# Communication Complexity Towards Lower Bounds on Circuit Depth

Jeff Edmonds *

Russell Impagliazzo

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A4

Steven Rudich

Jiří Sgall

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Abstract**

Karchmer, Raz, and Wigderson [2] discuss the circuit depth complexity of $n$ bit Boolean functions constructed by composing up to $d = \log n / \log\log n$ levels of $k = \log n$ bit boolean functions. Any such function is in $AC^1$. They conjecture that circuit depth is additive under composition, which would imply that any (bounded fan-in) circuit for this problem requires $dk \in \Omega(\log^2 n / \log\log n)$ depth. This would separate $AC^1$ from $NC^1$. They recommend using the communication game characterization of circuit depth [3]. They suggest an intermediate problem which they call the Universal Composition Relation. We give an almost optimal lower bound of $dk - O(d^2(k\log k)^{1/2})$ for this problem. In addition, we present a proof, directly in terms of communication complexity, that there is a function on $k$ bits requiring $\Omega(k)$ circuit depth. Although this fact can be easily established using a counting argument, we hope that the ideas in our proof will be incorporated more easily into subsequent arguments which use communication complexity to prove circuit depth bounds.

## 1 Introduction

Karchmer and Wigderson [3] observed that the circuit depth (over the basis $\wedge, \vee, \neg$) of a Boolean function $f$, $Depth(f)$, can be characterized as the communication complexity of a certain game $R_f$. For communication game $G$, we use the notation $CC(G)$ to represent the communication complexity of $G$, i.e., the number of bits communicated during the course of the best protocol solving $G$ on the worst-case instance of $G$. In this notation, $CC(R_f) = Depth(f)$.

Based on this characterization, Karchmer, Raz, and Wigderson [2] sketched a plan of attack for proving a super-logarithmic lower bound for the circuit depth required to compute a specific function in $P$. (In fact, the function is in $AC^1$). This function, which we will refer to as the Iterated Multiplexor, can be described as follows. The function has two parameters $k$ and $d$, which, for optimal results, are set at $k = \log n$ and $d = \log n / \log\log n$. The input to the function can be thought of as a complete, oriented $k$-ary tree with depth $d$. Levels are named by their distance to the root. Each of the $k^d$ leaves is labelled with an input bit. Every internal node of the tree is labelled by a $2^k$ input bit string, which is thought of as explicitly describing a function from $\{0,1\}^k$ to $\{0,1\}$. This initial labelling induces a labelling of every node by a bit as follows. The induced bit of a leaf is just its original label. If an interior node $v$ is labelled by a function $f_v$, and its children, in order from left to right, have induced bits $b_1, \ldots, b_k$, then the induced bit for $v$ is $b_v = f_v(b_1, \ldots, b_k)$. The output of the Iterated Multiplexor is the induced bit of the root. The input to the function is of length $k^d + 2^k(\sum_{i=1}^{d} k^i) \in O(n^2)$. This function can easily be seen to be in $AC^1$.

If, for each node of the tree, the $2^k$ bits describing the function are fixed, then the Iterated Multiplexor becomes a composition of these functions. For example, consider the case $d = 3$, where the nodes at level $i$ are labelled by $f_i$. Then the restricted function has $k^3$ boolean inputs, which we can partition into $k^2$ vectors of length $k$, $\vec{x}_{i,j}$, $i,j \in [1\ldots k]$. We can write the restricted function as $f_0(f_1(f_2(\vec{x}_{11}), \ldots f_2(\vec{x}_{1k})), \ldots f_2(\vec{x}_{kk})))$. Intuitively, the circuit depth complexity of this function should be the sum of that for $f_0, f_1$, and $f_2$, since any circuit should have to compute at least some of the values for $f_2$ before going on to $f_1$, etc.

To prove an $\Omega(\log^2 n / \log\log n)$ depth lower bound for the Iterated Multiplexor function, it suffices to

*NSERC A9176 and NSERC Post Graduate Scholarship

prove that the circuit depth of the composition of functions is roughly the sum of the depths required for each of the functions involved: if the functions being composed are randomly chosen $k$ bit functions, then they require depth $k - o(k)$. Hence, if there are $d = \log n / \log\log n$ levels of these, then depth $dk \in \Omega(\log^2 n / \log\log n)$ is required.

Proving that circuit depth is additive under composition seems extremely difficult. Therefore, [2] suggested an abstraction of the communication game for the Composition of Functions. They call this game *the Universal Composition Relation*. In this game, the functions being composed are eliminated. It is called the Universal Composition Relation, because any algorithm for it will solve the game for any composition of functions. In addition, the known algorithms for the Compositions of Functions Game apply to it. Therefore, the hope is that a lower bound for *the Universal Composition Relation* problem would yield insight into the circuit depth of compositions.

Let us first consider $d = 1$. The real communication game (one multiplexor) given a fixed function $f$, is referred to as $R_f$. One player, called the A-Player is given a $k$ bit vector $x$ for which $f(x) = 0$, and the B-Player, a $y$ for which $f(y) = 1$. The object of the game is to find a bit position $i$ where the two differ, i.e., $x_i \neq y_i$. In the Universal Game, the two players are each given a $k$ bit vector, however, they are only promised that their vectors are different. The communication complexity of the game for a random function $f$ and that of the Universal Game are both $k - o(k)$. Basically, one player needs to communicate his vector to the other player. Thus, intuitively, being told that for your inputs $x$ and $y$, $f(x) \neq f(y)$ for a random function $f$ doesn't convey any more useful information than being told $x \neq y$. The Universal Composition Relation is obtained from the Iterated Multiplexor by applying this intuition in an analogous way.

The **Universal Composition Relation with parameters $d$ and $k$, UCR(d,k)** is a communication game defined as follows. The adversary gives each player a boolean labeling of all the nodes of an ordered $k$-ary tree of depth $d+1$. This tree is called the **index tree**. The label of the root for the A-Player must be 0 and for the B-Player must be 1. In addition, the adversary must promise that for every node of the tree, if it is labeled differently for the two players, then at least one of the children of the node is labeled differently. It follows that one of the leaves is labeled differently. The goal of the players is to agree on a leaf node which is labeled differently in their two trees.

The obvious communication algorithm for the Universal Composition Relation goes as follows. The algorithm is divided into $d$ rounds, each requiring $k$ bits of communication. Initially, the players agree that the root (level 0) has a different label. At the beginning of the $r^{th}$ round, the players agree on a node at level $r-1$ that has a different label. We will call labels of this node's children the **child vector** of the node. The players communicate $k$ bits to find a difference in these vectors. This begins the next round.

In this paper, we prove the following lower bound for the communication complexity of the Universal Composition Relation:

**Theorem 1** $CC(UCR(d,k)) \geq dk - O(d^2(k\log k)^{1/2})$

This bound is almost tight for $d = o(\sqrt{\frac{k}{log k}})$. If this same bound were proven for the Iterated Multiplexor Function, it would yield an $\Omega(\frac{\log^{1.5} n}{\log\log^{.5} n})$ bound on circuit depth. Recently, using different (and highly interesting) techniques, Håstad and Wigderson [1] obtained a lower bound of $dk - 2^{O(d)}$. This is better for small values of $d$.

The goal now, in order to separate $AC_1$ from $NC_1$ is to use these techniques to prove the lower bound for the Iterated Multiplexor function. This motivated us to reexamine the case for $d = 1$. We give a proof directly in terms of communication complexity that there is a function on $k$ bits requiring $\Omega(k)$ circuit depth, i.e. $CC(R_f) = \Omega(k)$. Although this fact can be easily established using a simple counting argument, our proof has the advantage of being an "adversary argument," which is more in line with the proof techniques used for the Universal Relation. We hope our results will provide a step towards a theory for circuit depth.

## 2 Predictability

Consider a set of vectors $S \subseteq X^\tau$ indexed by the set $\tau$, and whose elements lie in set $X$. For our purposes $X$ is the set of $k$-bit vectors $\{0,1\}^k$, making $v \in S$ a vector of vectors. $v$ can also be thought of as a matrix with the rows indexed by $\tau$ and the columns by $j \in [1 \ldots k]$.

