \times \mathcal{Y}'_v$ defined by*

$$\mathcal{X}'_v \stackrel{\text{def}}{=} \{a : (X, a) \in \mathcal{X}_v\},$$
$$\mathcal{Y}'_v \stackrel{\text{def}}{=} \{b : (X, b) \in \mathcal{Y}_v\}.$$

Then, we remove all vertices whose rectangles are empty, and merge all pairs of vertices that have identical rectangles.

In order to prove the lower bound, we partition the matrices X into "good matrices" and "bad matrices." Intuitively, a "good matrix" is a matrix X for which Π_X solves R_g. We will derive the lower bound by showing that for each good matrix X, there are about $\mathsf{L}(g)$ pairs (ℓ, X), and that there are many good matrices. We define good and bad matrices as follows.

DEFINITION 3.7 (good matrices). *Let $t \stackrel{\text{def}}{=} \lceil \frac{6m}{n} \rceil + 2$. A matrix $X \in \{0,1\}^{m \times n}$ is* good *if Π_X is a protocol that solves the* relaxed *KW problem $R_g(t)$ (see Definition* 2.15*). Otherwise, we say that X is* bad.

The following lemma says that good matrices have many pairs (ℓ, X), and it is an immediate corollary of Proposition 2.19 (which says that $R_g(t)$ is not much easier than R_g).

LEMMA 3.8. *For every good matrix X, the protocol Π_X has at least $2^{-t \cdot (\log m + 2)} \cdot \mathsf{L}(g)$ leaves. In other words, there are at least $2^{-t \cdot (\log m + 2)} \cdot \mathsf{L}(g)$ pairs (ℓ, X).*

In the next subsection, we will prove the following lemma, which says that there are not many bad matrices, and therefore there are many good matrices.

LEMMA 3.9. *The number of bad matrices is at most $2^{-m} \cdot 2^{m \cdot n}$. Thus, the number of good matrices is at least $(1 - 2^{-m}) \cdot 2^{mn} \geq 2^{m \cdot n - 1}$.*

Together, Lemmas 3.8 and 3.9 imply Lemma 3.4, as required.

3.2.1. Proof of Lemma 3.9. The intuition for the proof is the following: recall that Alice and Bob output $\perp$, and this means that they have to be convinced that their matrices agree on some row j for which $a_j \neq b_j$. However, when X is bad, Alice and Bob do not know an index j such that $a_j \neq b_j$ 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, and they can only do so for a few matrices. We formalize this below.

First, recall that a matrix X is bad if and only if Π_X does not solve the relaxed KW problem $R_g(t)$. This implies that there exists some leaf ℓ' of Π_X, which is labeled with a rectangle $\mathcal{X}'_\ell \times \mathcal{Y}'_\ell$ and a string $a \in \mathcal{X}'_\ell$, such that the following holds:

- For every $\mathcal{J} \subseteq [m]$ such that $|\mathcal{J}| < t$, there exists $b \in \mathcal{Y}'_\ell$ such that $a|_{\mathcal{J}} = b|_{\mathcal{J}}$.

Going back from Π_X to Π, it follows that there exists some leaf ℓ of Π, which is labeled with a rectangle $\mathcal{X}_\ell \times \mathcal{Y}_\ell$ and a string $a \in g^{-1}(0)$, such that the following hold:

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

Now, without loss of generality, we may assume that

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

since otherwise Theorem 1.6 would follow immediately. Therefore, it suffices to prove that every pair of a leaf ℓ and a string a is "responsible" for at most $2^{-(3\cdot m+n)} \cdot 2^{m\cdot n}$ bad matrices. This would imply that there are at most $2^{-m} \cdot 2^{m\cdot n}$ bad matrices, by summing over all leaves of Π (at most 2^{m+n}) and all strings a (at most 2^m).

