# Separation of the Monotone NC Hierarchy

Ran Raz *
Weizmann Institute
Rehovot, Israel

Pierre McKenzie †
Université de Montréal
Montréal, Canada

March 25, 1998

**Abstract**

We prove tight lower bounds, of up to $n^\epsilon$, for the monotone depth of functions in monotone-P. As a result we achieve the separation of the following classes.

1. monotone-NC $\neq$ monotone-P.
2. For every $i \geq 1$, monotone-$NC^i \neq$ monotone-$NC^{i+1}$.
3. More generally: For any integer function $D(n)$, up to $n^\epsilon$ (for some $\epsilon > 0$), we give an explicit example of a monotone Boolean function, that can be computed by polynomial size monotone Boolean circuits of depth $D(n)$, but that cannot be computed by **any** (fan-in 2) monotone Boolean circuits of depth less than $Const \cdot D(n)$ (for some constant $Const$).

Only a separation of monotone-$NC^1$ from monotone-$NC^2$ was previously known.

Our argument is more general: we define a new class of communication complexity search problems, referred to below as DART games, and we prove a tight lower bound for the communication complexity of every member of this class. As a result we get lower bounds for the monotone depth of many functions. In particular, we get the following bounds:

1. For $st$-connectivity, we get a tight lower bound of $\Omega(\log^2 n)$. That is, we get a new proof for Karchmer-Wigderson's theorem, as an immediate corollary of our general result.
2. For the $k$-clique function, with $k \leq n^\epsilon$, we get a tight lower bound of $\Omega(k \log n)$. This lower bound was previously known for $k \leq \log n$ [AlBo87]. For larger $k$, however, only a bound of $\Omega(k)$ was previously known.

*Department of Applied Mathematics and Computer Science, Weizmann Institute, Rehovot, 76100 Israel. Work supported by an American-Israeli BSF grant 95-00238. Email: `ranraz@wisdom.weizmann.ac.il`

†Département d'informatique et recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal (Québec), H3C 3J7 Canada. Work supported by the NSERC of Canada and by the FCAR du Québec. Email: `mckenzie@iro.umontreal.ca`

## 1 Introduction

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if flipping a bit from 0 to 1 in any argument to $f$ cannot cause the value of $f$ to change from 1 to 0. A *monotone Boolean circuit* is an indegree-two single-output circuit over the monotone base $\{\wedge, \vee\}$. The *size* of a circuit is the number of gates in the circuit, and the *depth* of a circuit is the length of the longest path between a circuit input and the circuit output. The *monotone size* of a function is defined to be the smallest size of a monotone circuit for that function, and the *monotone depth* of the function is defined to be the smallest depth of a monotone circuit for that function.

In his breakthrough paper in 1985, Razborov [Ra85a] proved a super-polynomial lower bound for the monotone-size of the Clique function, and as a conclusion obtained the separation of monotone-P from monotone-NP. Using Razborov's technique, exponential lower bounds for the monotone size of other functions were proved by Andreev [An85], and an exponential lower bound for the monotone size of the Clique function was finally proved by Alon and Boppana [AlBo87]. A simpler proof for that lower bound was recently presented by Haken [Ha95].

Those lower bounds, and other lower bounds for the monotone size of functions, immediately translate into corresponding lower bounds (of up to $n^\epsilon$) for the monotone depth of the same functions. Lower bounds for the size, however, cannot give the separation of classes of monotone depth (e.g., the monotone-NC hierarchy), as those classes are sub-classes of monotone-P. Thus, in order to achieve a separation of those classes, one needs to prove lower bounds for the monotone depth of functions in monotone-P. Hence, in order to achieve a separation of classes of monotone depth, one needs to prove depth lower bounds directly, and not as a consequence of size lower bounds.

In 1988, Karchmer and Wigderson [KaWi88] obtained the important result that the monotone depth of the $st$-connectivity function is $\Omega(\log^2 n)$. Since $st$-connectivity is in monotone-$NC^2$, the separation of monotone-$NC^1$ from monotone-$NC^2$ was obtained. Since then, however, no better lower bounds for the monotone depth of functions in monotone-P were proved, and no larger gaps between the monotone depth of a function and the logarithm of its monotone size were obtained. Other proofs for monotone-$NC^1 \neq$ monotone-$NC^2$ were later presented in [GrSi92] (where a separation of monotone-L from monotone-$NC^1$ was also proved), and in [KaRaWi91].

Some other direct lower bounds for the monotone depth of functions are known. In particular, a tight lower bound of $\Omega(n)$ was proved for the monotone depth of the Matching function [RaWi90]. The Matching function, however, is not in monotone-P, as a super-polynomial lower bound for its monotone size was proved in [Ra85b]. If sub-exponential upper bounds for the monotone size of the Matching function were shown then the result of [RaWi90] would have given a separation of classes of monotone depth (by a padding argument). It is still open, however, whether such an upper bound exists.

For more information about the early results in monotone complexity see the excellent survey of [BoSi90].

In this paper, we prove tight lower bounds of up to $n^\epsilon$, for the monotone depth of functions in monotone-P. In particular, for $D(n) = n^\epsilon$ (for some constant $\epsilon$), we give an explicit example of a function in monotone-P that can be (uniformly) computed by a family of monotone Boolean circuits of polynomial size and of depth $D(n)$, but that cannot be computed by any family of monotone Boolean circuits of depth less than $Const \cdot D(n)$ (for some constant $Const$). By a padding argument, the same result follows for any function $D(n) \le n^\epsilon$ as well. Hence, the following corollaries follow immediately:

1. monotone-NC $\neq$ monotone-P.
2. For every $i \ge 1$, monotone-$NC^i$ $\neq$ monotone-$NC^{i+1}$.

## 1.1 Relevance of Monotone Complexity

Monotone complexity has always attracted many researchers. Since the result of [Ra85a], many papers on monotone complexity have appeared, and this includes many interesting papers in the last 3 years (e.g., [Ya94, GoHå95, Ha95, AmMa96, BeUl97, SiTs97, Ju97]). Is monotone complexity research useful ?

Although the separation results of [Ra85a], and [KaWi88], are among the most famous and most impressive results in complexity theory, it is still under debate whether monotone complexity is worth pursuing.

Indeed, by the results of [Ra85b, Ta88], the monotone size of a function may be exponentially larger than its non-monotone size. By the result of [RaWi90], the monotone depth of a function may be exponentially larger than its non-monotone depth. By the results of [Ra89, RaRu93, Ra94], some of the techniques used so far to obtain lower bounds for monotone circuits are not strong enough to obtain non-monotone separations such as P $\neq$ NP. It is therefore widely accepted that the known lower bounds for monotone complexity are only a very small step towards a separation of non-monotone classes.

On the other hand, monotone complexity is relevant for non-monotone complexity, at the very least because a separation theorem for non-monotone complexity classes (e.g., NC $\neq$ P) automatically gives the separation of the corresponding monotone classes as well. Therefore, if one is not able to separate monotone-NC from monotone-P then one is not able to separate NC from P either. Furthermore, although the known techniques for proving lower bounds for monotone complexity are not very likely to give significant lower bounds for non-monotone complexity, it is not unlikely that these techniques will be combined with some new techniques to obtain non-monotone lower bounds, or that monotone complexity will affect non-monotone complexity in some other way.

In addition, monotone complexity is also interesting in its own right. Indeed, determining the monotone size (or depth) of a function is a very natural combinatorial problem, and monotone complexity may be relevant for several other complexity issues. One important example is propositional proof theory, where following

[Ra94] and Bonet et. al [BoPiRa95], reductions to monotone complexity were extensively used. In particular, using techniques developed in the sequence of papers [ImPiUr94, BoPiRa95, Kr95], Pudlak [Pu95] used monotone complexity to obtain an impressive exponential lower bound for the length of cutting planes proofs (see also, [CoHa95, Fu96]). Other applications of monotone complexity are also known.

### 1.2 Methods and Other Results

We use Karchmer and Wigderson's communication complexity approach [KaWi88] (see also [Ka88]). In this approach, a lower bound for the monotone depth of a function $f$ is obtained by proving a lower bound for the complexity of the following communication game: Player I is given an input $x$, such that $f(x) = 1$. Player II is given an input $y$, such that $f(y) = 0$. The goal of the two players is to find a coordinate $i$ such that $x_i = 1$, and $y_i = 0$.

Our proof begins by defining a new class of communication games, which we call *dart* games. Briefly stated, a dart game is a game of the following type: Player I is given $x_1, \ldots, x_n$, where for every $i$, $x_i \in \{1, \ldots, m\}$. Player II is given $y_1, \ldots, y_n$, where for every $i$, $y_i$ is a coloring of $\{1, \ldots, m\}$. The goal of the two players is to solve a DNF search problem $R$, depending only on $e_1, \ldots, e_n$, where $e_i$ is the color of $x_i$ in the coloring $y_i$.

A *structured* communication protocol for a dart game is (briefly stated) one where the players reveal the variables $e_i$ one by one, that is, in each round Player I sends $x_i$ (for some $i$) and Player II answers with $y_i$. Our main theorem shows that if $m$ is much larger than $n$ (say $m \geq n^{20}$) then any communication protocol for a dart game can be simulated by a structured protocol of the same complexity (up to a multiplicative constant). Since structured protocols are usually very easy to analyze, this gives a general tight lower bound for the communication complexity of every dart game. It turns out that this lower bound implies lower bounds for the communication complexity of many monotone Karchmer-Wigderson's games, and hence gives lower bounds for the monotone depth of many functions.

The separation of the monotone NC hierarchy is then obtained by proving a lower bound for a variant, called *GEN* (see [JoLa77]), of the monotone P-complete problem *Path Systems* (see [Co74]). As mentioned above, our argument is general enough to prove lower bounds for many other functions. In particular, we get a new proof for Karchmer-Wigderson's $\Omega(\log^2 n)$ lower bound for $st$-connectivity, on a graph with $n$ vertices, and a new (tight) lower bound of $\Omega(k \cdot \log n)$ for the monotone depth of the $k$-Clique function for small cliques ($k \leq n^{\epsilon}$).

### 1.3 Sections Description

In Section 2 we formally define DART games, and we state our main theorem. In Section 3 and Section 4 we apply the main theorem to derive our lower bounds for the monotone depth of functions. In Section 5 we present some of the information theoretic tools, used in the proof for the main theorem. In Section 6 we present the complete proof of the main theorem.

## 2 Communication Complexity and DART Games

We consider the standard 2-party Communication Complexity model of Yao [Ya79]. For an excellent survey of communication complexity see [KuNi96].

Let $X$, $Y$, $Z$ be finite sets, and let $R \subseteq X \times Y \times Z$. For two subsets $A \subseteq X, B \subseteq Y$, a *communication protocol $P$ for $R$ over the domain $A \times B$* specifies, for each $(x, y) \in A \times B$, the exchange of information bits by two players, Player I and Player II, that initially receive as inputs $x$ and $y$ respectively, and finally agree on a value $P(x, y) \in Z$ such that $(x, y, P(x, y)) \in R$.

The *communication complexity of such a protocol $P$* is the maximum, over all $(x, y) \in A \times B$, of the number of bits exchanged by the two players on the input pair $(x, y)$ when using $P$. The *communication complexity $C_R(A, B)$ of $R$ over the domain $A \times B$* is the minimum, over all protocols $P$ for $R$ over $A \times B$, of the complexity of $P$. Finally, the *communication complexity of the relation $R$* is $C_R(X, Y)$, which will also be denoted $CC(R)$.