For every vector $v \in S$ and index $i \in \tau$, let $v_i \in X$ be the element indexed by $i$ in the vector. Similarly, for every subset of indexes $\sigma \subseteq \tau$, let $v_\sigma \in X^\sigma$ be the sub-vector of $v$ indexed by $\sigma$. Let $S_\sigma = \{v_\sigma \mid v \in S\}$ be the projection of $S$ onto $\sigma$. For every $v_\sigma \in S_\sigma$ , let $S(v_\sigma) = \{v' \in S \mid v'_\sigma = v_\sigma\}$ be those vectors consistent with the $v$ on the elements indexed by $\sigma$.

Suppose $t$ bits have been communicated about an input in $X^t$ and let $S$ be the set of inputs consistent with this conversation. ($|S| = 2^{-t}|X^\tau|$). The question is, for $i \in \tau$, how many of these bits were communicated "about the $i^{th}$ element". In other words, if a random

vector $v$ is chosen from $S$, how much information is known about $v_i$, even if the other elements $v_{\tau-\{i\}}$ have been revealed? $S(v_{\tau-\{i\}})$ is the set of vectors $v$ that are consistent with the revealed information. Therefore, the probability of being able to predict the value of $v_i$ is $1/\left|S(v_{\tau-\{i\}})\right|$. Hence, we define the **predictability** of the $i^{th}$ element of $v \in S$ to be

$$Pred_i(S) = Exp_{v \in S}\left[\frac{1}{|S(v_{\tau-\{i\}})|}\right]$$

It might be useful to think of $S_{\tau-\{i\}}$ and $S_{\{i\}}$ as two sets of nodes in a bipartite graph and each $v = \langle v_{\tau-\{i\}}, v_i\rangle \in S$ as an edge. In this light, $S(v_{\tau-\{i\}})$ is the set of edges incident to the node $v_{\tau-\{i\}}$.

If the $i^{th}$ element $v_i$ of $v \in S$ is fixed as a function of the other elements, then $Pred_i(S) = 1$. If $t$ bits have been communicated about $v_i$, then $Pred_i(S) = \frac{2^t}{|X|}$. Finally, if this element is completely undetermined, then $Pred_i(S) = \frac{1}{|X|}$. The following lemma is an alternate definition of predictability. Its proof is simple manipulation of summations.

**Lemma 1** $Pred_i(S) = \frac{|S_{\tau-\{i\}}|}{|S|}$.

Another measure of the amount of information that has been communicated about $v_i$ is called the **collision probability** and is denoted $CP_i(S)$. It is the probability, for two vectors randomly chosen (without replacement) from $S$, that the $i^{th}$ element is the same, i.e $CP_i(S) = Prob_{v,v' \in S}[v_i = v'_i]$. The predictability of $i$ is an upper bound on this probability.

**Lemma 2** $CP_i(S) \leq Pred_i(S)$.

**Proof:** Let $v' \in S$ be the vector that maximizes $Prob_{v \in S}[v_i = v'_i]$. Then $CP_i(S) = Prob_{v,v' \in S}[v_i = v'_i] \leq Prob_{v \in S}[v_i = v'_i] = \frac{1}{|S|}\sum_{u_{\tau-\{i\}} \in S_{\tau-\{i\}}} \sum_{v \in S(u_{\tau-\{i\}})}$ [1 if $v_i = v'_i$ else 0]. However, $\forall v \in S(u_{\tau-\{i\}})$ there is at most one $v$ which is consistent both with $u_{\tau-\{i\}}$ and with $v'_i$. Therefore, $CP_i(S) \leq \frac{1}{|S|}\sum_{u_{\tau-\{i\}} \in S_{\tau-\{i\}}} 1 = \frac{|S_{\tau-\{i\}}|}{|S|}$ ■

How does throwing away vectors from $S$ effect the predictability?

**Lemma 3** *Let* $S' \subseteq S$. *Then* $Pred_i(S') \leq \frac{|S|}{|S'|}Pred_i(S)$.

**Proof:** Note that $S'_{\tau-\{i\}} \subseteq S_{\tau-\{i\}}$. Therefore, $Pred_i(S') = \frac{|S'_{\tau-\{i\}}|}{|S|} \leq \frac{|S_{\tau-\{i\}}|}{|S'|} = \frac{|S|}{|S'|}\frac{|S_{\tau-\{i\}}|}{|S|} = \frac{|S|}{|S'|}Pred_i(S)$ ■

Suppose $t$ bits have been communicated about the vectors $v$ in $S$, i.e. $|S| = 2^{-t}|X^\tau|$. A natural property to want is that, for all $l$, at most $l$ elements of $v$ can have more than $\frac{t}{l}$ bits "revealed about it". The problem is that after only $\log(|X|)$ bits have been communicated, it is possible that $\forall i \in \tau$, $Pred_i(S) = 1$, implying that $\log(|X|)$ bits have been communicated "about each of the elements". Consider the following example, for $X = \{0,1\}^k$. Suppose that for $j \in [1 \ldots k]$, the $j^{th}$ bit of communication reveals that the parity of the $j^{th}$ column is 0. Then for each $i \in \tau$, the row $v_i$ is uniquely determined by the other rows $v_{\tau-\{i\}}$. Therefore, $\forall i \in \tau,\ Pred_i(S) = 1$.

When an element becomes highly predictable in $S$, this element is fixed as a function of the other elements and then ignored. From then on, the set of vectors $v_{\tau-\{i\}} \in S_{\tau-\{i\}}$ is considered. In our previous example, if any one of the rows $v_i$ is fixed as a function of the other rows, then no information at all is known about the remaining elements $v_{\tau-\{i\}}$. The $v_i$ value is fixed as a function of $v_{\tau-\{i\}}$, by choosing one vector $v \in S(v_{\tau-\{i\}})$ for each $v_{\tau-\{i\}} \in S_{\tau-\{i\}}$. Let $S' \subseteq S$ the the set of these vectors. This gives a natural 1-1 mapping between $S'$ and $S_{\tau-\{i\}}$. The following lemma says that if at most $t$ bits have been revealed about $S$, then there exists a set of at most $l$ elements such that, if we fixed them in this way, then no more than $\frac{t}{l}$ bits have been "revealed about" any of the other elements.

**Lemma 4** *Let* $|S| \geq 2^{-t}|X^\tau|$ *and* $|\tau| \geq l > 0$. *Then there exists a subset of elements* $\sigma \subseteq \tau$ *such that* $|\sigma| \leq l$ *and* $\forall i \in \tau - \sigma,\ Pred_i(S_{\tau-\sigma}) < \frac{2^{\frac{t}{l}}}{|X|}$.

**Proof:** Initially, let $\sigma = \emptyset$. We will keep adding indices to $\sigma$, maintaining the property that $\frac{|S_{\tau-\sigma}|}{|X^{\tau-\sigma}|} \geq 2^{-t\left(1-\frac{|\sigma|}{l}\right)}$. Note that, when $\sigma = \emptyset$, this property is true because $|S| \geq 2^{-t}|X^\tau|$. Now assume that $\sigma \subseteq \tau$, that $\frac{|S_{\tau-\sigma}|}{|X^{\tau-\sigma}|} \geq 2^{-t\left(1-\frac{|\sigma|}{l}\right)}$, and that there is an index $i \in \tau - \sigma$ for which $Pred_i(S_{\tau-\sigma}) \geq \frac{2^{\frac{t}{l}}}{|X|}$. By Lemma 1, $\frac{|S_{\tau-\sigma-\{i\}}|}{|S_{\tau-\sigma}|} \geq \frac{2^{\frac{t}{l}}}{|X|}$. It follows that $\frac{|S_{\tau-\sigma-\{i\}}|}{|X^{\tau-\sigma-\{i\}}|} \geq \frac{2^{\frac{t}{l}}}{|X|}\frac{|S_{\tau-\sigma}|}{|X^{\tau-\sigma-\{i\}}|} = 2^{\frac{t}{l}}\frac{|S_{\tau-\sigma}|}{|X^{\tau-\sigma}|} \geq 2^{\frac{t}{l}}\,2^{-t\left(1-\frac{|\sigma|}{l}\right)}$ $= 2^{-t\left(1-\frac{|\sigma\cup\{i\}|}{l}\right)}$. Thus the property holds for $\sigma \cup \{i\}$.

