The KW Games as a Teaser

Oded Goldreich[*]

December 30, 2021

Abstract

We advocate using KW-games as a teaser (or "riddle") for a complexity theoretic course. In particular, stating the KW-game for a familiar NP-complete problem such as 3-Colorability and asking to prove that it requires more than polylogarithmic communication poses a seemingly tractable question that requires no background. The fact that this is actually a formidable question puts much of complexity theory in the right perspective. Indeed, complexity theory contains numerous extremely appealing problems that require little background (beyond basic comfort with the notion of computation) and yet are formidable.

1 The riddle

After making sure that the audience is familiar with 3-Colorability, pose the following riddle regarding interaction between two parties, called Alice and Bob.

> Alice gets a 3-colorable n-vertex (labeled) graph, and Bob gets an n-vertex graph that is not 3-colorable, and they want to find a pair of vertices that are connected in one graph but not in the other. (Such a pair definitely exists, since the graphs are different.) How many bits do they need to exchange in order to achieve this goal?
>
> In particular, do $o(n)$ bit suffice? Can one prove that $\text{poly}(\log n)$ bits do not suffice? How about $10 \log_2 n$?

Note that the trivial solution is for one party, say Alice, to send her graph to the other party.

Actually, a nice observation that may be a teaser for the study of $\mathcal{NP}$ is that the parties can do better than sending the entire graph by using an NP-witness; that is, Alice can send a 3-coloring of *her graph* to Bob, who reply with an edge that is monochromatic in *his graph* (i.e., its two endpoints have that same color in his graph). Hence, $O(n)$ bits rather than $O(n^2)$ bits of communication suffice, where n is the number of vertices in the graph.

The foregoing task is a special case of a KW-game (introduced by Karchmer and Wigderson [4]). Of course, proving that this KW-game requires more than a polylogarithmic amount communication will imply that $\mathcal{NP}$ is not contained in $\mathcal{NC}$ (where $\mathcal{NC}$ is as in Definition 3.1). Even a $10 \log_2 n$ lower bound would be astonishing. This is due to the correspondence (shown in Section 3) between the amount of communication in this KW-game and the depth of Boolean circuits that decide whether n-vertex graphs are 3-colorable.

[*] Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL. Email: oded.goldreich@weizmann.ac.il.

2 A wider context

We start with a general statement of the KW-game, which was introduced by Karchmer and Wigderson [4].

Definition 2.1 (the KW-game): *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$, we consider the communication between two parties, called Alice and Bob, each given an n-bit string. Alice is given $x \in f^{-1}(1)$, Bob is given $y \in f^{-1}(0)$, and their aim is to output an index $i \in [n]$ such that $x_i \neq y_i$. Towards this end, they send messages to each other, by following predetermined strategies that depend on f. We denote the above game by $\mathtt{KW}_f$.*

We may require both parties to output the same index or just require one of them to do so, since we do not really care about an additive $\log_2 n$ term (which suffices for communicating the index). For sake of concreteness, we adopt the former convention.

The communication between the parties takes place in rounds, and it is predetermined which party goes first, and who goes last (i.e., terminates the interaction). Suppose that Alice goes first, and let A denote the strategy that she employs. Then, on input x, she will start by sending the message $\alpha_1 \leftarrow A(x)$, and Bob, employing the strategy B (and having input y), will answer with $\beta_1 \leftarrow B(y;\alpha_1)$. In the j^{th} round Alice will send the message $\alpha_j \leftarrow A(x;\beta_1,...,\beta_{j-1})$, and Bob will answer with $\beta_j \leftarrow B(y;\alpha_1,...,\alpha_j)$. At some predetermined round t, Bob's will terminate, and $B(y;\alpha_1,...,\alpha_t)$ is considered the output (rather than a message sent to Alice).[1] The question we focus upon is *how many bits are exchanged in the communication, on the worst-case* $(x,y) \in f^{-1}(1) \times f^{-1}(0)$, when using the best possible pair of strategies? This number, denoted $\mathcal{CC}(\mathtt{KW}_f)$, is called the communication complexity of $\mathtt{KW}_f$.