We think of $R$ also as a function from the domain $X \times Y \times Z$ to the range $\{TRUE, FALSE\}$, where $R(x, y, z) = TRUE$ iff $(x, y, z) \in R$.

### 2.1 DART Games

Denote by $[m]$ the set $\{1, \ldots, m\}$. For every $n, m \in \mathbb{N}$, we define a class of communication games $DART(m, n)$. A communication game, given by the relation $R \subseteq X \times Y \times Z$, is in $DART(m, n)$ if the following holds:

1. $X = [m]^n$. In other words, the input for Player I is a sequence of numbers $x = (x_1, x_2, \ldots, x_n)$, with $x_j \in [m]$ for every $j$.

2. $Y = (\{0, 1\}^m)^n$. In other words, the input for Player II is a sequence $y = (y_1, y_2, \ldots, y_n)$ of binary colorings of $[m]$, that is, each $y_j$ is an $m$-bit string. We think of $y_j$ also as a function $y_j : [m] \to \{0, 1\}$.

3. The relation $R(x, y, z)$ depends only on the sequence $(y_1(x_1), y_2(x_2), \ldots, y_n(x_n))$, and on $z$, (where $y_j(x_j)$ denotes the $x_j$-th bit in the string $y_j$). In other words, if $x, x' \in X$ and $y, y' \in Y$ satisfy that for every $j$, $y_j(x_j) = y'_j(x'_j)$ then for every $z$, $R(x, y, z) = R(x', y', z)$.

   Hence, $R(x, y, z)$ can be described as $R((e_1, \ldots, e_n), z)$, where $e_j \stackrel{\text{def}}{=} y_j(x_j)$.

4. $R((e_1, \ldots, e_n), z)$ is a DNF-Search-Problem, that is, there exists a DNF tautology $F_R(e_1, \ldots, e_n)$, with set of clauses $Z$, such that $R((e_1, \ldots, e_n), z) = TRUE$ iff $z$ is a satisfied clause of $F_R(e_1, \ldots, e_n)$.

For a relation $R$, we will say that $R$ is a dart relation, and use the notation $R \in DART(m, n)$ if the corresponding game is in $DART(m, n)$.

## 2.2 Structured Protocols and General Protocols for DART Games

A *structured* communication protocol for a dart game is a protocol of the following type: In each round of the protocol, Player I sends the value of $x_j$ for some index $j$, and Player II answers with $y_j(x_j)$. Thus in one round the players exchange $\lceil \log_2 m \rceil + 1$ bits of information, and find out the value of one $y_j(x_j)$. [1] A structured protocol can also be described as a decision tree for the corresponding DNF-search-problem, over the variables $(e_1, \ldots, e_n)$ (for the exact definition see [LoNeNaWi95]).

For a dart relation $R$, denote by $SC(R)$ (that is, the *Structured Complexity* of $R$) the number of rounds in the shortest structured protocol that solves $R$. Note, that if the number of rounds in a structured protocol is $k$ then the communication complexity is $k \cdot (\lceil \log_2 m \rceil + 1)$. Recall that the communication complexity of the best general protocol for the relation $R$ is denoted by $CC(R)$.

Thus, structured communication protocols for dart games are very limited. In each round, each player is allowed to give information on only one variable. It is, therefore, not very surprising that for many interesting relations, it is very easy to determine $SC(R)$ exactly. It turns out, however, that in many interesting cases general communication protocols for dart games can be simulated by structured ones ! In these cases, a lower bound for the structured complexity of a relation (i.e., $SC(R)$) gives a lower bound for the general communication complexity (i.e., $CC(R)$) as well.

Our main theorem shows that if $m$ is larger than some polynomial in $n$ ($m \geq n^{20}$) then structured protocols for $DART(m,n)$ games are as powerful (up to a multiplicative constant) as general protocols. The constant 20 here is not optimal. Obtaining the best possible constant is not the focus of this paper.

**Theorem 2.1** *Assume that $m \geq n^{20}$, and let $R \subseteq X \times Y \times Z$ be a relation in $DART(m,n)$. Then*

$$CC(R) = SC(R) \cdot \Omega(\log m).$$

(Recall that $SC(R)$ denotes the number of rounds in the shortest structured protocol for $R$, and not the communication complexity of that protocol).

## 2.3 Multi-Color DART Games

So far, we have defined dart games using colorings with two colors only. This is done for simplicity, and because for most applications two colors are enough. The main theorem, however, is true when one allows up to $m^\delta$ colors (for some small constant $\delta$). Let us therefore generalize the definition of dart games to the case of $r$ colors.

For every $n, m, r \in \mathbb{N}$, let us define the class $DART_r(m,n)$. A communication game, given by the relation $R \subseteq X \times Y \times Z$, is in $DART_r(m,n)$ if the following holds:

[1] W.l.o.g. it can be assumed that both players know the index $j$, and therefore $j$ does not have to be transmitted. Also, w.l.o.g. it can be assumed that the protocol depends only on the values of $y_j(x_j)$-s, transmitted in previous rounds, and not on the entire communication.

1. $X = [m]^n$.

2. $Y = (\{1, \ldots, r\}^m)^n$. In other words, the input for Player II is a sequence $(y_1, y_2, \ldots, y_n)$ of colorings of $[m]$ with $r$ colors. We think of $y_j$ also as a function $y_j : [m] \to [r]$.

3. The relation $R(x, y, z)$ depends only on the sequence $(y_1(x_1), y_2(x_2), \ldots, y_n(x_n))$, and on $z$. Hence, $R(x, y, z)$ can be described as $R((e_1, \ldots, e_n), z)$, where $e_j \stackrel{\text{def}}{=} y_j(x_j)$.

4. $R((e_1, \ldots, e_n), z)$ is a DNF-Search-Problem in $\{(e_i = j)\}_{i \in [n], j \in [r]}$.

   That is, there exists a tautology $F_R$, such that $F_R$ is a disjunction of conjunctions of expressions of the form $(e_i = j)$, and such that $Z$ is the set of clauses of $F_R$, and such that $R((e_1, \ldots, e_n), z) = TRUE$ iff $z$ is a satisfied clause of $F_R$.

As before, a structured communication protocol for $R$ is a protocol of the following type: In each round of the protocol, Player I sends the value of $x_j$ for some index $j$, and Player II answers with $y_j(x_j)$. Thus in one round, the players exchange $\lceil \log_2 m \rceil + \lceil \log_2 r \rceil$ bits of information, and find out the value of one $y_j(x_j)$. As before, denote by $SC(R)$ the number of rounds in the shortest structured protocol that solves $R$.

The following theorem is a generalization of Theorem 2.1 to the case of $r$ colors, where $r \leq m^\delta$ (for some small constant $\delta > 0$). For simplicity, we take $\delta = 1/1000$, which is not optimal.

**Theorem 2.2** *Assume that $m \geq n^{20}$, and $m \geq r^{1000}$, and let $R \subseteq X \times Y \times Z$ be a relation in $DART_r(m, n)$. Then*

$$CC(R) = SC(R) \cdot \Omega(\log m).$$

# 3 Separation of the Monotone-NC Hierarchy

In this section, we use Theorem 2.1 to prove the separation of the monotone-NC hierarchy. First, let us recall the connection between communication complexity and monotone depth : For a monotone Boolean function $f : \{0,1\}^l \to \{0,1\}$, define the relation $R_f$ by

$$R_f = \{\ (x, y, i) \in\ f^{-1}(1) \times f^{-1}(0) \times [l] \quad | \quad x_i = 1 \text{ and } y_i = 0\}.$$

The communication game played on $R_f$ is therefore the following : Player I gets an $l$-bit string, on which $f$ evaluates to 1. Player II gets an $l$-bit string, on which $f$ evaluates to 0. Their goal is to agree on a bit position $i$, in which player I's string has a 1, and player II's string has a 0. Below, we will refer to that game as the monotone KW-game for the function $f$.

The following observation was discovered by Yannakakis (unpublished), and by Karchmer and Wigderson who realized its full potential. (Here we only need the monotone form) :

**Lemma 3.1** *[KaWi88]* $CC(R_f)$ *is equal to the monotone depth of* $f$.

### 3.1 The *GEN* Function

The first insights leading to our separation results came from choosing a convenient function capturing the difficulty of the class monotone-P. Let us describe a variant of the very first P-complete function known [Co74], which Cook called: *Path Systems.* In this paper we call this function *GEN* (for "GENeration"), in analogy with Jones and Laaser's non-monotone version of the function [JoLa77] (see also [BaMc91]):

> **The *GEN* function:**
> The input for *GEN* is a string of $l^3$ bits $(t_{ijk})_{1\leq i,j,k\leq l}$. For $1 \leq k \leq l$, we say that 1 *generates* $k$ if $k = 1$, or for some $i$ and $j$ such that $t_{ijk} = 1$, 1 generates $i$ and 1 generates $j$ (where "1 generates $i$" and "1 generates $j$" are defined recursively in the same way).
> The function *GEN* determines whether 1, called the source, generates $l$, called the target. That is, $GEN(t_{111}, \ldots, t_{lll}) = 1$ iff 1 generates $l$.

In this context, we will refer to the set $\{1, 2, \ldots, l\}$ as the set of *GEN*-elements. We will sometimes write $i * j \rightarrow k$ rather than $t_{ijk} = 1$.

It is not hard to verify that *GEN* is a monotone Boolean function, computable by a monotone polynomial size circuit family. Our main goal here is to prove lower bounds for the monotone depth of *GEN*, as well as for some variations of it.

### 3.2 The *PYRGEN* game

Let $n = \binom{d+1}{2}$. We will now define the communication game $PYRGEN(m, d)$, that will later be related to *GEN*, and to some variations of it. The game $PYRGEN(m, d)$ will be in the class $DART(m, n)$.

Recall that in a $DART(m, n)$ game, Player I receives a sequence of $n$ integers in $[m]$, and Player II receives $n$ binary colorings of $[m]$. For the *PYRGEN* game, it is convenient to index each player's sequence by $(i, j)$, where $1 \leq j \leq i \leq d$, and to imagine that the sequence is laid out in a pyramidal fashion as described in Figure 1. Denote by $(x_{i,j})_{1\leq j\leq i\leq d}$ the sequence for Player I, and by $(y_{i,j})_{1\leq j\leq i\leq d}$ the sequence for Player II, and as before for every $i, j$, denote $e_{i,j} = y_{i,j}(x_{i,j})$.

The goal of the two players is to find $(i, j)$, such that one of the following is satisfied:

1. $i = 1, j = 1$, and $e_{i,j} = 0$, or
2. $i = d$, and $e_{i,j} = 1$, or

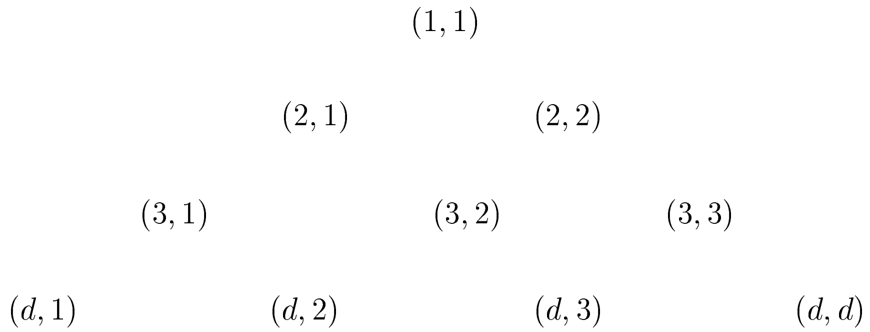


Figure 1: The pyramid structure for $d = 4$.