Eventually, $\forall i \in \tau - \sigma,\ Pred_i(S_{\tau-\sigma}) < \frac{2^{\frac{t}{l}}}{|X|}$. Since $S_{\tau-\sigma} \subseteq X^{\tau-\sigma}$, it follows that $1 \geq \frac{|S_{\tau-\sigma}|}{|X^{\tau-\sigma}|} \geq 2^{-t\left(1-\frac{|\sigma|}{l}\right)}$ and thus $|\sigma| \leq l$. ■

At certain points in the argument, we will want to ensure that the players are given vectors that are distinct at many locations. We will do this by partitioning $X$ for each $i \in \tau$ with a random function $g_i : X \rightarrow \{0,1\}$, and promising that, for every $i \in \tau$, $g_i(v^A{}_i) = 0$ and $g_i(v^B{}_i) = 1$, where $v^A$ and $v^B$ are the inputs for the A and the B-Player.

**Lemma 5** *Suppose* $\forall i \in \tau\ Pred_i(S) \leq \frac{1}{4|\tau|m}$. *For each* $i \in \tau$, *pick uniformly at random the functions* $g_i : X \rightarrow \{0,1\}$. *Then the Prob*[ $\exists v \in S$ *such that* $\forall i \in \tau,\ g_i(v_i) = 0] \geq 1 - \frac{|\tau|}{2^{m+1}}$.

**Proof:** Let $S^0 = S$ and let $S^i = \{v \in S^{i-1} \,|\, |S^{i-1}(v_{\tau-\{i\}})| \geq m\}$, for $i \in [1 \ldots |\tau|]$. First we will prove that $|S^i| \geq (1 - \frac{i}{2|\tau|})|S|$ by induction. For $i = 0$, it is trivial. Let $1 \leq i \leq |\tau|$ and assume that $|S^{i-1}| \geq \frac{1}{2}|S|$. Thus, by Lemma 3, $Pred_i\left(S^{i-1}\right) \leq \frac{1}{2|\tau|m}$. Because $Pred_i(S) = Exp\left[1/|S(v_{\tau-\{i\}})|\right] \geq Prob[|S(v_i)| \leq m] \times \frac{1}{m}$, it follows that $|S(v_{\tau-\{i\}})| > m$ with probability of at least $1 - Pred_i(S)m$, when $v$ is chosen uniformly from $S$. Therefore, $v \in S^i$ with probability at least $1 - \frac{1}{2|\tau|}$ for $v$ chosen uniformly from $S^{i-1}$. Hence $|S^i| \geq (1 - \frac{1}{2|\tau|})|S^{i-1}| \geq (1 - \frac{1}{2|\tau|})(1 - \frac{i-1}{2|\tau|})|S| > (1 - \frac{i}{2|\tau|})|S|$. Thus $S^{|\tau|}$ is nonempty.

Now we will construct a sequence of vectors $v^{(|\tau|)}, \ldots, v^{(0)}$ as follows: Let $v^{(|\tau|)} \in S^{|\tau|}$. For $i = |\tau|, \ldots, 1$, select $v^{(i-1)} \in S^{i-1}(v^{(i)}_{\tau-\{i\}})$ such that $g_i(v_i^{(i-1)}) = 0$, if possible. Let $v = v^{(0)}$. Since $v^{(i)} \in S^i$, $|S^{i-1}(v_i^{(i)})| \geq m + 1$ and the vectors in it have distinct $i^{th}$ entries. So the probability that $g_i$ will be 1 on all of them is at most $\frac{1}{2^{m+1}}$. Therefore, $v^{(0)}$ exists with probability at least $1 - \frac{|\tau|}{2^{m+1}}$, and $g_i(v^{(0)}) = 0,\ \forall i \in \tau$ by construction. ■

## 3 The Depth Two Case ($d = 2$)

The proof for general $d$ involves a great deal of complicated notation and messy details, which tends to obscure the basic intuition. Therefore, in this section, we will present the proof for the special case of $d = 2$.

Each player for the $d = 2$ game is given a $\{0,1\}$ labeling L of the complete $k$-ary tree of depth 3. Let $\tau_0$ be the root of the tree, $\tau_1$ be the $k$ children of the root and $\tau_2$ be the $k^2$ leaves. For such a labeling L, let $L_1$ be the induced labeling on $\tau_1$ and $L_2$ on $\tau_2$. Since the input to the A-Player always labels the root 0 and 1 to the B-Player, we can ignore the labeling of the root. The bit labeling $L_2$ can either be thought of as $k^2$ bits indexed by $\tau_2$ or as $k$ vectors indexed by $\tau_1$. For node $i \in \tau_1$, we call the vector labeling the children of $i$ the **child vector of** $i$.

### 3.1 Intuition

Before presenting a formal proof, let us develop some intuition. We would like to show that any reasonable protocol first communicates enough bits to find a difference in the players' $L_1$ vectors and then finds a difference in the child vector of this differently labeled node. This will take twice as many communication bits as to find a difference in two distinct $k$ bit vectors. This game is called the **Universal Game**. A simple argument that the Universal Game requires $k - 2$ bits is as follows.

Each player is given a $k$ bit vector, with the only requirement being that they are different. They must find a index where the vectors are different. The lower bound will be proved by a simple adversary argument. Given a fixed protocol that is assumed to solve the game in fewer time steps, we give a procedure for an adversary to find an input pair on which the protocol fails. The adversary restricts the set of inputs each round in order to fix the conversation between the players up to the current time step. After a certain number of bits have been communicated, let $S^A$ be the set of vectors $x$ that can be given to the A-Player and be consistent with the conversation so far. Define $S^B$ similarly. The adversary will maximize $\left|S^A \cap S^B\right|$ in order to maintain the symmetry between the players. A bit communicated by the A-Player partitions $S^A$ into those vectors on which he communicates a 0 and those on which a 1 is communicated. Thus the adversary chooses the half that keeps the intersection $S^A \cap S^B$ as large as possible. This at most halves the intersection. Similarly, for bits communicated by the B-Player. After $t$ bits, $\left|S^A \cap S^B\right| \geq 2^{k-t}$. Hence, after $k - 2$ bits $\left|S^A \cap S^B\right|$ contains at least three vectors. If the protocol ends at this point and outputs a bit position $i$, then the adversary can find at least two of the three vectors which are the same at this index. These two vectors are given to the two players. Therefore, the protocol fails on this input.

This adversary strategy can be summarized by saying that the adversary delays breaking the symmetry between the players for as long as possible, in order to make it difficult for the players to find a difference. At the last minute, this symmetry has to be broken in order to keep the promise of giving them different vectors.

The same tension between making the players' inputs the same and making them different occurs in our lower bound. The lower bound is broken into stages, one stage per level. In the first stage, the adversary maintains the symmetry between the players' values for both $L_1$ and $L_2$. Just before the players can communicate the $k$ bits describing $L_1$, their $L_1$ vectors are fixed to be different. In the second stage, the same is done with $L_2$. The symmetry is kept until just before the players can communicate one child vector in $L_2$.

The lower bound is more complicated than above, because the players may communicate bits about $L_2$ during the first stage of the protocol; hybrid bits in-

volving both the $L_1$ and $L_2$ parts of the input; or bits which involve many places in the $L_2$ vector. For simplicity, we will ignore the last two complications and come back to them later.

The intuition is that bits communicated about $L_2$ during the first stage of the protocol are wasted. The players do not know which of $k^2$ bits in $L_2$ to communicate. On the other hand, if the players wait until a difference in $L_1$ is known, then they can concentrate on finding a difference in the corresponding child vector within $L_2$.