To this end, fix a leaf ℓ of Π and a string $a \in g^{-1}(0)$. Let $\mathcal{T}$ be the set of bad matrices that are supported by ℓ and a. We prove that $|\mathcal{T}| \leq 2^{-(3\cdot m+n)} \cdot 2^{m\cdot n}$. The key idea is that since Alice does not know a small set $\mathcal{J}$ such that $a|_{\mathcal{J}} \neq b|_{\mathcal{J}}$, 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 3.10. *Every two matrices $X, X' \in \mathcal{T}$ agree on at least t rows.*

Proof. Let $X, X' \in \mathcal{T}$, and let $\mathcal{J}$ be the set of rows on which they agree. By definition of $\mathcal{T}$, it holds that $(X, a), (X', a) \in \mathcal{T}$. Suppose that $|\mathcal{J}| < t$. Then, by the assumption on ℓ and a, there exists $b \in g^{-1}(1)$ such that $(X, b) \in \mathcal{Y}_\ell$ and $a|_{\mathcal{J}} = b|_{\mathcal{J}}$.

Next, observe that if we give the input (X', a) to Alice and (X, b) to Bob, the protocol will reach the leaf ℓ. Now, ℓ is a rejecting leaf, and therefore there must exist some index $j \in [m]$ such that $a_j \neq b_j$ but $X_j = X'_j$. However, we know that $a|_{\mathcal{J}} = b|_{\mathcal{J}}$, and therefore $j \notin \mathcal{J}$. It follows that X and Y agree on a row outside $\mathcal{J}$, contradicting the definition of $\mathcal{J}$. ∎

Finally, we observe that Claim 3.10 is just another way of saying that $\mathcal{T}$ satisfies the t-agreement property (Definition 2.21), when viewed as a set of strings in $\mathbb{F}^m$ over the alphabet $\mathbb{F} = \{0,1\}^n$. Therefore, Lemma 2.24 implies[8] that $|\mathcal{T}| \leq 2^{(m-t)\cdot n}$. Wrapping up, it follows that

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

as required.

Remark 3.11. Note that Lemma 2.24 can only be applied if $m \leq 2^n$. However, this can be assumed without loss of generality, since for $m \geq 2^n$, the lower bound of Theorem 1.6 becomes less than $\log \mathsf{L}(g)$. However, it is easy to prove a lower bound of $\log \mathsf{L}(g)$ on $\log \mathsf{L}(R_{g \diamond \mathrm{U}_n})$ by reducing R_g to $R_{g \diamond \mathrm{U}_n}$.

Acknowledgments. We would like to thank Noga Alon, Gillat Kol, Omer Reingold, Luca Trevisan, and Ryan Williams for helpful discussions. We are also grateful to anonymous referees for comments that considerably improved the presentation of this work.

[8] Note that Lemma 2.24 requires that $m \leq |\mathbb{F}|$. However, we may assume this inequality without loss of generality, since otherwise Theorem 1.6 holds vacuously.

REFERENCES

[ADFS04] N. Alon, I. Dinur, E. Friedgut, and B. Sudakov, *Graph products, Fourier analysis and spectral techniques*, Geom. Funct. Anal., 14 (2004), pp. 913–940, https://doi.org/10.1007/s00039-004-0478-3.

[And87] A. E. Andreev, *On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes*, Moscow Univ. Math. Bull., 42 (1987), pp. 63–66.

[BB94] M. L. Bonet and S. R. Buss, *Size-depth tradeoffs for Boolean formulae*, Inform. Process. Lett., 49 (1994), pp. 151–155, https://doi.org/10.1016/0020-0190(94)90093-0.

[BBCR13] B. Barak, M. Braverman, X. Chen, and A. Rao, *How to compress interactive communication*, SIAM J. Comput., 42 (2013), pp. 1327–1363, https://doi.org/10.1137/100811969.

[Bre74] R. P. Brent, *The parallel evaluation of general arithmetic expressions*, J. ACM, 21 (1974), pp. 201–206, https://doi.org/10.1145/321812.321815.

[EIRS01] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall, *Communication complexity towards lower bounds on circuit depth*, Comput. Complexity, 10 (2001), pp. 210–246, https://doi.org/10.1007/s00037-001-8195-x.

[FT99] P. Frankl and N. Tokushige, *The Erdős-Ko-Rado theorem for integer sequences*, Combinatorica, 19 (1999), pp. 55–63, https://doi.org/10.1007/s004930050045.