3. $i \leq d-1$, and $(e_{i,j} = 1) \wedge (e_{i+1,j} = 0) \wedge (e_{i+1,j+1} = 0)$.

In other words, the goal is to find either a position in the bottom of the pyramid, with $e_{i,j} = 1$, or a position at the top (note that there is only one such position) with $e_{i,j} = 0$, or a "pyramid-triangle" $(i,j), (i+1,j), (i+1,j+1)$, such that $(e_{i,j} = 1) \wedge (e_{i+1,j} = 0) \wedge (e_{i+1,j+1} = 0)$. It is not hard to verify that it is always possible to achieve one of these goals. In other words, the DNF-formula

$$\bigvee_{1 \leq j \leq i \leq d-1} [(e_{i,j} = 1) \wedge (e_{i+1,j} = 0) \wedge (e_{i+1,j+1} = 0)] \bigvee_j (e_{d,j} = 1) \bigvee (e_{1,1} = 0)$$

is a tautology.

### 3.3 The Communication Complexity of *PYRGEN*

We will now show that the structured complexity of $PYRGEN(m,d)$ is $\Theta(d)$. Both the upper bound and the lower bound are not hard. The upper bound is in fact trivial, by the following protocol:

First, the players ask for the value of $e_{1,1}$. If $e_{1,1} = 0$ then the game is over. Otherwise, the players "work their way down", in the following way: in each step they know for some $(i,j)$ that $e_{i,j} = 1$, and they ask for the values of $e_{i+1,j}$, and $e_{i+1,j+1}$. If both of these values are 0 then the game is over, otherwise, w.l.o.g. $e_{i+1,j} = 1$, and the players can continue with $e_{i+1,j}$. If the game continues until the players reach the bottom of the pyramid then when they reach the bottom they have $j$ such that $e_{d,j} = 1$.

This shows that $SC(PYRGEN(m,d)) \leq 2d-1$, and therefore that

$$CC(PYRGEN(m,d)) \leq (2d-1)(\log_2 m + 2).$$

On the other hand, we have:

**Lemma 3.2** *$SC(PYRGEN(m,d)) \geq d$.*

*Proof.* Assume for simplicity $d \geq 3$. Consider Player II as an adversary, who will attempt to make the game run for a long time. Hence, her input entries are not fixed in advance. We will imagine, in the course of a game, that a "vertex" $(i, j)$ receives the color *blue* if Player I learns or knows that $e_{i,j} = 0$, and that the vertex $(i, j)$ receives the color *red* if Player I learns or knows that $e_{i,j} = 1$ (vertices such that $e_{i,j}$ is unknown to Player I have no color). We will consider *legal* paths, that is, paths of length $d-1$, from the top vertex $(1,1)$ to some bottom vertex $(d, j)$, formed solely of arcs of the type $((i, j), (i+1, j))$ and of the type $((i, j), (i+1, j+1))$. We will say that a legal path is *good* if it contains no blue vertex. In the beginning of a game, no vertices are colored, and therefore all legal paths from the top to the bottom of the pyramid are good. The adversary's strategy is now defined in the following way:

> **Player II's adversarial strategy:**
> At any point during a game, she answers a question from Player I with 0, unless no good path would remain if she did so.

Consider any exchange of information between Player I and his adversary. At any point during such an exchange, at least one good path remains, and every remaining good path meets every red vertex (this is proved by induction on the number of red vertices). We claim that Player I cannot determine a correct answer for the game until he receives a 1 answer to a question $x_{d,j}$ (that is, an answer for a vertex at the bottom of the pyramid).

To prove this claim, suppose for the contrary that Player II's answer to some question $x_{i,j}$, with $i < d$, allows Player I to determine a correct answer for the game, for the first time. There are two cases, according to the answer received. In the first case, Player II answered 1, so that $(i, j)$ received the color red. Then we know that some good path reaches the vertex $(i, j)$, so that the vertices $(i+1, j), (i+1, j+1)$ are not both blue. Hence $((i, j), (i+1, j), (i+1, j+1))$ is not a correct answer for the game, and certainly no other answer could be determined by the coloring of $(i, j)$ by red.

In the second case, Player II answered 0, and therefore $(i, j)$ received the color blue. Then, the answer for the game could conceivably be $((i-1, j-1), (i, j-1), (i, j))$, (we omit the analogous case $((i-1, j), (i, j), (i, j+1))$). But this is a correct answer only if $(i-1, j-1)$ is red and $(i, j-1)$ is blue, and again we get a contradiction with the fact that some good path is known to reach $(i-1, j-1)$ (as it is colored red). Hence our claim is proved.

By Claim 3.3 below, Player II will not answer 1 to a question $x_{d,j}$, until she has answered at least $d-1$ other questions with a 0. It follows that any structured *PYRGEN*$(m, d)$ protocol requires at least $d$ rounds.

■

**Claim 3.3** *In a d-level pyramid with some nodes missing, suppose that a (legal) path from top to bottom exists, and that all existing legal paths reach the same bottom node. Then at least $d-1$ nodes are missing.*

*Proof.* This is obviously true for $d = 2$. For $d > 2$, consider the two pyramid borders, i.e., the possible path $(1,1),(2,2),(3,3),\ldots,(d,d)$, and the possible path $(1,1),(2,1),(3,1),\ldots,(d,1)$. Since their bottom endpoints differ, one or both of these two borders is missing a node. Observe that the bottom node of a border missing a node is inaccessible from the top of the pyramid. So then, strike out the complete border which is missing a node nearest to the top of the pyramid (breaking ties arbitrarily). There remains a $(d-1)$-level pyramid whose top node is not missing, having the property that all paths from top to bottom (of which one exists) reach the same bottom node. By induction, at least $d-2$ nodes are missing in this $(d-1)$-level pyramid. Hence the total number of missing nodes in the original pyramid was at least $d-1$.

■

Using Lemma 3.2, and Theorem 2.1, we obtain that the *general* communication complexity of the $PYRGEN(m, d)$ game satisfies:

**Corollary 3.4** *Assume $m \geq d^{40}$. Then*

$$CC(PYRGEN(m,d)) = \Theta(d \cdot \log m).$$

### 3.4 The Monotone Depth of *GEN*

We will now prove a lower bound for the communication complexity of the monotone KW-game for *GEN*. This will be proved by a reduction to the communication complexity of *PYRGEN*. Let

$$l \stackrel{\text{def}}{=} l(m,d) \stackrel{\text{def}}{=} m \cdot \binom{d+1}{2} + 2.$$

We will consider a set of $l$ *GEN*-elements. The first element, 1, will be the source, and the last element, $l$, will be the target. The other $m \cdot \binom{d+1}{2}$ elements are indexed by $((i,j),k)$, where $1 \leq j \leq i \leq d$, and $1 \leq k \leq m$, that is, $(i,j)$ is a vertex of the pyramid, and $k \in [m]$. We therefore have $m$ *GEN*-elements corresponding to each vertex of the pyramid. We think of the source as placed below the bottom of the pyramid, and we think of the target as placed above the top of the pyramid.

We say that a triple $(v_1, v_2, v_3)$ of *GEN*-elements is **consistent with the structure of the pyramid** in one of the following cases:

1. $v_1, v_2$ are both the source, and $v_3$ corresponds to a vertex at the bottom of the pyramid (i.e., $v_3 = ((d,j),k)$, for some $j,k$). We call such a triple a **source-triple**.

2. $v_3$ is the target, and $v_1, v_2$ both correspond to vertices at the top of the pyramid (i.e., $v_1 = ((1,1),k_1)$, and $v_2 = ((1,1),k_2)$, for some $k_1, k_2$). We call such a triple a **target-triple**.

3. $(v_1, v_2, v_3)$ corresponds to a triangle of the pyramid. That is, for some $i \leq d-1$, and some $j$, and some $k_1, k_2, k_3$, we have $v_1 = ((i+1, j), k_1), v_2 = ((i+1, j+1), k_2)$, and $v_3 = ((i, j), k_3)$. We call such a triple a **triangle-triple**.

**Lemma 3.5** *The communication complexity of PYRGEN$(m, d)$ is at most the communication complexity of the monotone KW-game for the GEN function with $l(m, d)$ elements.*

*Proof.* The proof is by a reduction. Assume that we have a protocol for the monotone KW-game. We will use that protocol to solve the $PYRGEN$ game. As before, let $(x_{i,j})_{1 \leq j \leq i \leq d}, (y_{i,j})_{1 \leq j \leq i \leq d}$ be the inputs for the two players in the $PYRGEN$ game. Player I will construct from $(x_{i,j})$ an input U for the $GEN$ function, such that $GEN(U) = 1$, and Player II will construct from $(y_{i,j})$ an input V for the $GEN$ function, such that $GEN(V) = 0$.

For triples $(v_1, v_2, v_3)$ that are not consistent with the structure of the pyramid, $t_{v_1,v_2,v_3}$ is set to be 0 by both players. That is, in this case we never have $v_1 * v_2 \to v_3$. Thus, we only have to consider triples that are consistent with the structure of the pyramid.

The construction of Player I proceeds as follows: For $1 \leq j \leq i \leq d$, let $g_{i,j}$ be the $GEN$-element $((i, j), x_{i,j})$, that is, the element associated with the position $x_{i,j}$, within the pyramid vertex $(i, j)$. Player I sets each input bit $t_{v_1,v_2,v_3}$ in $U$ to 0, except the bits corresponding to:

$$
\begin{array}{rcl}
g_{1,1} * g_{1,1} & \to & target \\
g_{2,1} * g_{2,2} & \to & g_{1,1} \\
g_{3,1} * g_{3,2} & \to & g_{2,1} \quad ; \quad g_{3,2} * g_{3,3} \to g_{2,2} \\
g_{4,1} * g_{4,2} & \to & g_{3,1} \quad ; \quad g_{4,2} * g_{4,3} \to g_{3,2} \quad ; \quad g_{4,3} * g_{4,4} \to g_{3,3} \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
g_{d,1} * g_{d,2} & \to & g_{d-1,1} \quad ; \quad \cdots \; ; \; \cdots \; ; \; g_{d,d-1} * g_{d,d} \to g_{d-1,d-1} \\
source * source & \to & \{g_{d,1}, g_{d,2}, \ldots, g_{d,d}\}.
\end{array}
$$

In other words, we use Player I's input to construct an input $U$ for the $GEN$ function, in which the source generates the target in a way constrained by the pyramid. Obviously $GEN(U) = 1$.