Suppose, pessimistically, that the players communicate $k$ bits about $L_2$ during the first stage. They could either communicate a lot of bits about a few child vectors or a few bits about lots of child vectors. Either way this is not useful for the players. Assume lots of bits have been communicated about a few child vectors. Remember that the adversary only needs to make these child vectors different for the two players if the corresponding bits in $L_1$ are different. Thus, the adversary will simply make the corresponding bits in $L_1$ the same, freeing her to fix the entire child vectors in the same way for both players. Thus, the many bits communicated about $L_2$ could have been replaced by only a few bits about $L_1$. On the other hand, if only a few bits about any particular child vectors are known, these bits will not significantly help the players once a difference in the $L_1$'s is discovered, and the players focus on one child vector. The dividing line between lots of bits communicated and a few will be set to $\frac{k}{l}$ bits. ($l$ is a parameter which will later be set to $\sqrt{k}$ in order to optimize the lower bound.) Note there are at most $l$ child vectors for which at least $\frac{k}{l}$ bits have been communicated.

Therefore, the general strategy for the adversary is as follows. During the first $k - l - f$ bits of communication, where $f$ is a small "fudge factor", we assume pessimistically that the players learn $k - l - f$ bits about both $L_1$ and $L_2$. After this period, the adversary performs a "clean-up" which further restricts the allowable input pairs. The adversary first identifies at most $l$ child vectors for which the players know at least $k/l$ bits, and fixes the corresponding bits in $L_1$. This leaves only $f$ bits of $L_1$ unknown; the adversary must now break the symmetry at the $L_1$ level. If no hybrid bits concerning both $L_1$ and $L_2$ have been communicated, this is simple: the adversary picks a different $L_1$ vector for each player from the $2^f$ possible vectors. In the real proof, this will be slightly more complicated. Once this symmetry is broken, the parties may know many places in which their $L_1$'s differ, and hence know that the corresponding child vectors are different. But they will not know more than $k/l$ bits about any one of these child vectors. Thus, the adversary will be able to force the protocol to go for about $k - k/l$ more steps before any of these rows are in danger of being forced to be the same for both players. This yields a lower bound of $2k - O(l + k/l) = 2k - O(k^{1/2})$ bits for any protocol solving the game.

The players have more options than mentioned above. Although it does not appear to help them, instead of communicating a bit about $L_1$ or a bit about $L_2$, they could communicate a bit about the combination. Our lower bound gets around this complication by (more or less) converting one combination bit into one bit about $L_1$ and one bit about $L_2$. Another complication is that, if the players communicate "hybrid" bits, the number of bits known about a child vector is not well-defined. For this purpose, we will use the concept of "predictability" developed in Section 2.

## 3.2 Formal Proof for $d = 2$

### 3.2.1 A Set Of $L_1$ For Each $L_2$

After a certain number of bits have been communicated, let $S^A$ be the set of $\langle L_1, L_2 \rangle$ pairs that can be given to the A-Player and be consistent with the conversation so far. Define $S^B$ similarly. During the first stage, the adversary will maximize $\left|S^A \cap S^B\right|$ in order to maintain the symmetry between the players. More precisely, we will define a set $S \subseteq S^A \cap S^B$. The adversary would like to concentrate separately on $L_1$ vectors during the first stage of the protocol and on the $L_2$ vectors during the second stage. However, because a player is able to communicate a hybrid bit combining information about $L_1$ and about $L_2$, which $L_1$ vectors can be given to a player depends on which $L_2$ vector he is given. Therefore, consider $S$ organized as follows. Let $T$ be the set of $L_2$ vectors that can be given to either player. For each $L_2 \in T$, let $S(L_2) = \{L_1 \mid \langle L_1, L_2 \rangle \in S\}$, be the set of $L_1$ vectors such that $\langle L_1, L_2 \rangle$ can be given to either player. In this way, we can view $S$ as a tree of depth 2. The root of the tree is the set $T$ of $L_2$ vectors and for each $L_2 \in T$, there is a leaf node labeled with $S(L_2)$. We define $S^A_{\tau_2}$,$S^B_{\tau_2}$, $S^A(L_2)$, and $S^B(L_2)$ analogously. When a player communicates a hybrid bit combining information about $L_1$ and about $L_2$, the adversary (more or less) converts this into one bit about $L_1$ and one bit about $L_2$. This is done as follows. The adversary wants to ensure that after $t$ bits of communication, at most $t$ bits have been "communicated" about $L_2$ and at most $t$ bits about $L_1$. Therefore, she maintains the inequalities: $|T| \geq 2^{k^2-t}$; and, for each $L_2 \in T$, $|S(L_2)| \geq 2^{k-t}$.

When a player communicates a bit, it partitions $S$.

This can also be viewed as partitioning $S(L_2)$ for each vector $L_2$. For each $L_2$ the adversary determines which part of the partition is largest (i.e. size of a least $2^{k-(t+1)}$) and restricts $S(L_2)$ to this part. In this way, each $L_2$ vector votes whether it is better to force the player to communicate a 0 or a 1. This vote partitions the set $T$. The adversary restricts $T$ to the larger of the two parts. This is at least of size $2^{k^2-(t+1)}$. This maintains the induction hypotheses. This process continues for $t = k - l - \log(2k^2)$ bits of communication. At this point, $|T| \geq 2^{k^2-k}$ and for each $L_2 \in S$, $|S(L_2)| \geq 2^{l+\log(2k^2)}$.

### 3.2.2 Cleaning Up Between Stages

Between the stages, the adversary does some "cleaning up". The child vectors of $L_2$ for which more than $\frac{k}{l}$ bits have been communicated are fixed as functions of the remaining child vectors and the corresponding bits of $L_1$ are fixed to be the same for both players.

Think of each $L_2$ as a vector of length $k$ indexed by $\tau_1$ composed of child vectors. For $i \in \tau_1$, consider the $i^{th}$ child vector of $L_2$. $Pred_i(T)$ will be used to measure the number of bits that has been communicated "about this child vector". See Section 2.

Lemma 4 says that if $|T| \geq 2^{k^2-k}$ (i.e. no more than $k$ bits have been communicated about $L_2$), then there exists a subset of child vectors $\sigma \subseteq \tau_1$, $|\sigma| \leq l$, such that if these are ignored, then the remaining child vectors $i \in \tau_1 - \sigma$ have predictability $Pred_i(T_{\tau_1-\sigma}) \leq \frac{2^{\frac{k}{l}}}{2^k}$ (i.e. no more than $\frac{k}{l}$ bits have been communicated "about each of these child vectors"). $T_{\tau_1-\sigma}$ is defined to be the set of possible values for the remaining child vectors. For each $L_2' \in T_{\tau_1-\sigma}$, we arbitrarily select a full $L_2 \in T$ extending $L_2'$. We denote $L_2$ with $E(L_2')$. Let $T' = \{E(L_2) \mid L_2 \in T_{\tau_1-\sigma}\}$. Note that $T_{\tau_1-\sigma} = T'_{\tau_1-\sigma}$.

In order to keep her promise, the adversary must fix (in a symmetric way) the bits of $L_1$ corresponding to the fixed child vectors in $\sigma$. This is handled as if one of the players communicates the bits in positions $\sigma$ of $L_1$. See Section 3.2.1 to see how this is done. Initially, $|S(L_2)| \geq 2^{l+\log(2k^2)}$, for each $L_2 \in T'$. Therefore, after fixing all $l$ bits in $\sigma$, each of these sets still has at least size $2k^2$. Similarly, $T'$ has been restricted to a subset of $2^{-l}$ the size. Lemma 3 that, if a set is cut down by a factor of $2^l$, than the predictability of any child vector can go up by at most the same factor. Therefore, $\forall i \in \tau_1 - \sigma$, $Pred_i(T_{\tau_1-\sigma}) \leq \frac{2^{\frac{k}{l}+l}}{2^k}$ (i.e. no more than $\frac{k}{l} + l$ bits have been communicated "about each of these child vectors").