Example 2.2 (a trivial case): *Consider the function $f(z) = z_1$. Then, the index 1 is always a correct output, since $x_1 = 1$ and $y_1 = 0$ must hold for any $x \in f^{-1}(1)$ and $y \in f^{-1}(0)$. Hence, there is no need for communication between the parties, and $\mathcal{CC}(\mathtt{KW}_f) = 0$.*

Example 2.3 (easy cases): *Consider the function $f(z) = \bigvee_{i\in[n]} z_i$. Then, an index i such that $x_i = 1$ is a valid output, because $f(y) = 0$ implies that $y_i = 0$ must hold* (i.e., $f^{-1}(0) = \{0^n\}$). *Hence, Alice knows such an index i and just sends it to Bob, which means that $\mathcal{CC}(\mathtt{KW}_f) \leq \lceil \log_2 n \rceil$. The same holds for $f(z) = \bigwedge_{i\in[n]} z_i$, but in this case Bob knows the answer and sends it to Alice.*

It is quite easy to see that the upper bound is actually tight.[2] The same holds for any function f that depends on all its bits; this is easy to see from the connection to circuit depth that will be reviewed in Section 3.

Example 2.4 (another easy case): *Consider the function $f(z) = \bigoplus_{i\in[n]} z_i$, and observe that $f(z'z'') = f(z') \oplus f(z'')$. This observation suggest a binary search for the desired index. In the first round, Alice partitions x into two equal parts x' and x'', and sends $f(x')$ to Bob, who responds with $f(y')$. If the two values are equal then the parties proceed with x'' and y''; otherwise, they proceed with x' and y'. Hence, we get $\mathcal{CC}(\mathtt{KW}_f) \leq 2 \cdot \lceil \log_2 n \rceil$.*

[1] By our convention, Alice must know this index already; this can be formalized by requiring that $A(x;\beta_1,...,\beta_{j-1},\lambda) = B(y;\alpha_1,...,\alpha_t)$, where λ denotes the empty string.

[2] For $f(z) = \bigvee_{i\in[n]} z_i$, consider the case that $x \in X \stackrel{\text{def}}{=} \{0^{i-1}10^{n-i} : i \in [n]\}$ and $y = 0^n$. Then, the communication between Alice and Bob cannot be the same for two different x's in X (since Bob's outputs must be different in such two cases).

The foregoing upper bound is also tight. This follows from combining the connection to circuit depth with a lower bound on the circuit depth of parity.[3]

In contrast to the foregoing easy cases, it turns out that almost all functions f yield KW-games of extremely high communication complexity; that is, $\mathcal{CC}(\texttt{KW}_f) \geq n - O(\log n)$. This fact can also be proved by the connection to circuit depth, but it does not yield an "explicit" function, let alone a function (specifying a set) in NP.

Indeed, at this point, we should clarify that n should be viewed as a parameter, and that we are not talking about a single n and a single $f : \{0,1\}^n \to \{0,1\}$, but rather about a collection of function $\{f_n : \{0,1\}^n \to \{0,1\}\}_{n \in \mathbb{N}}$. We consider the communication complexity of $\texttt{KW}_{f_n}$ as a function of n, but do not bound the computational resources of the parties (i.e., the time complexity of the strategies that they employ).

3 The connection to circuit depth

KW-games were introduced by Karchmer and Wigderson [4] with the intention of providing a technique for proving lower bounds on the depth of (bounded fan-in) Boolean circuits computing various explicit functions. Pivotal to that technique is the connection (which they established) between the communication complexity of $\texttt{KW}_f$ and the depth of Boolean circuits computing f.