[GL74] D. Greenwell and L. Lovász, *Applications of product colouring*, Acta Math. Hungar., 25 (1974), pp. 335–340, https://doi.org/10.1007/BF01886093.

[GMWW13] D. Gavinsky, O. Meir, O. Weinstein, and A. Wigderson, *Toward Better Formula Lower Bounds: An Information Complexity Approach to the KRW Composition Conjecture*, Electronic Colloquium on Computational Complexity (ECCC), TR13-190, 2013.

[GS91] M. Grigni and M. Sipser, *Monotone separation of logspace from NC^1*, in Proceedings of the Sixth Annual Structure in Complexity Theory Conference, IEEE Computer Society, Washington, DC, 1991, pp. 294–298, https://doi.org/10.1109/SCT.1991.160272.

[Hås98] J. Håstad, *The shrinkage exponent of de Morgan formulas is* 2, SIAM J. Comput., 27 (1998), pp. 48–64, https://doi.org/10.1137/S0097539794261556.

[HJP95] J. Håstad, S. Jukna, and P. Pudlák, *Top-down lower bounds for depth-three circuits*, Comput. Complexity, 5 (1995), pp. 99–112, https://doi.org/10.1007/BF01268140.

[HW93] J. Håstad and A. Wigderson, *Composition of the universal relation*, in Advances in Computational Complexity Theory, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 13, AMS, Providence, RI, 1993, pp. 119–134.

[IN93] R. Impagliazzo and N. Nisan, *The effect of random restrictions on formula size*, Random Structures Algorithms, 4 (1993), pp. 121–134, https://doi.org/10.1002/rsa.3240040202.

[Khr72] V. M. Khrapchenko, *A method of obtaining lower bounds for the complexity of π-schemes*, Math. Notes Acad. Sci. USSR, 11 (1972), pp. 474–479.

[KKN95] M. Karchmer, E. Kushilevitz, and N. Nisan, *Fractional covers and communication complexity*, SIAM J. Discrete Math., 8 (1995), pp. 76–92, https://doi.org/10.1137/S0895480192238482.

[KRW95] M. Karchmer, R. Raz, and A. Wigderson, *Super-logarithmic depth lower bounds via the direct sum in communication complexity*, Comput. Complexity, 5 (1995), pp. 191–204, https://doi.org/10.1007/BF01206317.

[KW90] M. Karchmer and A. Wigderson, *Monotone circuits for connectivity require super-logarithmic depth*, SIAM J. Discrete Math., 3 (1990), pp. 255–265, https://doi.org/10.1137/0403021.

[PZ93] M. S. Paterson and U. Zwick, *Shrinkage of de Morgan formulae under restriction*, Random Structures Algorithms, 4 (1993), pp. 135–150, https://doi.org/10.1002/rsa.3240040203.

[Raz90] A. A. Razborov, *Applications of matrix methods to the theory of lower bounds in computational complexity*, Combinatorica, 10 (1990), pp. 81–93, https://doi.org/10.1007/BF02122698.

[RR97] A. A. Razborov and S. Rudich, *Natural proofs*, J. Comput. Syst. Sci., 55 (1997), pp. 24–35, https://doi.org/10.1006/jcss.1997.1494.

[RW92] R. Raz and A. Wigderson, *Monotone circuits for matching require linear depth*, J. ACM, 39 (1992), pp. 736–744, https://doi.org/10.1145/146637.146684.

[Spi71] P. M. Spira, *On time-hardware complexity tradeoffs for Boolean functions*, in Proceedings of the Fourth Hawaii International Symposium on System Sciences, 1971, pp. 525–527.

[Sub61] B. A. Subbotovskaya, *Realizations of linear functions by formulas using* $+, \cdot, -$, Soviet Math. Dokl., 2 (1961), pp. 110–112.

[Yao79] A. C.-C. Yao, *Some complexity questions related to distributive computing (preliminary report)*, in Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing (STOC), 1979, pp. 209–213, https://doi.org/10.1145/800135.804414.