The construction of Player II proceeds as follows: First, let us use the coloring of Player II to color each $GEN$-element, in the following way: The source is colored 0, the target is colored 1, and every element $((i, j), k)$ is colored by $y_{i,j}(k)$. Now, for triples $(v_1, v_2, v_3)$ consistent with the structure of the pyramid, Player II sets each input bit $t_{v_1,v_2,v_3}$ in $V$ to 1, except if $v_1, v_2$ are both colored 0, and $v_3$ is colored 1. Obviously, an element is generated by the source only if the element is colored 0. Therefore, the target is not generated by the source, and hence $GEN(V) = 0$.

The two players can now apply the protocol for the monotone KW-game. The answer of that protocol is $(v_1, v_2, v_3)$, such that $t_{v_1,v_2,v_3}$ was set to 1 by Player I, and to 0 by Player II. Since $t_{v_1,v_2,v_3}$ was set to 1 by Player I, we know that the triple $(v_1, v_2, v_3)$ is consistent with the structure of the pyramid. Since $t_{v_1,v_2,v_3}$ was set to 0 by Player II, we know that $v_1, v_2$ are both colored 0, and $v_3$ is colored 1. We therefore have the following possibilities:

1. $(v_1, v_2, v_3)$ is a source-triple, in which case it must be that $v_3 = g_{d,j}$, for some $j$, and hence we found $j$ such that $g_{d,j}$ is colored 1, that is, $y_{d,j}(x_{d,j}) = 1$.

2. $(v_1, v_2, v_3)$ is a target-triple, in which case it must be that $v_1 = v_2 = g_{1,1}$, and hence we found that $g_{1,1}$ is colored 0, that is, $y_{1,1}(x_{1,1}) = 0$.

3. $(v_1, v_2, v_3)$ is a triangle-triple. That is, for some $i \leq d-1$, and some $j$, $v_1 = g_{i+1,j}, v_2 = g_{i+1,j+1}$, and $v_3 = g_{i,j}$. Hence we found $i, j$, such that $g_{i,j}$ is colored 1, and both $g_{i+1,j}$, $g_{i+1,j+1}$ are colored 0, that is, $y_{i,j}(x_{i,j}) = 1$, and $y_{i+1,j}(x_{i+1,j}) = y_{i+1,j+1}(x_{i+1,j+1}) = 0$.

In all cases, the goal of the game *PYRGEN*$(m, d)$ is achieved. This concludes the proof.

■

We can now obtain the following lower bound for the monotone depth of *GEN*:

**Corollary 3.6** *For some $\epsilon > 0$, the monotone depth of GEN (with $l$ elements) is $\Omega(l^\epsilon)$.*

*Proof.* Set $m = d^{40}$, and

$$l = l(m, d) = d^{40} \cdot d \cdot (d + 1)/2 + 2 = O(d^{42}).$$

By Lemma 3.1, Lemma 3.5, and Corollary 3.4, the monotone depth of *GEN* is at least $\Omega(d \cdot \log m)$, which is larger than $\Omega(l^{1/42})$.

■

**Corollary 3.7** *Monotone-NC $\neq$ monotone-P.*

### 3.5 A Tight Monotone Depth Hierarchy

As before, consider a set of $l = l(m, d)$ *GEN*-elements. As before, the first element, 1, will be the source, and the last element, $l$, will be the target, and the other $m \cdot \binom{d+1}{2}$ elements are indexed by $((i, j), k)$, where $(i, j)$ is a vertex of the pyramid, and $k \in [m]$. As before, a triple $(v_1, v_2, v_3)$ of *GEN*-elements can be consistent or inconsistent with the structure of the pyramid.

To achieve tight lower bounds for monotone-P, let us introduce the following variation of *GEN*. Note that the *PYRAMID-GEN function* is not to be confused with the *PYRGEN game* defined in Section 3.2. Of course the definition of *PYRAMID-GEN* is specifically targeted for the game *PYRGEN*.

**The *PYRAMID-GEN* function:**
The input is a string of $l^3$ bits $(t_{ijk})_{1 \le i,j,k \le l}$. First, for every triple $(i,j,k)$ that is not consistent with the structure of the pyramid, change $t_{ijk}$ to 0. Now apply the *GEN* function on the new sequence $(t_{ijk})$. The output of *PYRAMID-GEN* on this input will be the output of *GEN*.

In other words: the function *PYRAMID-GEN* determines whether 1 (the source) generates $l$ (the target), using only triples that are consistent with the structure of the pyramid. Note that triples that are not consistent with the structure of the pyramid can be removed from the input, and therefore the relevant input is of length lower than $l^3$.

**Proposition 3.8** *PYRAMID-GEN$(m,d)$ can be solved by a monotone polynomial size circuit family of depth $O(d \cdot \log m)$.*

*Proof.* The structure of the circuit resembles the one of the pyramid. The circuit simply computes, in the $r$-th stage (for $1 \le r \le d$), the set of *GEN*-elements generated by the source, and belonging to the $(d-r+1)$-th layer of the pyramid. Since, for any *GEN* element $g$ corresponding to the vertex $(i,j) = (d-r+1,j)$ of the pyramid, we only care about triples $f * h \to g$, in which $f$ and $h$ respectively correspond to the two vertices $(i+1,j)$, and $(i+1,j+1)$ (except when $i = d$, in which case we only care about $1 * 1 \to g$), the depth of each stage is $O(\log m)$. The circuit completes its task by checking that some element $g$, generated by the source and belonging to the top layer of the pyramid, satisfies $g * g \to target$.

■

Since the proof of Lemma 3.5 applies to *PYRAMID-GEN* as well, we have:

**Lemma 3.9** *The communication complexity of PYRGEN$(m,d)$ is at most the communication complexity of the monotone KW-game for the function PYRAMID-GEN$(m,d)$.*

■

**Corollary 3.10** *Assume $m \ge d^{40}$, then the monotone depth of PYRAMID-GEN$(m,d)$ is $\Theta(d \cdot \log m)$.*

Now fix $m = d^{40}$, and use a standard padding argument (when needed), to get the following corollary:

**Corollary 3.11** *There exist constants $\epsilon, c > 0$, such that for any integer function $D(n) \le n^\epsilon$, there exists an explicit monotone function $F : \{0,1\}^n \to \{0,1\}$, that can be (uniformly) computed by a family of monotone Boolean circuits of polynomial size and of depth $D(n)$, and cannot be computed by any family of monotone Boolean circuits of depth less than $c \cdot D(n)$.*

As a result we obtain the following corollary:

**Corollary 3.12** *For every $i \ge 0$, monotone-NC$^i$ $\neq$ monotone-NC$^{i+1}$.*

# 4 Other Applications

In this section, we prove other applications of Theorem 2.1 and Theorem 2.2:

## 4.1 Lower Bound for $st$-Connectivity

As mentioned above, a tight lower bound of $\Omega(\log^2 n)$ was proved for $st$-connectivity in [KaWi88]. Here we show how to obtain that lower bound as an immediate consequence of Theorem 2.1. The proof, moreover, is different from the one given in [KaWi88].

For $m, n$, define the communication game $CONN(m,n)$ to be the following $DART(m,n)$ game: The input for Player I is $(x_i)_{1 \leq i \leq n}$, and for Player II $(y_i)_{1 \leq i \leq n}$. As before, for every $i$, define $e_i = y_i(x_i)$. The goal of the players is to find $i$ such that one of the following is satisfied:

1. $i = 1$, and $e_i = 1$, or
2. $i = n$, and $e_i = 0$, or
3. $i \leq n-1$, and $(e_i = 0) \wedge (e_{i+1} = 1)$, or
4. $i \leq n-1$, and $(e_i = 1) \wedge (e_{i+1} = 0)$.

It is easy to verify that

$$\bigvee_{i \leq n-1} (e_i \neq e_{i+1}) \bigvee (e_1 = 1) \bigvee (e_n = 0)$$

is a tautology, and therefore the goal can always be achieved.

First, we claim that

$$SC(CONN(m,n)) = \lceil \log_2(n+1) \rceil.$$

The upper bound and the lower bound are both trivial: The upper bound follows by the obvious "divide and conquer" protocol. As for the lower bound, note that in every protocol for the game, every clause of the first 3 types must appear as an answer at least once, and hence the number of "leaves" in the protocol is at least $n+1$.

Thus, by Theorem 2.1 we have for $m \geq n^{20}$,

$$CC(CONN(m,n)) = \Omega(\log n \cdot \log m).$$

To see the connection to $st$-connectivity, consider $n \cdot m + 2$ vertices: the two special vertices, $s$ and $t$, and $n \cdot m$ other vertices, indexed by $(i, j)$, where $1 \leq i \leq n$, and $1 \leq j \leq m$. Given any protocol for the monotone KW-game for $st$-connectivity on these vertices, the two players can use that protocol to solve $CONN(m,n)$, in the following way:

Player I uses $(x_i)$ to construct a graph U, where $s$ and $t$ are connected. The edges of the graph $U$ will contain the following pairs (and no other pair):

$$(s,(1,x_1)) \ , \ ((1,x_1),(2,x_2)) \ , \ ((2,x_2),(3,x_3)) \ , \ \ldots \ , \ ((n-1,x_{n-1}),(n,x_n)) \ , \ ((n,x_n),t).$$

Player II uses $(y_i)$ to construct a graph V, where $s$ and $t$ are not connected. First, Player II uses $(y_i)$ to color every vertex in the following way: $s$ is colored 0, $t$ is colored 1, and every vertex $(i,j)$ is colored $y_i(j)$. The edges of the graph $V$ will contain exactly all pairs where both vertices are colored the same.

The two players can now apply the protocol for the monotone KW-game. It is easy to verify that the answer given by that protocol is a correct answer for the $CONN(m,n)$ game as well.

Now fix $m = n^{20}$, and use Lemma 3.1 to get that the monotone depth of $st$-connectivity is $\Omega(\log^2 n)$.

### 4.2 Lower Bound for $k$-Clique

For the monotone depth of the $k$-clique function on a graph with $n$ vertices, a lower bound of $\Omega(k)$ was proved in [RaWi90]. Obviously, for $k = \Omega(n)$, that lower bound is tight. For smaller values of $k$, however, the lower bound is not tight. Here we prove that for $k \leq n^\epsilon$ (for some small constant $\epsilon > 0$), the monotone depth of $k$-clique is $\Omega(k \cdot \log n)$. Obviously, this lower bound is tight. This lower bound was previously known for $k \leq \log n$ [AlBo87].

For $m, k$, define the communication game $CLQ(m,k)$ to be the following $DART_{(k-1)}(m,k)$ game: The input for Player I is $(x_i)_{1 \leq i \leq k}$, and for Player II $(y_i)_{1 \leq i \leq k}$. As before, for every $i$, define $e_i = y_i(x_i)$. Note that $e_i \in [k-1]$. The goal of the players is to find $i,j$, such that $e_i = e_j$. Since the number of colors is $k-1$, the goal can always be achieved.

First, we claim that

$$SC(CLQ(m,k)) = k.$$

The upper bound follows by the trivial protocol of asking about $e_1, \ldots, e_k$ one by one. The lower bound follows by the obvious adversarial strategy (for Player II) of answering each question with a different color, until no colors are left.