At this point we need to introduce the model of Boolean circuits; here is a brief outline (see [2, Sec. 1.2.4.1] for more details). Basically, a Boolean circuit is a directed acyclic graph (DAG) with source vertices corresponding to variables (or their negations), and other vertices (including the sinks) associated with Boolean gates. (Recall that source vertices have in-degree zero, whereas sink vertices have out-degree zero.) We refer to the bounded fan-in model in which it is postulated that each non-source vertex has in-degree two and is associated with either an AND-gate or an OR-gate. (The vertices having directed edges to the vertex are called its children.) Assigning values to the variables (and proceeding from the sources to the sink (i.e., from children to their parents)), the circuit induces a computation in the natural manner, and its output is obtained at the sink vertices. Here we use Boolean circuits with a single output bit (i.e., a single sink vertex). Two central complexity classes are the following.

Definition 3.1 (the classes $\mathcal{P}$/poly and $\mathcal{NC}$):

- *A Boolean predicate $f : \{0,1\}^* \to \{0,1\}$ is in $\mathcal{P}$/poly if there exists a polynomial p such that for every n the restriction of f to inputs of length n can be computed by a circuit of size at most $p(n)$, where the* size *of a Boolean circuit is the number of vertices in it.*

- *A Boolean predicate $f : \{0,1\}^* \to \{0,1\}$ is in $\mathcal{NC}$ if there exists a polynomial p such that for every n the restriction of f to inputs of length n can be computed by a circuit of size at*

[3] An alternative direct proof follows as a special case of [4, Thm. 3.1]. We consider the restriction of $\texttt{KW}_f$ to pairs (x,y) that are at Hamming distance 1 of one another. This set of pairs equals $P \stackrel{\rm def}{=} \{(x, x \oplus 0^{i-1}10^{n-1}) : x \in f^{-1}(1) \& i \in [n]\}$, and in this case the valid output is uniquely determined. Let $\overline{\alpha}(x,y)$ be the sequence of messages sent by Alice, in interaction on input pair (x,y). Then, for every fixed y, the $\overline{\alpha}(y \oplus 0^{i-1}10^{n-i}, y)$'s must be distinct, since otherwise Bob's answer is wrong on some input (i.e., its outputs should be distinct). Since these $\overline{\alpha}$'s may not be a prefix of one another, for every $y \in f^{-1}(0)$, it follows that $\mathrm{E}_{i \in [n]}[|\overline{\alpha}(y \oplus 0^{i-1}10^{n-i}, y)|] \geq \log_2 n$. Hence, $\mathrm{E}_{(x,y) \in P}[|\overline{\alpha}(x,y)|] \geq \log_2 n$, and similarly $\mathrm{E}_{(x,y) \in P}[|\overline{\beta}(x,y)|] \geq \log_2 n$, where $\overline{\beta}(x,y)$ denotes the sequence of Bob's messages.

most $p(n)$ and depth at most $p(\log n)$, where the depth *of a Boolean circuit is the length of the longest path from one of its sources to its sink.*

Recall that Boolean predicates are associated with sets; that is, f is associated with the set $f^{-1}(1)$. We stress that the sequence of circuits used in the definition is arbitrary; in contrast, "uniform" complexity classes are defined by requiring that on input n the n^{th} circuit can be constructed within certain resource bounds (e.g., in poly(n)-time, or even in $O(\log n)$-space).

We shall focus on the depth of Boolean circuits, and will not confine ourself to any specific depth. That is, for any $f : \{0,1\}^n \to \{0,1\}$, we denote by $\mathcal{D}(f)$ the minimal depth of a Boolean circuit that computes f. Indeed, we move back from predicates defined over all bit strings to ones defined only over bit strings of a specific length, denoted n. (But n should be viewed as a parameter.)