The adversary has also promised, that because the A-Player's root and the B-Player's root are labeled differently, their $L_1$ vectors will be different. Thus, the adversary must now break the symmetry between the two players' possible values for $L_1$. For each $L_2 \in T'$, one $L_1$ vector is chosen for the A-Player from the remaining set $S(L_2)$ and one is chosen for the B-Player. In this way, the $L_2$ chosen for each player at the end of the protocol will fix the $L_1$ vectors for the two players in an anti-symmetric way as a function of the $L_2$. In order to ensure that the players get different $L_1$ vectors special care needs to be taken. The adversary chooses a function $G : \{0,1\}^k \Rightarrow \{0,1\}$ randomly. The A-Player will be given an $L_1$ from $G^{-1}(0)$ and the B-Player from $G^{-1}(1)$. This works fine as long as for each $L_2 \in T'$, $S(L_2) \cap G^{-1}(0)$ and $S(L_2) \cap G^{-1}(1)$ contain at least one vector each. With high probability, this is the case. For a given $L_2$, $Prob_G[S(L_2) \cap G^{-1}(0) = \emptyset] = 2^{-|S(L_2)|}$. Therefore, $Prob_G[\exists L_2 \in T'$ such that $S(L_2) \cap G^{-1}(0) = \emptyset$ or $S(L_2) \cap G^{-1}(1) = \emptyset] \leq 2|T'|/2^{|S(L_2)|} \leq 2\ 2^{k^2}/2^{2k^2} << 1$. (In the general case, $d > 2$, we will need to use Lemma 5 to handle this step.)

### 3.2.3 The Second Stage

During the second stage of the lower bound the players continue to communicate bits. Since, for each player, we have fixed $L_1$ as a function of $L_2$, each bit communicated partitions the set $T'$ of possible $L_2$. Since we have fixed the child vectors indexed by $\sigma$ as a function of $L_2' \in T'_{\tau_1-\sigma}$ (i.e. $T'$ and $T'_{\tau_1-\sigma}$ are in 1-1 correspondence via $E$), this is the same as partitioning $T'_{\tau_1-\sigma}$.

The adversary chooses the larger half. After $t$ bits of communication, the set has been reduced by a factor of $2^t$ and hence, by Lemma 4, the predictability of any child vector has gone up by at most the same factor. At the beginning of the second stage, $\forall i \in \tau_1 - \sigma$, $Pred_i(T'_{\tau_1-\sigma}) \leq \frac{2^{\frac{k}{l}+l}}{2^k}$. This stage continues for $k - (\frac{k}{l} + l) - \log(3k)$ bits of communication. Therefore, in the end, each child vector has predictability of no more than $\frac{1}{3k}$.

### 3.2.4 Choosing The Inputs

Suppose that the protocol ends at this point and outputs a bit position from $\tau_2$. The adversary must find a $\langle L_1, L_2 \rangle$ pair for each of the players for which all the promises are met and for which the specified bit is labeled the same. The adversary chooses these inputs by choosing two $L_2'$ vectors uniformly at randomly from the remaining set $T'_{\tau_1-\sigma}$, one for each player. This chooses the child vectors indexed by $\tau_1 - \sigma$. The child vectors indexed by $\sigma$ are fixed in a symmetric way as a function of these and the $L_1$ vectors are fixed in an anti-symmetric way. The claim is that with probabil-

ity greater than $\frac{1}{6}$, these inputs meet that adversary's requirements.

First let us check that the promises are met. One promise is that if the root is different (which it is) then the $L_1$ vectors must be different. The adversary has already ensured this: the A-Player's $L_1$ is from $G^{-1}(0)$ and the B-Player's is from $G^{-1}(1)$. The other promise is that if a bit of $L_1$ is different then the corresponding child vectors of $L_2$ must be different. The adversary has ensured that the bits of $L_1$ indexed by $\sigma$ are the same. Therefore, the promise is here is met. The remaining bits of $L_1$ might be different. Therefore, for each $i \in \tau_1 - \sigma$, the $i^th$ vector of $L_2$ must be different for the two players. Lemma 2 states that the probability, for two $L_2$ vectors randomly chosen (without replacement) from $T'_{\tau_1-\sigma}$, that the $i^{th}$ child vector is the same is no more than $Pred_i(T'_{\tau_1-\sigma})$. The adversary has ensured that each of these predictabilities is no more than $\frac{1}{3k}$. The probability that one of the $k$ such child vectors is the same is, hence, no more than 1/3. All the promises are met with probability 2/3

Now let us calculate the probabilities that the specified index of $\tau_2$ is labeled the same. Let $p$ be the probabilities that the specified index is labeled 1. Because for both players $L_2$ is chosen uniformly from $T'$, this probability is the same for both players. Therefore the probability that the label is the different is $p(1-p) + (1-p)p \leq 1/2$. The total probability of a bad choice is at most $1/3 + 1/2 < 1$. Therefore, there exists a choice of inputs which causes the protocol to fail.

The first stage lower bound was $k - 2l$ bits and the second stage was $k - (\frac{k}{l} + l) - \log(3k)$. $l = \sqrt{k}$. Therefore, for $d = 2$, the lower bound is $2k - 4\sqrt{k} - \log(3k)$. This generalizes to the lower bound $dk - O(d^2 k^{1/2} \log k)$.

## 4 Proof for $d \geq 3$

We can now prove our main theorem for this section:
**Theorem** 1 $CC(UCR(d,k)) \geq dk - O(d^2(k \log k)^{1/2})$.
**Proof of Theorem 1:** Our proof is a complicated induction in terms of the stages of the protocol, where we try to force the algorithm to spend almost $k$ bits of communication discussing each level of the input tree. Before we begin the proof, we develop some notation. For $d \geq j \geq 0$, let $\tau_j$ represent the nodes of the complete depth $d$ $k$-ary tree at depth $j$ and let $\tau_{\geq j} = \cup_{j \leq j' \leq d} \ \tau_{j'}$. We can view a labelling $L_j$ of $\tau_j$ either as a sequence of bits indexed by $\tau_j$ or as a sequence of $k$-bit vectors indexed by $\tau_{j-1}$. We call the latter the "child vector" representation of $L_j$. In general, we will reserve $S$ to denote a set of labellings, but use $T$ to denote the same set in child vector representation.

For a set $S_{\geq j}$ of labellings of $\tau_{\geq j}$, and $d \geq j' \geq j$, let $S_{\geq j'} = (S_{\geq j})_{\tau_{\geq j'}}$. For a labelling $L_{\geq j'+1}$ of $\tau_{\geq j'}$, we let $S_{j'}(L_{\geq j'+1})$ be the set of labellings of $\tau'_j$ which extend $L_{\geq j'+1}$ to an element of $S_{\geq j'}$. (For $j' = d$, we think of $\tau_{d+1}$ as the empty set of nodes, $L_{\geq d+1}$ as the only labelling of this set, so $S_d(L_{d+1}) = S_{\geq d}$). We use $T_{j'}(L_{\geq j'+1})$ to represent the same set in child vector form.

We can think of $S_{\geq j}$ as being presented in the form of a tree of depth $d - j + 1$, called the hierarchy tree. A path through the tree gradually defines an element of $S_{\geq j}$, starting at a labelling of $\tau_d$, next a labelling of $\tau_{d-1}$, etc. The root (depth 0) is labelled by the set $S_{\geq d} = S_d(L_{\geq d+1})$; the nodes at depth $d - j', d \geq j' \geq j$, by the sets $S_{j'}(L_{\geq j'+1}), L_{\geq j'+1} \in S_{j'+1}$; and the leaves by the elements of $S_{\geq j}$. The children of a node correspond to the elements of its label: the children of a node labelled $S_{j'}(L_{\geq j'+1})$ are labelled by the sets $S_{j'-1}(L_{\geq j'})$ where $L_{\geq j'} \in S_{\geq j'}$ extends $L_{\geq j'+1}$; unless $j' = j$, in which case the children are the elements of $S_{\geq j}$ extending $L_{\geq j+1}$, i.e., $L_{\geq j+1}$ together with an element of $S_j(L_{\geq j+1})$.