Thus, by Theorem 2.2 we have for $m \geq k^{1000}$,

$$CC(CLQ(m,k)) = \Omega(k \cdot \log m).$$

To see the connection to $k$-clique, consider $k \cdot m$ vertices, indexed by $(i,j)$, where $1 \leq i \leq k$, and $1 \leq j \leq m$. Given any protocol for the monotone KW-game for $k$-clique on these vertices, the two players can use that protocol to solve $CLQ(m,k)$, in the following way:

Player I uses $(x_i)$ to construct a graph U that will be a clique of size $k$. In the graph $U$ we will have for every $i,j$, the edge $((i,x_i),(j,x_j))$ (and no other edges).

Player II uses $(y_i)$ to construct a graph V, containing no clique of size $k$. First, Player II uses $(y_i)$ to color the vertices in the following way: a vertex $(i,j)$ is colored

$y_i(j)$. The edges of the graph $V$ will contain exactly all pairs where the two vertices are colored differently.

The two players can now apply the protocol for the monotone KW-game. It is easy to verify that the answer given by that protocol translates into a correct answer for the $CLQ(m,k)$ game as well.

Now fix $m = k^{1000}$, and use Lemma 3.1 to get that for $k \leq n^{1/1001}$, the monotone depth of $k$-clique is $\Omega(k \cdot \log n)$.

## 5 Thickness and Predictability

In this section, we present some of our main tools, and notations, used for the proof of Theorem 2.1. The "average-degree", $AVDEG_j(A)$, defined below, is analogous to the *predictability* notion, introduced by [EdImRuSg91]. Our proof uses tools and intuitions from [EdImRuSg91].

Let $X = [m]^n$. As before, let $A$ be a subset of $X$. The bipartite graph $GRAPH_1(A)$ is defined in the following way: Consider a bipartite graph with disjoint vertex sets $V_L = [m]$ (the "left nodes"), and $V_R = [m]^{n-1}$ (the "right nodes"). The set of edges $E$ contains all the pairs $(x_1, (x_2, \ldots, x_n))$ s.t., $(x_1, x_2, \ldots, x_n) \in A$. In other words, the set of edges is the set $A$, where each $(x_1, x_2, \ldots, x_n) \in A$ is viewed as an edge between the "left" node $x_1$ and the "right" node $(x_2, \ldots, x_n)$.

For every $1 \leq j \leq n$, the bipartite graph $GRAPH_j(A)$ is now defined in the same way, where as before $V_L = [m]$, $V_R = [m]^{n-1}$, and each $(x_1, x_2, \ldots, x_n) \in A$ is viewed as an edge between the "left" node $x_j$ and the "right" node $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$.

For the graph $GRAPH_j(A)$, define the set $\hat{V}_j$ to be the set of all nodes in $V_R$ with non-zero degree (that is, the set of right nodes with non-zero degree). The average-degree $AVDEG_j(A)$ is defined to be the average degree of a right node in $\hat{V}_j$ in the graph $GRAPH_j(A)$, that is,

$$AVDEG_j(A) = \frac{|A|}{|\hat{V}_j|}.$$

Using different notations, we define $AVDEG_j(A)$ by

$$AVDEG_j(A) = \frac{|A|}{|A_{[n]\setminus\{j\}}|},$$

where $A_{[n]\setminus\{j\}}$ denotes the projection of $A$ on $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$, that is, $|A_{[n]\setminus\{j\}}|$ is the number of assignments to $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$ such that there exists at least one assignment to $x_j$ satisfying $(x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n) \in A$.

Observe that $AVDEG_j(A)$ ranges from $m$ to 1. When $AVDEG_j(A) = 1$, $x_j$ is fixed as a function of $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$. In that case, "the $j^{th}$ slot is totally predictable" in the sense that knowing $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$ determines the value of $x_j$. When $AVDEG_j(A) = m$, the degree of every right node is precisely $m$, (since the average degree of the right nodes is $m$ and clearly $m$ is also the maximum degree of any right node). In that case, $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$ gives no information on $x_j$.

While $AVDEG_j(A)$ is the *average* degree of right nodes in $\hat{V}_j$, we will also be interested in the minimal degree of such a right node. Define $MINDEG_j(A)$ to be the minimal degree of a right node in $\hat{V}_j$, in the graph $GRAPH_j(A)$.

The **thickness of** $A$ is now defined by

$$Thickness(A) = \min_{1 \leq j \leq n} MINDEG_j(A).$$

### 5.1 Some Useful Observations

The following observation is analogous to [EdImRuSg91, Lemma 4]:

**Claim 5.1** *Let $A' \subseteq A$. Then for every $j$,*

$$AVDEG_j(A') \geq \frac{|A'|}{|A|} \cdot AVDEG_j(A).$$

*Proof.* Obviously, $A'_{[n]\setminus\{j\}} \subseteq A_{[n]\setminus\{j\}}$. Therefore,

$$\frac{|A'|}{|A|} \cdot AVDEG_j(A) = \frac{|A'|}{|A|} \cdot \frac{|A|}{|A_{[n]\setminus\{j\}}|} = \frac{|A'|}{|A_{[n]\setminus\{j\}}|} \leq \frac{|A'|}{|A'_{[n]\setminus\{j\}}|} = AVDEG_j(A').$$

■

The projection set $A_{[n]\setminus\{j\}}$ can be viewed as a subset of $[m]^{n-1}$. For $n \geq 2$ and for every $i \in \{1, \ldots, j-1, j+1, \ldots, n\}$, one can define, as before, the bipartite graph $GRAPH_i(A_{[n]\setminus\{j\}})$, and the minimal and average degrees, $MINDEG_i(A_{[n]\setminus\{j\}})$ and $AVDEG_i(A_{[n]\setminus\{j\}})$. The following claim shows that the thickness of $A_{[n]\setminus\{j\}}$ is never smaller than the thickness of $A$, that is, projections never decrease the thickness.

**Claim 5.2** *For any $j$,*

$$Thickness(A_{[n]\setminus\{j\}}) \geq Thickness(A).$$

*Proof.* For every $i \neq j$, we will show that

$$MINDEG_i(A_{[n]\setminus\{j\}}) \geq MINDEG_i(A).$$

This is proved by the following argument:

W.l.o.g., assume that $j = n$, and $i = n - 1$. For every non-zero-degree right-node $(x_1, \ldots, x_{n-2})$ of the graph $GRAPH_{n-1}(A_{[n]\setminus\{n\}})$, there exists (at least one) $x_{n-1}$, such that $(x_1, \ldots, x_{n-2}, x_{n-1}) \in A_{[n]\setminus\{n\}}$. Hence, by the definition of $A_{[n]\setminus\{n\}}$, there exists (at least one) $x_n$ such that $(x_1, \ldots, x_{n-2}, x_{n-1}, x_n) \in A$. Therefore, by the definition of $MINDEG_{n-1}(A)$, there exists at least $MINDEG_{n-1}(A)$ different elements $x'_{n-1}$, such that $(x_1, \ldots, x_{n-2}, x'_{n-1}, x_n) \in A$. Obviously, all these elements satisfy $(x_1, \ldots, x_{n-2}, x'_{n-1}) \in A_{[n]\setminus\{n\}}$. Hence, the degree of the right node $(x_1, \ldots, x_{n-2})$ in the graph $GRAPH_{n-1}(A_{[n]\setminus\{n\}})$ is at least $MINDEG_{n-1}(A)$.

■

## 5.2 The Thickness Lemma

The following lemma is our most important technical tool. It shows that if for a set $A$, $AVDEG_j(A)$ is large for every $j$, then there exists a large subset $A'$ of $A$, such that $Thickness(A')$ is large. We will first state the lemma in a general form, and then restate it in a simpler form that will be used herein.

**Lemma 5.3** *If for some $1 > \delta > 0$, and for every $j$, $AVDEG_j(A) \geq \delta \cdot m$ then for any $\alpha \geq 0$, there exists $A' \subset A$, such that:*

1. $|A'| \geq (1-\alpha)|A|$, *and*
2. $Thickness(A') \geq \Delta$, *where*
$$\Delta \stackrel{\text{def}}{=} \frac{\alpha\delta m}{n}$$

*Proof.* [2] Assume that $\alpha > 0$ (otherwise the proof is trivial). Define $A^0 = A$, and define a sequence, $A^0 \supset A^1 \supset A^2 \ldots$, of subsets of $A$, in the following way: For every $i \geq 0$,

1. if $|A^i| < (1-\alpha)|A|$ then STOP.
2. if $Thickness(A^i) \geq \Delta$ then STOP.
3. Otherwise, there exists $j$ (w.l.o.g. $j = 1$) with $MINDEG_j(A^i) < \Delta$. Thus in the graph $GRAPH_1(A^i)$ there exists a right node $(v_2, \ldots, v_n)$, with degree larger than 0 and smaller than $\Delta$. To get $A^{i+1}$, remove from $A^i$ all elements $(x_1, x_2, \ldots, x_n)$, with $x_2 = v_2, \ldots, x_n = v_n$, that is, remove the vertex $(v_2, \ldots, v_n)$, and all edges from it.

The last set in the sequence, $A^l$, is the required $A'$. To show that $A^l$ satisfies the requirements, we just have to prove that we always stop because of the second condition, and never because of the first one (hence, $A^l$ satisfies both: $|A^l| \geq (1-\alpha)|A|$, and $Thickness(A^l) \geq \Delta$).

In each step, $A^{i+1}$ is defined from $A^i$ by removing one right node from the graph $GRAPH_j(A^i)$ (for some $j$). Hence, for that particular $j$,

$$|A^{i+1}_{[n]\setminus\{j\}}| = |A^i_{[n]\setminus\{j\}}| - 1,$$

and for every $j' \neq j$,

$$|A^{i+1}_{[n]\setminus\{j'\}}| \leq |A^i_{[n]\setminus\{j'\}}|.$$

Therefore, the total number of steps, $l$, satisfies

$$l \leq \sum_{j=1}^{n} |A_{[n]\setminus\{j\}}| = \sum_{j=1}^{n} |A|/AVDEG_j(A),$$

[2]This proof, suggested by Mike Saks, is simpler and more elegant than our original proof.

and since for every $j$, $AVDEG_j(A) \geq \delta \cdot m$, we can conclude that

$$l \leq (n \cdot |A|)/(\delta \cdot m).$$

Since $A^{i+1}$ is defined from $A^i$ by removing at most $\Delta$ elements, the total number of elements removed from $A$ (in all the steps combined) to get $A^l$ is $\leq$

$$\Delta \cdot l \leq \Delta \cdot (n \cdot |A|)/(\delta \cdot m) = \alpha \cdot |A|$$

(the last equality follows by the definition of $\Delta$).

Hence $|A^l| \geq (1 - \alpha) \cdot |A|$ (i.e., we couldn't have stopped because of the first condition).

■

**Corollary 5.4** *Assume that $m \geq n^{20}$. If for every $j$, $AVDEG_j(A) \geq 4 \cdot m^{19/20}$ then there exists $A' \subset A$, such that:*

1. $|A'| \geq |A|/2$, *and*
2. $Thickness(A') \geq m^{17/20}$.

*Proof.* Take in Lemma 5.3; $\alpha = 1/2$, and $\delta = 4 \cdot m^{-1/20}$ to get

$$\Delta = \frac{2 \cdot m^{19/20}}{n} \geq m^{17/20}.$$

■

# 6 Proof of the Main Theorem

In this section we prove Theorem 2.1. The extension of Theorem 2.1 to the case of multi-color dart games (Theorem 2.2) is discussed in subsection 6.4.