The cleanest relation between $\mathcal{CC}(\mathtt{KW}_f)$ and $\mathcal{D}(f)$ is obtained by using an auxiliary convention regarding the strategies of the two parties. The (subtle) issue at hand is determining when does the current message end (i.e., when does a party complete the transmission of the current message). This issue is resolved by requiring that, at any stage, the set of possible messages set at this stage must be prefix-free (where a set of strings S is prefix-free if no string in S is a proper prefix of another string in S).[4] Note that failure to adopt this convention allows for the possible encoding of information in the length of messages (i.e., $\log_2 |\alpha|$ bits can be encoded by splitting the message α between two messages), which means that information is communicated without being accounted for in the communication complexity. On the other hand, strategies that do not satisfy this convention can be converted to ones that satisfy the convention, while increasing the communication complexity by at most a logarithmic factor; for example, we may prepend each message with a $\lceil \log_2 m \rceil$-bit long header that indicates the length of the message, where m is the communication complexity of the original protocol (i.e., pair of strategies). We mention that one often presents the communication model by mandating messages of fixed length (e.g., one-bit messages)[5], but the equivalence to circuit-depth requires using an analogous convention for circuits (i.e., the child of each vertex is associated with a gate type different than the one of its parent).

With these preliminaries in place, we shall prove that $\mathcal{CC}(\mathtt{KW}_f) = \mathcal{D}(f)$. This result is due to Karchmer and Wigderson [4], and was intended as a technique for proving lower bounds on $\mathcal{D}(f)$; that is, it is suggested to first establish a lower bound on $\mathcal{CC}(\mathtt{KW}_f)$, and then use $\mathcal{D}(f) \geq \mathcal{CC}(\mathtt{KW}_f)$. We shall also use $\mathcal{CC}(\mathtt{KW}_f) = \mathcal{D}(f)$ in order to establish lower bounds on $\mathcal{CC}(\mathtt{KW}_f)$ by using lower bounds on $\mathcal{D}(f)$ along with $\mathcal{CC}(\mathtt{KW}_f) \geq \mathcal{D}(f)$. But let us first show that $\mathcal{D}(f) \geq \mathcal{CC}(\mathtt{KW}_f)$, or rather its contrapositive.

Theorem 3.2 (using circuits to obtain protocols): *For every $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathcal{CC}(\mathtt{KW}_f) \leq \mathcal{D}(f)$.*

In particular, in order to show that $\mathcal{NP}$ (resp., $\mathcal{P}$) is not contained in $\mathcal{NC}$, pick a set S in the former class and prove that the restriction of its characteristic function to n-bit strings yield a KW-game of communication complexity that is larger than poly$(\log n)$.

[4] Recall that $A(x; \beta_1, ..., \beta_{j-1})$ denotes the message sent by Alice in the j^{th} round, when she holds input x and received the messages $\beta_1, ..., \beta_{j-1}$. Then, by the foregoing convention, for every j and $\beta_1, ..., \beta_{j-1}$, the potential message $A(x; \beta_1, ..., \beta_{j-1})$ must be a prefix of $A(x'; \beta_1, ..., \beta_{j-1})$, and ditto for $B(y; \alpha_1, ..., \alpha_j)$ and $B(y'; \alpha_1, ..., \alpha_j)$.

[5] This more restricted formalism can be enforced at the cost of using dummy messages in the opposite direction.

Proof: The parties, who know the Boolean circuit (computing f), essentially trace a path from the sink vertex to some source vertex such that the corresponding sub-circuits (all along this path) evaluate to different values at the two local inputs (i.e., x and y).[6] Specifically, the parties proceed in iterations such that at the beginning of each iteration they "reside" in a vertex v such that the sub-circuit rooted at v, denoted C_v, evaluates to different outputs under x and y (i.e., $C_v(x) \neq C_v(y)$). Denoting the two children of v by w_1 and w_2, the parties determine $j \in \{1,2\}$ such that the sub-circuit rooted at w_j yields different values (i.e., $C_{w_j}(x) \neq C_{w_j}(y)$). This can be done by exchanging the values $C_{w_1}(x)$ and $C_{w_1}(y)$. Once the parties reach a source vertex, they output the index of the variable with which this vertex is associated. Note that the communication complexity of this protocol is $2 \cdot \mathcal{D}(f)$. A more economical implementation is detailed next.