Let $L^A, L^B$ be labellings of $\tau = \tau_{\geq 0}$. We say the pair is *as promised up to level r* if the roots are labelled as appropriate for players $A$ and $B$ respectively, and if, for every $r - 1 \geq j \geq 0$ and every $i \in \tau_j$, if $L^A$ and $L^B$ differ in the bit labelling $i$, then they differ in the child vector indexed by $i$.

Let $l = \sqrt{k/\log k}, s = k - 3dk/l - 2l - 5dl \log k = k - O(d(k \log k)^{1/2})$. Fix any protocol which supposedly solves $UCR(d,k)$ using at most $sd$ bits of communication. We call the time between the $(r-1)s + 1$'th and $rs$'th bits communicated the $r$'th stage of the protocol.

We will prove, by induction on $r \in [1 \ldots d]$, that there is:

1. a conversation $C_{(r-1)s}$ for the first $(r-1)s$ bits, which induces consistent sets of inputs $S^A$ and $S^B$ for the $A$ and $B$-players respectively;
2. a set $S_{\geq r}$ of labellings of $\tau_{\geq r}$;
3. extension functions $E^A, E^B$ mapping a labelling $L_{\geq r}$ to a complete input $L_{\geq 0}$ extending $L_{\geq r}$, so that $E^A(S_{\geq r}) \subseteq S^A$ and $E^B(S_{\geq r}) \subseteq S^B$; and
4. $\sigma \subset \tau_{r-1}$, $|\sigma| \leq l$

which satisfy:

1. For $d + 1 \geq j \geq r + 1$, $\forall L_{\geq j+1} \in S_{\geq j+1}$, $|S_j(L_{\geq j+1})| \geq 2^{|t_j| - (r-1)k}$.
2. For j=r, $\forall L_{\geq r+1} \in S_{\geq r+1}$, let $T = T_r(L_{\geq r+1})$, then $\forall i \in \tau_{r-1} - \sigma$, $Pred_i(T_{\tau_{r-1}-\sigma}) \leq 2^{-s - 3d \log k - l - dl \log k}$.

3. Let $T$ be as in the previous item. Then the projection function $P$ given by $P(\mathrm{L}_r) = (\mathrm{L}_r)_{\tau_{r-1}-\sigma}$ is a bijection between $T$ and $T_{\tau_r-\sigma}$ , i.e., the values of the child vectors indexed by $\sigma$ have been fixed as a function of the other child vectors.

4. $\forall \mathrm{L}_{\geq r}, \mathrm{L}'_{\geq r} \in S_{\geq r}$, $E^A(\mathrm{L}_{\geq r})$ and $E^B(\mathrm{L}'_{\geq r})$ are as promised up to level $r$.

5. $\forall i \in \sigma$, $\exists b_i \in \{0,1\}$ so that $\forall \mathrm{L}_{\geq r} \in S_{\geq r}$, $E^A(\mathrm{L}_{\geq r})(i) = E^B(\mathrm{L}_{\geq r})(i) = b_i$. (I.e., the labels for nodes in $\sigma$ have been fixed to the same constants for both players.)

For $r = 1$, $S_{\geq 1}$ is the set of all labellings of all nodes except the root of the index tree; $C_0 = \epsilon$; $\sigma = \emptyset$; and $E^A$, $E^B$ extend an element of $S$ by labelling the root with a 0 or 1, respectively. All conditions above are vacuously satisfied.

The following lemma says, intuitively, that revealing a bit about the entire labelling is no worse than revealing one bit about the labelling at each level.

**Lemma 6** *Let $H$ be a finite rooted tree, and let $L_1, \ldots, L_p$ be a partition of the leaves of $H$. Then there is a non-empty sub-tree $H'$ of $H$ and a $1 \leq b \leq k$ so that all leaves of $H'$ are in $L_b$, and so that, for every interior node $h \in H'$, if $h$ has $C$ children in $H$, $h$ has at least $C/p$ children in $H'$.*

The proof is trivial and omitted. Applying the lemma to the hierarchy tree, we see, that for any partition of $S_{\geq r}$ into $p$ pieces, one piece has a subset, (call it $S'_{\geq r}$), which satisfies: for every $d \geq j \geq r$, and every $\mathrm{L}_{\geq j+1} \in S'_{\geq j+1}$, $|S'_j(\mathrm{L}_{\geq j+1})| \geq \frac{1}{p}|S_j(\mathrm{L}_{\geq j+1})|$. Now, each bit communicated by player $A$ in round $r$ creates a partition of $S_{\geq r}$ into 2 pieces, the side of the partition for $\mathrm{L}_{\geq r}$ being given by the bit communicated if $A$ has input $E^A(\mathrm{L}_{\geq r})$. Similarly for $B$. For each such bit, the adversary will restrict $S_{\geq r}$ to the subset $S'_{\geq r}$ guaranteed to exist by the lemma, and add the corresponding communication bit to the partial conversation. At the end of $s$ such restrictions, our new set (which, for convenience, we shall still refer to as $S'_{\geq r}$) will satisfy: for $d+1 \geq j \geq r$, $\forall \mathrm{L}_{\geq j+1} \in S'_{\geq j+1}$, $|S'_j(\mathrm{L}_{\geq j+1})| \geq \frac{1}{2^s}|S_j(\mathrm{L}_{\geq j+1})|$. In particular, for $j = r$, for any $\mathrm{L}_{\geq r+1} \in S'_{r+1}$, let $T = T_r(\mathrm{L}_{\geq r+1})$ and $T' = T'_r(\mathrm{L}_{\geq r+1})$. Then we have $|T'_{\tau_r-\sigma}| = |T'| \geq 2^{-s}|T| = 2^{-s}|T_{\tau_r-\sigma}|$; hence, by Lemma 2, $\forall i \in \tau_r - \sigma, Pred_i(T'_{\tau_r-\sigma}) \leq 2^s Pred_i(T_{\tau_r-\sigma}) \leq 2^s 2^{-s-3d\log k-l} = 2^{-3d\log k-l}$.

The adversary then performs a clean-up operation along the lines of the $d = 2$ case. The first task is to use Lemma 4 to find a small set of child vectors of nodes at level $r$ whose removal makes the other child vectors unpredictable. However, here this set might be different for different $T' = T'_{r+1}(\mathrm{L}_{\geq r+2})$. For each such $T'$, by Lemma 4 since $|T'| \geq 2^{|\tau_{r+1}|-rk-s} \geq 2^{-dk}(2^k)^{|\tau_r|}$, there is a set $\sigma_{T'} \subset \tau_r$, $|\sigma_{T'}| \leq l$, so that $\forall i \in \tau_r - \sigma_{T'}, Pred_i(T'_{\tau_r-\sigma_{T'}}) \leq 2^{dk/l}2^{-k}$. We need to pick a single such set $\sigma'$. If $r = d-2$, there is only one set $T'$, so the choice of $\sigma'$ is $\sigma_{T'}$. Otherwise, note that there are at most $p = |t_r|^l = (k^r)^l \leq k^{dl}$ choices for $\sigma_{T'}$. These different choices partition $S'_{\geq r}$; in fact, they partition the tree obtained from the hierarchy tree when all leaves and neighbors of leaves are removed, i.e., the hierarchy tree for $S'_{\geq r+2}$. By applying Lemma 6 to this tree, we obtain $\sigma'$ and $S''_{\geq r+2} \subseteq S'_{\geq r+2}$ whose extensions to elements of $S'_r$ form a subset $S''_{\geq r}$ so that: for $d \geq j \geq r+2$ and $\mathrm{L}_{\geq j+1} \in S''_{j+1}$, $|S''_j(\mathrm{L}_{\geq j+1})| \geq 2^{-dl\log k}|S'_j(\mathrm{L}_{\geq j+1})|$; for $j = r$ or $r+1$, $S''_j(\mathrm{L}_{\geq j+1}) = S'(\mathrm{L}_{\geq j+1})$; and for any $\mathrm{L}_{\geq r+2} \in S''_{\geq r+2}$, for $T'' = T''_{r+1}(\mathrm{L}_{\geq r+2})$, $s_{T''} = \sigma'$.