Assume that $m \geq n^{20}$, and assume (for simplicity) that $m^{1/20}$ is larger than some big constant (say $m^{1/20} \geq 1000$). As before, we denote by $X$ the set $[m]^n$, we denote by $Y$ the set $(\{0,1\}^m)^n$, and we denote by $R \subseteq X \times Y \times Z$ a relation in $DART(m,n)$. As before, we denote by $A$ a subset of $X$, and we denote by $B$ a subset of $Y$. As before, for a relation $R$, and for two subsets $A \subseteq X$, and $B \subseteq Y$, we denote by $C_R(A,B)$ the deterministic communication complexity of the relation $R$, over the domain $A \times B$.

We measure the size of $A, B$ by

$$\alpha = \log_2 \left( \frac{|X|}{|A|} \right) \quad , \quad \beta = \log_2 \left( \frac{|Y|}{|B|} \right),$$

that is, $\alpha, \beta$ are the number of bits of information known about $A, B$ respectively. We will be interested in sets $A$ with

$$Thickness(A) \geq m^{17/20}.$$

Such a set $A$ is said to be *thick*.

For any $\alpha, \beta, k \geq 0$, and $m \geq 1000^{20}$, denote by $GAMES_m[\alpha, \beta, k]$ the set of all triples $(R, A, B)$ such that for some $n \leq m^{1/20}$:

1. $R$ is a relation in $DART(m,n)$, such that

$$SC(R) \geq k.$$

2. $A$ is a thick subset of $X$, such that $\log_2(|X|/|A|) \leq \alpha$, that is,

$$Thickness(A) \geq m^{17/20},$$

   and

$$|A| \geq 2^{-\alpha} \cdot |X|.$$

3. $B$ is a subset of $Y$, such that $\log_2(|Y|/|B|) \leq \beta$, that is,

$$|B| \geq 2^{-\beta} \cdot |Y|.$$

$COMP_m[\alpha,\beta,k]$ is now defined to be the minimum of $C_R(A,B)$, over all triples $(R,A,B) \in GAMES_m[\alpha,\beta,k]$. We will prove here a general lower bound for $COMP_m[\alpha,\beta,k]$.

Given $\alpha,\beta,k,m$, let $(R,A,B) \in GAMES_m[\alpha,\beta,k]$ be a triple with minimal $C_R(A,B)$, that is,

$$C_R(A,B) = COMP_m[\alpha,\beta,k].$$

To bound $C_R(A,B)$ we will consider two cases:

1. CASE 1: For every $1 \leq j \leq n$, $AVDEG_j(A) \geq 8 \cdot m^{19/20}$.

2. CASE 2: For some $1 \leq j \leq n$, $AVDEG_j(A) < 8 \cdot m^{19/20}$.

### 6.1 A Recursive Bound for $C_R(A,B)$ in CASE 1

To bound $C_R(A,B)$ in the first case, we use the following lemma:

**Lemma 6.1** *For any $\alpha,\beta,k,m \geq 0$, with $\beta \leq m^{2/20}$, and $m \geq 1000^{20}$, and for any $(R,A,B) \in GAMES_m[\alpha,\beta,k]$, if for every $1 \leq j \leq n$, $AVDEG_j(A) \geq 8 \cdot m^{19/20}$ then*

$$C_R(A,B) \geq MIN(\ COMP_m[\alpha+2,\beta,k]\ ,\ COMP_m[\alpha,\beta+1,k]\ ) + 1.$$

*Proof.* First, let us prove that $C_R(A,B)$ is not $0$ : If $C_R(A,B) = 0$ then in the domain $A \times B$ an answer for $R$ is already known. By the fourth requirement in the definition of dart games, there exists a clause of the DNF-formula $F_R$, that is always satisfied over $A \times B$. Therefore, in the domain $A \times B$, for at least one $1 \leq j \leq n$, $y_j(x_j)$ is a constant function (that is, $y_j(x_j)$ is either always 0 or always 1). W.l.o.g. assume that $y_j(x_j) = 0$ (in that domain). Since $AVDEG_j(A) \geq 8 \cdot m^{19/20}$, there are at least $8 \cdot m^{19/20}$ possible values for $x_j$ (in the domain $A \times B$), and since $y_j(x_j) = 0$, the coloring $y_j$ colors all these values by 0. This, however, contradicts the assumption $\beta \leq m^{2/20}$.

Let $P$ be the best protocol for solving $R$ over $A \times B$, that is, a protocol with communication complexity $C_R(A,B)$. Consider the first bit transmitted by $P$. That bit is transmitted either by Player I or by Player II.

If Player II transmits the first bit then partition the set $B$ into $B = B_0 \cup B_1$, according to the bit transmitted, that is, $B_0$ is the set of inputs (for Player II) where 0 is transmitted, and $B_1$ is the set of inputs where 1 is transmitted. Obviously, $|B_0| + |B_1| = |B|$. W.l.o.g., assume that $|B_0| \geq |B|/2$, and consider the triple $(R, A, B_0)$. The protocol $P$ solves $R$ on $A \times B_0$, using only $C_R(A,B) - 1$ communication bits (since one bit was already transmitted). Since $(R, A, B_0)$ is obviously in $GAMES_m[\alpha, \beta+1, k]$, we have in this case $COMP_m[\alpha, \beta+1, k] \leq C_R(A,B) - 1$.

If Player I transmits the first bit then partition the set $A$ into $A = A_0 \cup A_1$, according to the bit transmitted, and assume w.l.o.g. that $|A_0| \geq |A|/2$. $A_0$ is not necessarily thick, and therefore $(R, A_0, B)$ is not necessarily in $GAMES_m[\alpha+1, \beta, k]$. However, since for every $j$, $AVDEG_j(A) \geq 8 \cdot m^{19/20}$, we know by Claim 5.1 that for every $j$, $AVDEG_j(A_0) \geq 4 \cdot m^{19/20}$. Therefore, by Corollary 5.4, there exists $A' \subset A_0$ such that $|A'| \geq |A_0|/2$, and $Thickness(A') \geq m^{17/20}$. Therefore, $(R, A', B) \in GAMES_m[\alpha+2, \beta, k]$. Since $P$ solves $R$ on $A' \times B$, using only $C_R(A,B) - 1$ communication bits, we have in this case $COMP_m[\alpha+2, \beta, k] \leq C_R(A,B) - 1$.

■

### 6.2 A Recursive Bound for $C_R(A,B)$ in CASE 2

In the second case, we use the following lemma to bound $C_R(A,B)$:

**Lemma 6.2** *For any $\alpha, \beta, k, m \geq 0$, with $\beta \leq m^{2/20}$, $k \geq 1$, and $m \geq 1000^{20}$, and for any $(R, A, B) \in GAMES_m[\alpha, \beta, k]$, if for some $1 \leq j \leq n$, $AVDEG_j(A) < 8 \cdot m^{19/20}$ then*

$$C_R(A,B) \geq COMP_m[\alpha + 3 - (\log_2 m)/20, \beta+1, k-1].$$

*Proof.* W.l.o.g., assume that $j = n$, that is,

$$AVDEG_n(A) < 8 \cdot m^{19/20}.$$

Since $Thickness(A) \geq m^{17/20}$,

$$MINDEG_n(A) \geq m^{17/20}.$$

Denote by $R_0$ the restriction of the relation $R$ to the first $n-1$ coordinates, derived by fixing $e_n \stackrel{\text{def}}{=} y_n(x_n)$ to be 0, and denote by $R_1$ the restriction derived by fixing $e_n \stackrel{\text{def}}{=} y_n(x_n)$ to be 1. Obviously, both $R_0, R_1$ are relations in $DART(m, n-1)$. Since $SC(R) \geq k$, at least one of $SC(R_0), SC(R_1)$ is $\geq k-1$. W.l.o.g., assume that

$$SC(R_0) \geq k-1.$$

We will prove the lemma by showing the existence of $A' \subset [m]^{n-1}$, and $B' \subset (\{0,1\}^m)^{n-1}$, such that

$$(R_0, A', B') \in GAMES_m[\alpha + 3 - (\log_2 m)/20, \beta + 1, k - 1],$$

and

$$C_{R_0}(A', B') \leq C_R(A, B).$$

Therefore, we will have

$$C_R(A, B) \geq C_{R_0}(A', B') \geq COMP_m[\alpha + 3 - (\log_2 m)/20, \beta + 1, k - 1],$$

which proves the lemma.

For every subset $U \subset [m]$, let us define sets $A_U \subset [m]^{n-1}$, $B_U \subset (\{0,1\}^m)^{n-1}$. The sets $A', B'$ above will be the sets $A_U, B_U$ for some particular choice of $U$.

- The set $A_U$ is defined in the following way: $(x_1, \ldots, x_{n-1}) \in A_U$ iff there exists an element $v \in U$, such that $(x_1, \ldots, x_{n-1}, v) \in A$. In other words, $A_U$ is the set of all right nodes in the graph $GRAPH_n(A)$ that are connected by an edge to (at least one) element of the set $U$ (viewed as a subset of the set of left nodes).

- The set $B_U$ is defined in the following way: $(y_1, \ldots, y_{n-1}) \in B_U$ iff there exists a coloring $w \in \{0,1\}^{[m]}$, such that all elements of $U$ are colored 0 by $w$, and such that $(y_1, \ldots, y_{n-1}, w) \in B$.

**Claim 6.3** *For every $U \subset [m]$,*

$$C_{R_0}(A_U, B_U) \leq C_R(A, B).$$

*Proof.* Given a communication protocol $P$ that solves $R$ over $A \times B$, we will describe a protocol of the same communication complexity (or less) that solves $R_0$ over $A_U \times B_U$. The protocol is in fact very simple: Given an input $(x_1, \ldots, x_{n-1}) \in A_U$ for Player I, and an input $(y_1, \ldots, y_{n-1}) \in B_U$ for Player II, there exist $v \in U$, such that $(x_1, \ldots, x_{n-1}, v) \in A$, and $w \in \{0,1\}^{[m]}$, such that all elements of $U$ are colored 0 by $w$, and such that $(y_1, \ldots, y_{n-1}, w) \in B$. $v$ doesn't depend on $(y_1, \ldots, y_{n-1})$, and can therefore be computed by Player I (without communication). In the same way, $w$ doesn't depend on $(x_1, \ldots, x_{n-1})$, and can therefore be computed by Player II. The players can then apply $P$ on $(x_1, \ldots, x_{n-1}, v), (y_1, \ldots, y_{n-1}, w)$ to get a solution for $R$. However, since by our construction $w(v)$ is always 0, a solution for $R$ on the inputs $(x_1, \ldots, x_{n-1}, v), (y_1, \ldots, y_{n-1}, w)$ is also a solution for $R_0$ on the inputs $(x_1, \ldots, x_{n-1}), (y_1, \ldots, y_{n-1})$.

■

To complete the proof of the lemma, we still have to prove that for some $U \subset [m]$, $(R_0, A_U, B_U) \in GAMES_m[\alpha + 3 - (\log_2 m)/20, \beta + 1, k - 1]$. To prove this we still have to show that for some $U$:

1. $|A_U| \geq 2^{-[\alpha+3-(\log_2 m)/20]} \cdot m^{n-1}$,

2. $|B_U| \geq 2^{-[\beta+1]} \cdot 2^{m\cdot(n-1)}$, and

3. $Thickness(A_U) \geq m^{17/20}$.