Let C be a circuit of depth $\mathcal{D}(f)$ computing f, and let C_v be the sub-circuit of C rooted at v (i.e., the sub-DAG that feeds into the vertex v). Denoting the sink vertex of C by s, it holds that $C_s(x) = 1 \neq C_s(y)$. After j rounds (where initially $j = 0$), the parties are at a vertex v, which is at distance j from the vertex s, and it holds that $C_v(x) = 1 \neq C_v(y)$. If v is a source vertex associated with some variable (or its negation), then the parties output the index of that variable, denoted i. (Note that $x_i = 1 \neq y_i$ if the vertex is associated with the i^{th} variable, whereas $x_i = 0 \neq y_i$ if the vertex is associated with the negation of the i^{th} variable.) Otherwise (which mandates that $j < \mathcal{D}(f)$), we consider two cases.

1. Vertex v is associated with an AND-gate; that is, $C_v = C_{w_1} \wedge C_{w_2}$, where w_1 and w_2 are the vertices that feed v (i.e., have an outgoing edge to v). In this case it holds that $C_v(x) = C_{w_1}(x) \wedge C_{w_2}(x) = 1$, whereas $C_v(y) = C_{w_1}(y) \wedge C_{w_2}(y) = 0$. This means that there exists $k \in \{1,2\}$ such that $C_{w_k}(y) = 0$ whereas $C_{w_{k'}}(x) = 1$ holds for both $k' \in \{1,2\}$. Hence, $C_{w_k}(x) = 1 \neq C_{w_k}(y)$. In this case, Bob just sends (the parity of) this k to Alice, and the two parties set $v \leftarrow w_k$ for the next iteration.

2. Vertex v is associated with an OR-gate; that is, $C_v = C_{w_1} \vee C_{w_2}$, where w_1 and w_2 are the vertices that feed v. In this case it holds that $C_v(x) = C_{w_1}(x) \vee C_{w_2}(x) = 1$, whereas $C_v(y) = C_{w_1}(y) \vee C_{w_2}(y) = 0$. This means that there exists $k \in \{1,2\}$ such that $C_{w_k}(x) = 1$ whereas $C_{w_{k'}}(y) = 0$ holds for both $k' \in \{1,2\}$. Hence, Alice just sends (the parity of) this k to Bob, and the two parties set $v \leftarrow w_k$ for the next iteration.

After at most $\mathcal{D}(f)$ iterations, the parties reach a source vertex v that satisfies $C_v(x) = 1 \neq C_v(y)$ and output the index of the corresponding variable. (Note that not all source vertices are necessarily at distance $\mathcal{D}(f)$ from the sink node; only the maximum distance equals $\mathcal{D}(f)$.)

We mention that both strategies rely on the circuit C. Hence, if we only know that C exists, then we can only infer that such strategies exist. However, if C can be constructed within some time (or space) bounds, then essentially the same bounds apply to the computational complexity of the strategies. ■

The following result asserts that the technique suggested for establishing depth lower bounds is actually "complete" in the sense that if the lower bound is valid then it can be established by considering the corresponding KW-game.

Theorem 3.3 (using protocols to obtain circuits): *For every $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathcal{D}(f) \leq \mathcal{CC}(\mathtt{KW}_f)$.*

[6] By a sub-circuit corresponding to (resp., rooted at) a vertex v, we mean the circuit obtained by omitting all gates (i.e., vertices) that do not have a directed path to vertex v.