Next, the adversary must fix all the bits labelling nodes of $\sigma'$. This is done again using Lemma 6, obtaining $S'''_r$ so that $|S'''_j(\mathrm{L}_{\geq j+1})| \geq 2^{-l}|S''_j(\mathrm{L}_{\geq j+1})|$ for all $\mathrm{L}_{\geq j+1} \in S'''_{\geq j+1}$.

The adversary will now ensure that any input pair is as promised up to level $r$. Because they are already as promised up to level $r-1$ and because the bits labelled by $\sigma$ have been fixed to be the same for both players, it will suffice to ensure that the child vectors indexed by $\tau_{r-1}-\sigma$ are labelled differently for the two players. She does this by picking a random function $g_i$ from $\{0,1\}^k$ to $\{0,1\}$ for each $i \in \tau_{r-1} - \sigma$. Let $T''' = T'''_r(\mathrm{L}_{\geq r+1})$ for any $\mathrm{L}_{\geq r+1} \in S'''_{\geq r+1}$. By combining inequalities and equalities above, we get: $|T'''| \geq 2^{-l}|T'_r(\mathrm{L}_{\geq r+1})|$. Thus, $\forall i \in \tau_{r-1} - \sigma, Pred_i(T'''_{\tau_{r-1}-\sigma}) \leq 2^l Pred_i(T'_{\tau_{r-1}-\sigma}) \leq k^{-3d}$. Therefore, from Lemma 5, we have, with probability at least $1 - 2^{-k^{2d}/4}$, there is an element $\mathrm{L}^A$ of $T'''_{\tau_{r-1}-\sigma}$, and hence of $T'''$, so that $g_i(\mathrm{L}^A{}_i) = 0$ for all $i \in \tau_{r-1} - \sigma$, where $\mathrm{L}^A{}_i$ is the child vector of $i$ in $\mathrm{L}^A$, and similarly, we can pick an $\mathrm{L}^B$ mapping to all 1's. Since there are at most $2^{k^{d+1}}$ such $T'''$, for some choice of $g_i$'s, there will be such elements for all $T'''$. Call an element mapping to all 0's $A$-good, and one mapping to all 1's, $B$-good.

The adversary now fixes, for each $T''' = T'''_{r+1}(\mathrm{L}_{r+2}), \mathrm{L}_{r+2} \in S'''_{r+2}$, the values of the child vectors of $\sigma'$ as a function of the other child vectors. In other words, for each $\mathrm{L}'_{r+1} \in (T''')_{\tau_r-\sigma'}j$, the adversary selects a unique $\mathrm{L}_{r+1} \in T'''$ extending $\mathrm{L}'_{r+1}$. The adversary throws all other extensions away, obtaining a subset $S''''_{\geq r+1}$ of $S'''_{\geq r+1}$.

If we then let our new $S_{\geq r+1} = S''''_{\geq r+1}$, we can then let our new extension functions $E^A$ and $E^B$, on input $\mathrm{L}_{\geq r+1}$ , first select an $A$-good (respectively, $B$-good)

element from $T'''_r(\mathrm{L}_{\geq r+1})$ and then apply the old extension function to the result. Finally, our new value for $\sigma$ is $\sigma'$. We must check that, $\forall \mathrm{L}_{\geq r+2} \in S'''_{\geq r+2}$, for $T'''' = T''''_{r+1}(\mathrm{L}_{\geq r+2})$, then $\forall i \in \tau_r$, $Pred_i(T''''_{\tau_r - \sigma'}) \leq 2^{-s-l-3d\log k}$. But it follows from Lemma 2 and the above inequalities that this last is at most $2^{-k+dk/l+l}$, and so this follows from our choice of $s$. Finally, for all $\mathrm{L}_{\geq j+1}$ for $d \geq j \geq r+1$, we have $|S'''_j(\mathrm{L}_{\geq j+1})| \geq 2^{|t_j|-(r-1)k-s-dl\log k-l} \geq 2^{|t_j|-rk}$. Thus, we have proved the claim for $r$.

Applying the claim for $r = d-1$, we get a set $S_{\geq d} = T$ of labellings of $\tau_d$, $\sigma \subset \tau_{d-1}$, and extension functions $E^A$ and $E^B$ so that any pair of elements of $S$ extend to an input pair consistent with both the conversation and all promises with the possible exception of those for the child vectors of $\tau_{d-1} - \sigma$. Moreover, $T$ and $T_{\tau_{d-1}-\sigma}$ are in 1-1 correspondence, and $\forall i \in \tau_{d-1} - \sigma, Pred_i(T) \leq 2^{-s-3d\log k}$. Thus, the adversary can go for $s$ more communication bits (picking the larger half of the induced split in $T$), and still maintain: $\forall i \in \tau_{d-1} - \sigma, Pred_i(T) \leq 2^{-3d\log k}$. At this point, the protocol has run for $ds$ time steps, and must output. With high probability, a randomly selected pair from $T$ differ in all child vectors indexed by $\tau_{d-1} - \sigma$, and so extend to inputs keeping all promises. On the other hand, the inputs agree at the place output by the protocol with probability at least 1/2. Therefore, the adversary can find a pair of inputs for which the protocol fails. This contradiction proves the theorem. ■

## 5 The Multiplexor Game, $d = 1$

**Def$^n$:** The $v_0$-$v_1$-$f$ **Communication Game** is played on a fixed set of vectors $U \subseteq \{0,1\}^k$. An adversary chooses two vectors $v_0, v_1 \in V$ and one function $f \in \{0,1\}^U$, with the restriction that $f(v_0) = 0$ and $f(v_1) = 1$. Player $P_0$ is given $(v_0, f)$ and player $P_1$ is given $(v_1, f)$. The players' goal is to find an index $i \in [1,k]$ for which the vectors are different.

**Theorem 2** *The $v_0$-$v_1$-$f$ Communication Game requires $\Omega(\log|U|)$ bits of communication. For example, for $U = \{0,1\}^k$, $\Omega(k)$ bits are needed.*

Note that Theorem 2 can be proved by a simple argument. Riordan and Shannon in 1942 [5] prove there are many more functions in $\{0,1\}^{\{0,1\}^k}$ than there are circuits with depth $O(k)$. Since $CC(R_f) = Depth(f)$, it follows that there are functions with high communication complexity. The adversary must only give the players such a function and the lower bound follows. It is not difficult to see that the converse is also true; a lower bound for the $v_0$-$v_1$-$f$ Communication Game yields a matching bound for $R_f$ for some $f$. The counting argument of [5] is notorious for yielding no intuition for constructing hard functions. Hopefully, our new techniques will provide new insight especially when combined with other communication complexity type proofs. The new proof finds a smaller version of the $v_0$, $v_0$-$v_1$-$f$ Communication Game, after each bit of communication.

**Proof:** Given a fixed communication protocol, the adversary goes through the protocol round by round maintaining a set of vectors $V \subseteq \{0,1\}^k$ and a set of functions $F \subseteq \{0,1\}^{\{0,1\}^k}$ from which $v_1$, and $f$ are chosen. After $t-1$ rounds of communication, the adversary ensures that the sets $V$ and $F$ have the following three properties.

1. $\exists a_1, a_2, \ldots, a_{t-1} \in \{0,1\}$ $\forall v_0, v_1 \in V$, $\forall f \in F$, if $f(v_0) = 0$ and $f(v_1) = 1$, then $a_1, a_2, \ldots, a_{t-1}$ are the first $t-1$ bits communicated by the protocol given $(v_0, v_1, f)$.
2. $|V| \geq \frac{|U|}{c^{t-1}}$ ($c = 12.27$ works.)
3. $F_V = \{0,1\}^V$

$F_V \subseteq \{0,1\}^V$ is defined to be the projection of $F$ on $V$. It is the set of functions mapping vectors in $V$ to $\{0,1\}$ each of which is consistent with some function in $F$. Because we are no longer considering the vectors not in $V$, we do not care what $f$ evaluates to on these. Property 3 states that all functions defined on $V$ are possible. In fact, the proof is easier if we assume that for every function in $F_V$, there is one and only one function consistent with it in $F$. In this case, $|F| = 2^{|V|}$.