We will use a probabilistic argument:

Let $U$ be a random subset of $[m]$, of size $m^{5/20}$ (we assume for simplicity that $m^{5/20}$ is an integer). The following claim shows that with high probability $A_U = A_{[n]\setminus\{n\}}$, (that is, $A_U$ contains every single element of $A_{[n]\setminus\{n\}}$).

**Claim 6.4** *For a random set $U$ of size $m^{5/20}$,*

$$Prob_U \left[A_U = A_{[n]\setminus\{n\}}\right] \geq 3/4.$$

*Proof.* Obviously, $A_U \subseteq A_{[n]\setminus\{n\}}$. The other direction is not always true.

Recall that

$$MINDEG_n(A) \geq m^{17/20}.$$

For every element $(x_1, \ldots, x_{n-1}) \in A_{[n]\setminus\{n\}}$, denote by $V_{(x_1,\ldots,x_{n-1})}$ the set of all $v \in [m]$, such that $(x_1, \ldots, x_{n-1}, v) \in A$, (that is, the set of all left nodes, connected to $(x_1, \ldots, x_{n-1})$ in the graph $GRAPH_n(A)$). Then, for every $(x_1, \ldots, x_{n-1}) \in A_{[n]\setminus\{n\}}$,

$$|V_{(x_1,\ldots,x_{n-1})}| \geq m^{17/20},$$

and $(x_1, \ldots, x_{n-1}) \in A_U$ iff $U$ intersects $V_{(x_1,\ldots,x_{n-1})}$.

Since $U$ is of size $m^{5/20}$, the probability that $U$ doesn't intersect $V_{(x_1,\ldots,x_{n-1})}$ is at most

$$\left(1 - m^{17/20}/m\right)^{m^{5/20}} = \left(1 - m^{-3/20}\right)^{m^{3/20}\cdot m^{2/20}} \leq \mathbf{e}^{-m^{2/20}}.$$

Since the number of elements $(x_1, \ldots, x_{n-1}) \in A_{[n]\setminus\{n\}}$ is less than

$$m^n \leq m^{m^{1/20}} \leq 2^{m^{1/20}\cdot\log_2 m},$$

the probability that one of them is not in $A_U$ is at most

$$\mathbf{e}^{-m^{2/20}} \cdot 2^{m^{1/20}\cdot\log_2 m} < 2^{-(m^{2/20} - m^{1/20}\cdot\log_2 m)} < 1/4.$$

■

The following claim shows that with high probability $B_U$ is large.

**Claim 6.5** *For a random set $U$ of size $m^{5/20}$,*

$$Prob_U \left[|B_U| \geq |B|/2^{m+1}\right] \geq 3/4.$$

*Proof.* For every element $(y_1,\ldots,y_{n-1}) \in (\{0,1\}^m)^{n-1}$, denote by $W_{(y_1,\ldots,y_{n-1})}$ the set of all $w \in \{0,1\}^{[m]}$, such that $(y_1,\ldots,y_{n-1},w) \in B$. Then, for every $(y_1,\ldots,y_{n-1}) \in (\{0,1\}^m)^{n-1}$, $(y_1,\ldots,y_{n-1}) \in B_U$ iff there exists (at least one) coloring $w \in W_{(y_1,\ldots,y_{n-1})}$ that colors all the elements of $U$ by 0.

Denote, $\delta = |B|/2^{m\cdot n}$. We have to prove that with probability of at least 3/4,

$$|B_U|/2^{m\cdot(n-1)} \geq (1/2)\cdot\delta.$$

Denote by $\hat{B}$ the set of all $(y_1,\ldots,y_{n-1}) \in (\{0,1\}^m)^{n-1}$, with

$$|W_{(y_1,\ldots,y_{n-1})}| \geq (1/4)\cdot\delta\cdot 2^m.$$

Then,

$$|B| \leq |\hat{B}|\cdot 2^m + |(\{0,1\}^m)^{n-1}\setminus\hat{B}|\cdot(1/4)\cdot\delta\cdot 2^m \leq |\hat{B}|\cdot 2^m + (1/4)\cdot\delta\cdot 2^{m\cdot n},$$

and since $|B| = \delta\cdot 2^{m\cdot n}$,

$$|\hat{B}|\cdot 2^m \geq (3/4)\cdot\delta\cdot 2^{m\cdot n},$$

that is

$$|\hat{B}|/2^{m\cdot(n-1)} \geq (3/4)\cdot\delta.$$

Hence, to complete the proof of the claim, it is enough to prove that with probability of at least 3/4,

$$|B_U| \geq (2/3)\cdot|\hat{B}|.$$

To prove this, it is enough to prove that every element of $\hat{B}$ is contained in $B_U$, with probability of at least $11/12$ $(= (3/4)\cdot 1 + (1/4)\cdot(2/3))$. Hence, let us consider one element $(y_1,\ldots,y_{n-1}) \in \hat{B}$, and prove that with high probability $(y_1,\ldots,y_{n-1}) \in B_U$. Since for $(y_1,\ldots,y_{n-1}) \in \hat{B}$,

$$|W_{(y_1,\ldots,y_{n-1})}| \geq (1/4)\cdot\delta\cdot 2^m \geq (1/4)\cdot 2^{-\beta}\cdot 2^m,$$

and since the lemma assumes that $\beta \leq m^{2/20}$, we have for $(y_1,\ldots,y_{n-1}) \in \hat{B}$,

$$|W_{(y_1,\ldots,y_{n-1})}| \geq (1/4)\cdot 2^{-m^{2/20}}\cdot 2^m.$$

Therefore, Claim 6.5 follows by the following claim:

**Claim 6.6** *Let $W \subset \{0,1\}^m$ be any set of colorings, such that $|W| \geq (1/4)\cdot 2^{-m^{2/20}}\cdot 2^m$, and let $U$ be a random subset of $[m]$, of size $m^{5/20}$, then with probability of at least $11/12$ there exists a coloring $w \in W$ that colors all the elements of $U$ by 0.*

*Proof.* For simplicity, let us assume that $l \stackrel{\text{def}}{=} (1/2)\cdot m^{15/20}$ is an integer.

Let us pick a (uniformly distributed) random set $U \subset [m]$ of size $m^{5/20}$ in the following way:

1. first pick a random set $\tilde{U} \subset [m]$ of size $2\cdot m^{5/20}$,

2. then pick a random coloring $C$ of $\tilde{U}$, with exactly $m^{5/20}$ elements colored 0 and exactly $m^{5/20}$ elements colored 1,

3. finally define $U \subset \tilde{U}$ to be the set of zeros of $C$.

Denote by $\tilde{W}$ the restriction of $W$ to the subset $\tilde{U}$, that is, the set of all colorings of $\tilde{U}$ that are restrictions to $\tilde{U}$ of colorings in $W$.

First, we claim that with very high probability (over the choice of $\tilde{U}$), the set $\tilde{W}$ contains almost all colorings of $\tilde{U}$. More precisely, with probability of at least (say) $(1 - m^{-1/20})$, we have (say)

$$|\tilde{W}| \geq (1 - m^{-5/20}) \cdot 2^{|\tilde{U}|}.$$

To prove this, note that otherwise, by the probabilistic method, one can partition the set $[m]$ into $l$ disjoint subsets $\tilde{U}_1, \tilde{U}_2, \ldots, \tilde{U}_l$, of size $2 \cdot m^{5/20}$ each, such that for a fraction of at least $m^{-1/20}$ of the sets $\tilde{U}_i$, the restriction, $\tilde{W}_i$, of $W$ to $\tilde{U}_i$ contains a fraction of less than $(1 - m^{-5/20})$ of the colorings of $\tilde{U}_i$. A contradiction is then derived by the following inequality

$$\begin{aligned} |W| \;&\leq\; \prod_{i=1}^{l} |\tilde{W}_i| \leq \left(1 - m^{-5/20}\right)^{m^{-1/20} \cdot l} \cdot 2^m \leq \\ &\left(1 - m^{-5/20}\right)^{m^{5/20} \cdot m^{8/20}} \cdot 2^m \leq \mathbf{e}^{-m^{8/20}} \cdot 2^m < (1/4) \cdot 2^{-m^{2/20}} \cdot 2^m. \end{aligned}$$

Thus, with probability of at least $(1 - m^{-1/20})$, $\tilde{W}$ contains a fraction of at least $(1 - m^{-5/20})$ of the colorings of $\tilde{U}$. Assume therefore that indeed $\tilde{W}$ contains a fraction of at least $(1 - m^{-5/20})$ of the colorings of $\tilde{U}$, and consider colorings of $\tilde{U}$ with exactly $m^{5/20}$ elements colored 0, and exactly $m^{5/20}$ elements colored 1. Since the set of all these colorings is of fraction larger than (say) $(1/10) \cdot \sqrt{m^{-5/20}}$ of the set of all colorings (of $\tilde{U}$), and since $(1/10) \cdot \sqrt{m^{-5/20}} >> m^{-5/20}$, the set $\tilde{W}$ contains most of these special colorings as well. More precisely, $\tilde{W}$ contains a fraction of at least (say) $(1 - m^{-1/20})$ of the special colorings of $\tilde{U}$. Therefore, in this case, $\tilde{W}$ contains the coloring $C$ with probability of at least $(1 - m^{-1/20})$.

Thus the total probability that $C$ is not contained in $\tilde{W}$ is at most $2 \cdot m^{-1/20} \leq 2/1000$. Hence, with probability of at least $998/1000$, $W$ contains a coloring that colors all the elements of $U$ by 0.

■

This completes the proof of Claim 6.5.

■

By Claim 6.4, and Claim 6.5 it follows that with probability of at least 1/2 we have both:

1. $A_U = A_{[n] \setminus \{n\}}$, and

2. $|B_U| \geq |B|/2^{m+1}$.

Take a set $U$ that satisfies both. Since $A_U = A_{[n]\setminus\{n\}}$, we have by Claim 5.2,

$$Thickness(A_U) \geq Thickness(A) \geq m^{17/20}.$$

Also, since $|A|/|A_{[n]\setminus\{n\}}| = AVDEG_n(A) \leq 8 \cdot m^{19/20}$, we have

$$|A_U| \geq \left(8 \cdot m^{19/20}\right)^{-1} \cdot |A| \geq \left(8 \cdot m^{19/20}\right)^{-1} \cdot 2^{-\alpha} \cdot m^n = 2^{-[\alpha+3-(\log_2 m)/20]} \cdot m^{n-1}.$$

Since $|B_U| \geq |B|/2^{m+1}$, we have

$$|B_U| \geq 2^{-\beta} \cdot 2^{m \cdot n}/2^{m+1} = 2^{-(\beta+1)} \cdot 2^{m\cdot(n-1)}.$$

Thus, $A_U, B_U$ satisfy the required properties, and Lemma 6.2 follows.

■

## 6.3 An Explicit Bound for $COMP_m[\alpha, \beta, k]$, and Proof of the Main Theorem

Lemma 6.1, and Lemma 6.2 immediately give the following recursive bound for $COMP_m[\alpha, \beta, k]$.