As mentioned above, Theorem 3.3 can also be used for establishing communication complexity lower bounds on KW-games of functions for which we know a depth lower bound. In particular, in order to show that almost all functions f have communication complexity at least $n - O(\log n)$, show that with high probability a random function f has no circuit of size $2^n/O(n)$ (see [2, Exer. 4.1]), which implies $\mathcal{D}(f) \geq n - \log_2 n - O(1)$.

Proof: We consider a generalization of the claimed result, which refers to two disjoint subsets of $\{0,1\}^n$, denoted X and Y. In the generalized KW-game, denoted $\texttt{KW}_{X,Y}$, Alice gets $x \in X$ and Bob gets $y \in Y$, and their task (as before) is to find i such that $x_i \neq y_i$. Indeed, $\texttt{KW}_f$ is a special case obtained by letting $X = f^{-1}(1)$ and $Y = f^{-1}(0)$. Our goal is to construct a circuit C (of depth $\mathcal{CC}(\texttt{KW}_{X,Y})$) such that $C(x) = 1$ for every $x \in X$ and $C(y) = 0$ for every $y \in Y$, where we don't care about the value of C on inputs outside $X \cup Y$; that is, C solves the "promise problem" (see [2, Sec. 1.2.2.3]) of distinguishing inputs in X from inputs in Y.

We decompose the protocol for $\texttt{KW}_{X,Y}$ according to the first bit (in the first message) that is being sent. If Alice sends the first bit, then we consider the residual protocols for $\texttt{KW}_{X_0,Y}$ and $\texttt{KW}_{X_1,Y}$, where X_b is the subset of X that causes Alice to send b as the first bit.[7] Assuming that we already have circuits C_b's that distinguish inputs in X_b from inputs in Y, for both $b \in \{0,1\}$, we construct the circuit $C(z) = C_0(z) \vee C_1(z)$. Note that if $x \in X$, then $x \in X_b$ for some $b \in \{0,1\}$, and $C(x) = C_0(x) \vee C_1(x) = 1$ follows, since $C_b(x) = 1$ must hold (for every $x \in X_b$). On the other hand, if $y \in Y$, then $C(y) = C_0(y) \vee C_1(y) = 0$, since $C_0(y) = 0$ and $C_1(y) = 0$ must both hold.

An analogous argument holds for a first bit sent by Bob, but in this case we construct the circuit $C(z) = C_0(z) \wedge C_1(z)$. (That is, Bob's first bit partitions Y into (Y_0, Y_1), and C_b distinguishes inputs in X from inputs in Y_b.)[8] Hence, assuming we can construct (depth t) circuits for promise problems that correspond to generalized KW-games of communication complexity t, we construct such (depth $t+1$) circuits for communication complexity $t+1$. The basis of this induction corresponds to KW-games of communication complexity 0. In this case, an index i such that $x_i \neq y_i$ holds for every $(x,y) \in X \times Y$ is uniquely determined by (the current) $X \times Y$. Furthermore, there exists a bit b such that, for every $x \in X$ (resp., $y \in Y$), it holds that $x_i = b$ (resp., $y_i = 1 - b$).[9] If $b = 1$, then we use the (depth 0) circuit $C(z) = z_i$, and otherwise we use $C(z) = \neg z_i$.

As hinted above, the actual proof is by induction. The induction hypothesis is that, for every disjoint pair (X,Y) such that $\mathcal{CC}(\texttt{KW}_{X,Y}) \leq t$, we can construct circuits of depth t that distinguish inputs in X from inputs in Y. The basis of the induction corresponds to $t = 0$, and (by our convention) the circuits $C(z) = z_i$ and $C(z) = \neg z_i$ have depth 0. The induction step uses the fact that the protocols for the residual games (i.e., $\texttt{KW}_{X_0,Y}$, $\texttt{KW}_{X_1,Y}$, $\texttt{KW}_{X,Y_0}$, and $\texttt{KW}_{X,Y_1}$) have communication complexity at most $\mathcal{CC}(\texttt{KW}_{X,Y}) - 1$. Note that, analogously to Theorem 3.2, the computational complexity of constructing the circuits is determined by the computational complexity of the strategies. ■

[7] We assume, without loss of generality, that both X_0 and X_1 are non-empty; otherwise, sending the first bit is redundant.