By property 2, after no more than $t = \frac{k-2}{\log c}$ bits of communication, $V$ still contains at least three vectors. At this point, the communication game is over and the players must specify an index $\in [1,k]$ for which the vectors $v_0$ and $v_1$ are different. At least two of the three vectors in $V$ have the same bit at the specified index. One of these two vectors is given to player $P_0$ and the other to $P_1$. By property 3, there exists a function $f \in F$ for which $f(v_0) = 0$ and $f(v_1) = 1$. In fact, the property says something much stronger, namely that $\forall \vec{x} \in \{0,1\}^V$, $\exists f \in F$ such that $\forall v \in V$, $f(v) = \vec{x}_v$. This function is given to both players. By Property 1, the protocol's communication pattern on $(v_0, v_1, f)$ is as stated. It follows that the protocol fails to find a difference in the vectors given these inputs.

Clearly, when $t = 1$, $V = U$, $F = \{0,1\}^U$, and the three properties hold. Assume that the three properties are met for time $t-1$ and that, given the conversation $a_1, a_2, \ldots, a_{t-1}$, player $P_\epsilon$ communicates during round

$t$. For every pair $(v, f)$ for which $f(v) = \epsilon$, the protocol states what bit $P_\epsilon$ will communicate if given the pair. We will denote this bit with $b^{v,f}$. The rules of the game ensure that $P_\epsilon$ will never have a pair $(v, f)$ for which $f(v) \neq \epsilon$. Therefore, the protocol does not specify what bit $P_\epsilon$ will communicate when given this pair. In this case, let $b^{v,f} = *$. The adversary is able to set this bit to 0 or 1 as needed. With this notation, the statement "if $f(v) = \epsilon$, then player $P_\epsilon$ communicates the bit $a$ given the pair $(v, f)$ for round $t$", becomes simply $[b^{v,f} \in \{a, *\}]$. The adversary then proceeds in three steps.

1. First, she finds $F' \subseteq F$ of size $(3/2)^{|V|}$ such that $\forall v \in V,\ \exists a^v \in \{0,1\},\ \forall f \in F',\ [b^{v,f} \in \{a^v, *\}]$.
2. Next, she finds $V' \subseteq V$ of size $\frac{2|V|}{c}$ such that $F'_{V'} = \{0,1\}^{V'}$.
3. Finally, she finds $V'' \subseteq V'$ of size $\frac{|V|}{2}$ such that $\exists a \in \{0,1\},\ \forall v \in V'',\ \forall f \in F',\ [b^{v,f} \in \{a, *\}]$.

It should be clear that, after these steps, $V''$ and $F'$ satisfy the three properties given. First, we will give the intuition for how step 1 is accomplished and then we will give its proof. The goal of step 1 is to find a subset $F'$ of the functions for which $b^{v,f}$ is constant with respect to $f$ for each $v$. Consider any $v \in V$. This vector partitions $F$ into 3 parts: those $f$'s for which $b^{v,f}$ equals 0, 1, and *, i.e 0 is communicated, 1 is communicated, and $f(v) \neq \epsilon$. The adversary is free to choose between the 0 and the 1 part, by setting $a^v$ appropriately and, either way, is able to keep the entire * part. The * part (i.e. those $f \in F$ for which $f(v) \neq \epsilon$) likely makes up half of $F$ and the adversary is able to choose the larger of the 0 and the 1 parts; hence the goal can be met for this $v$, by restricting $F$ by at most a factor of 3/4. Initially, $F$ is of size $2^{|V|}$. Therefore, we expect $|F'|$ to be of size $2^{|V|}\,(3/4)^{|V|} = (3/2)^{|V|}$.

The vector $\vec{a} = a^{v_1} \ldots a^{v_{|V|}} \in \{0,1\}^V$ is said to be consistent with the function $f \in F$, if $\forall v \in V,\ [b^{v,f} \in \{a^v, *\}]$. In other words, $\vec{a}$ and $f$ are consistent, if $\vec{a}$ specifies how $P_\epsilon$ will communicate during time step $t$ as a function of which vector $v$ he is given, when given $f$. Therefore, in step 1, the adversary must find a $\vec{a}$ which is consistent with $(3/2)^{|V|}$ functions in $F$. Towards this goal, she constructs a $2^V \times |F|$ Boolean matrix $M$. Each row is labeled by an $\vec{a} \in \{0,1\}^V$ and each column is labelled with a function $f \in F$. The entry $M_{\vec{a},f}$ is set to be 1 if and only if $\vec{a}$ and $f$ are consistent.

For a given $f \in F$, let $l \in [0, n]$ be the number of $v \in V$ for which $f(v) \neq \epsilon$, for which $b^{v,f} = *$, and for which $a^v$ does not have to equal $b^{v,f}$. The number of $\vec{a}$ consistent with $f$ is then $2^l$. This is also the number of 1's in the column labeled by $f$. Because $F_V = \{0,1\}^V$, there are $\binom{|V|}{l}$ functions $f \in F$ with the number of $v \in V$ for which $f(v) \neq \epsilon$ being $l$. It follows that the total number of 1's in the matrix is $\sum_{l=0}^{|V|} \binom{|V|}{l} 2^l = (1+2)^{|V|} = 3^{|V|}$.

There are $2^{|V|}$ rows. Therefore, there exists an $\vec{a}$ for which its row contains $\frac{3^{|V|}}{2^{|V|}} = \left(\frac{3}{2}\right)^{|V|}$ 1's. Let $F'$ be those $f \in F$ that have a 1 in this row. It follows that $\forall v \in V,\ \exists a^v \in \{0,1\},\ \forall f \in F',\ [b^{v,f} \in \{a^v, *\}]$. This completes step 1.

**Lemma 7 ([Sauer, [4]])** *If the size of a family of $n$-vectors is bigger than the sum of the first $d$ binomial coefficients then there are $d$ coordinates on which the projection is the whole $d$-cube.*

Thus $|F'| \geq (3/2)^{|V|} \geq \sum_{l=0}^{\frac{2|V|}{c}} \binom{|V|}{l}$, for some suitable constant $c$. ($c = 12.27$ works.) Then, by Lemma 7, we know that step 2 can be accomplished.

Step 3 is very straightforward. From step 1, the bit communicated by $P_\epsilon$ does not depend on the function $f \in F'$, but does depend on $v$. Label each $v \in V'$ with the bit $a^v$ in the vector $\vec{a}$ chosen above. Partition $V'$ according to this labelling and let $V''$ be the larger half. It follows that $\exists a_t \in \{0,1\},\ \forall v_\epsilon \in V'',\ \forall f \in F'$, if $f(v_\epsilon) = \epsilon$ then player $P_\epsilon$ communicates the bit $a_t$ given $(v_\epsilon, f)$ for round $t$. This gives us Property 1. ■

## Acknowledgments

We would like to thank Toni Pitassi, Dan Simon, and Faith Fich for their insightful conversations and support.

## References

[1] J. Håstad, A. Wigderson, "Composition of the Universal Relation," *manuscript*, 1991.

[2] M. Karchmer, R. Raz, A. Wigderson, "On Proving Super-logarithmic Depth Lower Bounds via the Direct Sum in Communication Complexity," *Proceedings of $6^{th}$ Structures in Complexity Theory*, pp. 299-304, 1991.

[3] M. Karchmer, A. Wigderson, "Monotone circuits for connectivity require super-logarithmic depth," *Proceedings of $20^{th}$ STOC*, pp. 539-550, 1988.

[4] N. Sauer, "On the Density of Families of Sets," *J. Combinatorial Theory (A) 13*, pp. 145-147, 1972.

[5] J. Riordan, C. E. Shannon, "The Number of Two-Terminal Series-Parallel Networks," *J. Math. Phys., Vol 21*, pp. 83-93, 1942.