[8] Indeed, Y_b is the subset of Y that causes Bob to send b as the first bit. On the one hand, if $y \in Y$, then $y \in Y_b$ for some $b \in \{0,1\}$, and $C(y) = C_0(y) \wedge C_1(y) = 0$ follows, since $C_b(y) = 0$ must hold. On the other hand, if $x \in X$, then $C(x) = C_0(x) \wedge C_1(x) = 1$, since $C_0(x) = 1$ and $C_1(x) = 1$ must both hold.

[9] Assume, towards the contradiction, that $x', x'' \in X$ satisfy $x'_i \neq x''_i$. Then, it cannot be that both $x'_i \neq y_i$ and $x''_i \neq y_i$.

4 Variations

As noted above, KW-games and their relation to circuit depth were devised by Karchmer and Wigderson [4] with the intension of providing a technique for proving lower bounds on the depth of (Boolean) circuits (of bounded fan-in) that compute explicit functions (e.g., function in $\mathcal{P}$). This suggestion was materialized by them in the context of monotone circuits. Needless to say, a corresponding notion of KW-games needs to be defined for that to work. A different incarnation refers to "alternating circuits" (underlying the definition of the class $\mathcal{AC}$). These two variants are reviewed below.

4.1 Monotone circuits and monotone KW-games

A monotone circuit is one in which all variables appear non-negated. Needless to say, such circuits can only compute monotone functions (i.e., functions f that satisfy $f(x) \geq f(y)$ for every $x > y$).[10] For a monotone function f, we let $\mathrm{m}\mathcal{D}(f)$ denote the minimum depth of a monotone circuit computing f. The monotone KW-game for f, denoted $\mathtt{mKW}_f$, is defined as the KW-game except that the parties are required to output an index i such that $x_i = 1 \neq y_i$ (whereas in Definition 2.1 it was not required that $x_i = 1$). A careful examination of the proofs of Theorems 3.2 and 3.3 shows that they yield the following result.

Theorem 4.1 (monotone circuits vs monotone KW-games): *For every $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathcal{CC}(\mathtt{mKW}_f) = \mathrm{m}\mathcal{D}(f)$.*

(A variant of) Theorem 4.1 was used by Karchmer and Wigderson [4] in order to prove that deciding connectivity of n-vertex graphs requires monotone circuits of depth $\Omega(\log^2 n)$. They proved this by showing that the monotone communication complexity of this problem is $\Omega(\log^2 n)$.

4.2 Alternating circuits and alternating KW-games

We say that a Boolean circuit of depth d has t alternations if on every path from the sink vertex to some source vertex there are $t-1$ alternations between AND-gates and OR-gates, and vice versa. Furthermore, on all paths, these alternations occur at the same distance from the sink vertex.[11] It is customary to collapse the gates that are of the same type, forming a circuit of depth t with unbounded fan-in AND-gates and OR-gates, but we shall not follow this convention here. Let us denote by $\mathcal{D}_t(f)$ the minimum depth of a Boolean circuit with at most t alternation that computes f. The corresponding notion of protocols for $\mathtt{KW}_f$ refers to protocols in which Alice and Bob switch turns (in communication) at most $t-1$. We denote the corresponding measure of communication complexity by $\mathcal{CC}_t$. A careful examination of the proofs of Theorems 3.2 and 3.3 shows that they yield the following result.

Theorem 4.2 (alternating circuits vs alternating protocols for KW-games): *For every $f : \{0,1\}^n \to \{0,1\}$ and every t, it holds that $\mathcal{CC}_t(\mathtt{KW}_f) = \mathcal{D}_t(f)$.*

[10] For $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$, it holds that $x > y$ if there exists i such that $x_i = 1 > y_i = 0$ and $x_j \geq y_j$ for all $j \in [n]$.

[11] That is, there exist $b \in \{0,1\}$ and $d_0 \stackrel{\rm def}{=} 0 < d_1 < d_2 < \cdots < d_t \stackrel{\rm def}{=} d$ such that for every $i \in [t]$ all vertices that are at distance $d' \in [d_{i-1}, d_i)$ from the sink vertex are associated with an AND-gate if $i \equiv b \pmod 2$ and with an OR-gate otherwise.

5 Digest

In retrospect, one may argue that the riddle I posed is misleading, but I guess all riddles are so. It is natural for the audience to forget that x is confined to $f^{-1}(1)$ while y is confined to $f^{-1}(0)$, and reach the conclusion that proving a linear lower bound on the communication complexity is easy. Indeed, proving a linear lower bound when the confinement of x and y is waived is easy (see below). But, as Examples 2.2–2.4 illustrate, the foregoing confinement is crucial and under it the communication complexity can be much lower (depending, of course, on f).

Let us consider the "unconfined" problem. Here Alice and Bob get inputs x and y, and it is only guaranteed that $x \neq y$. As before, they need to output an index i such that $x_i \neq y_i$. (Almost equivalently, $x = y$ may be allowed, in which case the output is required to be 0.) In this case a lower bound of n can be proved by reduction from the communication complexity of computing the (in)equality function (i.e., $\texttt{NEQ}(x, y) = 1$ if $x \neq y$ and $\texttt{NEQ}(z, z) = 0$ otherwise).

The communication complexity of the (in)equality function is one of the first results presented in any standard exposition of the area called *communication complexity* (see, e.g., [5, Exer. 1.21] and [6, Thm. 1.16]).[12] It is well known that the communication complexity of (in)equality (of n-bit long strings) is $n + 1$. A simple reduction shows that this implies that the communication complexity of the (unconfined) problem of finding an index on which n-bit strings disagree is at least $n - 1$. (Specifically, to compute $\texttt{NEQ}(x, y)$, the parties invoke the protocol for the unconfined problem, obtain the index i, exchange the i^{th} bit of their inputs, and output 1 if and only if $x_i \neq y_i$.)

6 Notes

With the exception of Theorem 4.2, the contents of this memo originates from the work of Karchmer and Wigderson [4]. I heard Theorem 4.2 from Or Meir.

Communication complexity is a sub-area of complexity theory. It has its own merits as well as applications to other sub-areas of complexity theory and theory of computation at large. Two excellent books that survey the area were written some 25 years apart; the earlier one is by Kushilevitz and Nisan [5], and a recent one is by Rao and Yehudayoff [6].

Circuit complexity is another sub-area of complexity theory. Studies in this area tend to consider non-uniform families of circuits, as we have done in this memo, while believing that the uniformity condition adds a layer of complication that is better avoided in the initial studies. A very brief survey of this area appears in [2, Apdx. B.2], and selected results are presented in [1, Chap. 13]. For a book on the area, see [3].

References

[1] Sanjeev Arora and Boaz Barak. *Complexity Theory: A Modern Approach.* Cambridge University Press, 2009.

[2] Oded Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press, 2008.

[3] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Algorithms and Combinatorics 27, Springer, 2012.

[12] Such expositions typically consider the equality function $\texttt{EQ} = 1 - \texttt{NEQ}$.

[4] Mauricio Karchmer and Avi Wigderson. Monotone Circuits for Connectivity Require Super-Logarithmic Depth. *SIAM J. Discret. Math.*, Vol. 3 (2), pages 255–265, 1990. Preiminiary version in *20th STOC*, pages 539–550, 1988.

[5] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1996.

[6] Anup Rao and Amir Yehudayoff. *Communication Complexity.* Cambridge University Press, 2020.