**Corollary 6.7** *For any $\alpha, \beta, k, m \geq 0$, with $\beta \leq m^{2/20}$, $k \geq 1$, and $m \geq 1000^{20}$,*

$$\begin{aligned} COMP_m[\alpha,\beta,k] \geq MIN( \quad & COMP_m[\alpha+2,\beta,k]+1, \\ & COMP_m[\alpha,\beta+1,k]+1, \\ & COMP_m[\alpha+3-(\log_2 m)/20, \beta+1, k-1] \quad ). \end{aligned}$$

Using the recursive bound, it is now easy to prove explicit bounds for $COMP_m[\alpha, \beta, k]$ :

**Theorem 6.8** *Denote by $BOUND_m[\alpha, \beta, k]$ the function:*

$$BOUND_m[\alpha,\beta,k] \stackrel{\text{def}}{=} k \cdot [(\log_2 m)/20 - 5]/2 - \alpha/2 - \beta.$$

*Then, for any $\alpha, \beta, k \geq 0$, and $m \geq 1000^{20}$,*

$$COMP_m[\alpha,\beta,k] \geq BOUND_m[\alpha,\beta,k].$$

*Proof.* The proof is by induction, (using Corollary 6.7). Formally, to prove the theorem for $\alpha, \beta, k$, we assume (an induction hypothesis) that the theorem is correct for all $\alpha', \beta', k'$ such that one of the following is satisfied

1. $k' \leq k-1$, **or**
2. $k' = k$ , $\alpha' \geq \alpha$ , $\beta' > \beta$, **or**
3. $k' = k$ , $\alpha' > \alpha$ , $\beta' \geq \beta$.

Since by the definition of $BOUND_m[\alpha, \beta, k]$ we have

$$\begin{aligned} BOUND_m[\alpha, \beta, k] &= BOUND_m[\alpha + 2, \beta, k] + 1, \\ BOUND_m[\alpha, \beta, k] &= BOUND_m[\alpha, \beta + 1, k] + 1, \\ BOUND_m[\alpha, \beta, k] &= BOUND_m[\alpha + 3 - (\log_2 m)/20, \beta + 1, k - 1], \end{aligned}$$

the inductive step is immediate.

The base case $k < 1$ (i.e., $k = 0$), and the base case $\beta \geq m^{2/20}$ are immediate because they imply $BOUND_m[\alpha, \beta, k] \leq 0$.

■

One can now take in Theorem 6.8; $\alpha = 0, \beta = 0$, to get for $m \geq 1000^{20}$,

$$COMP_m[0, 0, k] \geq k \cdot [(\log_2 m)/20 - 5]/2 = k \cdot \Omega(\log m),$$

which proves Theorem 2.1.

## 6.4 The Proof for Multi-Color Games

It is not hard to verify that the proof for Theorem 2.1 generalizes to the case of multi-color games. Hence, the proof for Theorem 2.2 is the same as the one of Theorem 2.1, with the following minor changes.

Fix $r \leq m^{1/1000}$ to be the number of colors. For simplicity of notations, we use the colors $0, ..., r-1$ (rather than $1, ..., r$), and we denote (in this subsection only) by $[r]$ the set $\{0, ..., r-1\}$.

Here, we denote by $Y$ the set $([r]^m)^n$ (rather than the set $(\{0,1\}^m)^n$), and we denote by $R$ a relation in $DART_r(m, n)$ (rather than $DART(m, n)$). The definition of $GAMES_m[\alpha, \beta, k]$ is the same as before, except now $R$ is required to be a relation in $DART_r(m, n)$. We consider the same two cases as before, and we prove the same Lemma 6.1 and Lemma 6.2 (i.e., the statements of these two lemmas do not change). The proof of Lemma 6.1 is the same as before.

To prove Lemma 6.2, we assume (as before) that $SC(R_0) \geq k - 1$. The sets $B'$ and $B_U$ are now defined as subsets of $([r]^m)^{n-1}$ (rather than $(\{0,1\}^m)^{n-1}$). The definition of $B_U$ is the same as before, except now the coloring $w$ is in $[r]^{[m]}$. The proofs of Claim 6.3, and Claim 6.4 are the same as before.

Since $B_U$ is now a subset of $([r]^m)^{n-1}$, Claim 6.5 is now stated in the following way:

*For a random set $U$ of size $m^{5/20}$,*

$$Prob_U [|B_U| \geq |B|/(2 \cdot r^m)] \geq 3/4.$$

To prove the new Claim 6.5 we now define the set $W_{(y_1,...,y_{n-1})}$ as a subset of $[r]^{[m]}$ (rather than $\{0,1\}^{[m]}$). The definition of $W_{(y_1,...,y_{n-1})}$ is the same as before, except now the coloring $w$ is in $[r]^{[m]}$. We denote $\delta = |B|/r^{m \cdot n}$ (rather than $\delta = |B|/2^{m \cdot n}$), and we denote by $\hat{B}$ the set of all $(y_1, \ldots, y_{n-1}) \in ([r]^m)^{n-1}$, with

$$|W_{(y_1,...,y_{n-1})}| \geq (1/4) \cdot \delta \cdot r^m.$$

By the same argument as before, we show that in order to prove the claim it is enough to prove that every element of $\hat{B}$ is contained in $B_U$, with probability of at least 11/12. This is proved (in the same way as before) using the following claim, which is the analogous of Claim 6.6.

**Claim 6.9** *Let $W \subset [r]^m$ be any set of colorings, such that $|W| \geq (1/4) \cdot 2^{-m^{2/20}} \cdot r^m$, and let $U$ be a random subset of $[m]$, of size $m^{5/20}$, then with probability of at least 11/12 there exists a coloring $w \in W$ that colors all the elements of $U$ by 0.*

The proof of Claim 6.9 is the same as the one of Claim 6.6 with the following changes.

The (uniformly distributed) random set $U \subset [m]$ of size $m^{5/20}$ is now picked in the following way:

1. first pick a random set $\tilde{U} \subset [m]$ of size $r \cdot m^{5/20}$,
2. then pick a random coloring $C$ of $\tilde{U}$, with exactly $m^{5/20}$ elements colored 0,
3. finally define $U \subset \tilde{U}$ to be the set of zeros of $C$.

The rest of the proof is as before:
We denote by $\tilde{W}$ the restriction of $W$ to the subset $\tilde{U}$, and as before it follows that with very high probability (over the choice of $\tilde{U}$), the set $\tilde{W}$ contains almost all colorings of $\tilde{U}$.

We then consider colorings of $\tilde{U}$ with exactly $m^{5/20}$ elements colored 0. Since the set of all these colorings is of large fraction in the set of all colorings (of $\tilde{U}$), the set $\tilde{W}$ contains (with high probability) most of these special colorings as well. Therefore, with high probability, $\tilde{W}$ contains the coloring $C$.

# 7 Conclusions

We have shown that for $m$ larger than some polynomial in $n$, the communication complexity of the best protocol for a $DART(m, n)$ game is bounded from below by the communication complexity of the best structured protocol for that game. As a result, we obtained lower bounds for the monotone-depth of several functions.

We claim that our method gives lower bounds for the monotone depth of many other functions. Informally, we argue the following:

1. The monotone Karchmer-Wigderson's games corresponding to many functions can be reduced to dart games.
2. Proving lower bounds for the best structured protocol is usually not hard.

More formally, it is not hard to see that **every** dart game is in fact (a sub-case of) a monotone Karchmer-Wigderson's game for some function !

As for the lower bounds for the best structured protocol, we have already seen several examples were the argument was very easy (or trivial). In general, as mentioned above, given a relation $R$ (with a DNF tautology $F_R$), the structured complexity of $R$ is the same as the depth of the best decision tree for the corresponding DNF-search problem, over the variables $e_1, \ldots, e_n$. As observed by V. Chvatal and E. Szemeredi, this is also the same as the depth of the best regular Resolution proof for $F_R$ (for details see [LoNeNaWi95]).

We can therefore conclude that any lower bound for regular Resolution implies a lower bound for the corresponding dart game. As mentioned in the introduction, lower bounds for monotone complexity were used before to derive lower bounds for propositional proof systems (e.g., for Cutting-Planes and for Resolution). Here, we conclude that the other direction is also possible.

## Acknowledgments

We would like to thank Avi Wigderson for helpful discussions (and in particular for pointing out the connections to regular resolution), and Sasha Razborov for helpful comments. We would like to thank Mike Saks for suggesting the simpler proof for Lemma 5.3.

The first steps leading to the research reported herein were done during a conference in Barbados (1995). Part of this research was done during a conference in Dagstuhl (1997).

## References

[AlBo87] N. Alon and R. Boppana, The monotone circuit complexity of Boolean functions, *Combinatorica* **7(1)** (1987), pp. 1–22.

[AmMa96] K. Amano and A. Maruoka, Potential of the approximation method, *Proc. of the 37th IEEE Symp. on the Foundations of Computer Science* (1996), pp. 431–440.

[An85] A. Andreev, On a method for obtaining lower bounds for the complexity of individual monotone functions, *Dolk. Akad. Nauk. SSSR* **282**(5) (1985), 1033–1037 (in Russian). English translation in: *Soviet Math. Dokl.* **31**(3) (1985), 530–534.

[BaMc91] D. Barrington and P. McKenzie, Oracle branching programs and Logspace versus $P$, *Information and Computation* **95** (1991), pp. 96–115.

[BeUl97] C. Berg and S. Ulfberg, Symmetric approximation arguments for monotone lower bounds without sunflowers, To appear in: *Computational Complexity.*

[BoPiRa95] M. Bonet, T. Pitassi and R. Raz, Lower bounds for cutting planes proofs with small coefficients, *Proc. of the 27th ACM Symp. on the Theory of Computing* (1995), pp. 575–584. Full version to appear in: *Journal of Symbolic Logic.*

[BoSi90] R. Boppana and M. Sipser, The complexity of finite functions, in *Handbook of Theoretical Computer Science: Volume A Algorithms and Complexity*, J. van Leeuwen editeur, MIT Press/Elsevier, 1990, pp. 757–804.

[Co74] S.A. Cook, An observation on time-storage trade-off, J. Computer and Systems Science **9(3)** (1974), pp. 308–316.

[CoHa95] S.A. Cook and A. Haken, Lower bounds for cutting planes proofs and monotone circuit complexity, Preprint 1995.

[EdImRuSg91] J. Edmonds, R. Impagliazzo, S. Rudich and J. Sgall, Communication complexity towards lower bounds on circuit depth, *Proc. of the 32nd IEEE Symp. on the Foundations of Computer Science* (1991), pp. 249–257.

[Fu96] X. Fu, Modular coloring formulas are hard for cutting plane proofs, *Proc. of the 28th ACM Symp. on the Theory of Computing* (1996), pp. 595–602.

[GoHå95] M. Goldmann and J. Håstad, Monotone circuits for connectivity have depth $(\log n)^{2-o(1)}$, *Proc. of the 27th ACM Symp. on the Theory of Computing* (1995), pp. 569–574.

[GrSi92] M. Grigni and M. Sipser, Monotone complexity, in *Boolean function complexity,* ed: M.S. Paterson, London Math. Soc. Lecture Notes Series 169, Cambridge Univ. Press, 1992.

[Ha95] A. Haken, Counting bottlenecks to show monotone P≠NP, *Proc. of the 36th IEEE Symp. on the Foundations of Computer Science* (1995), pp. 36–40.

[ImPiUr94] R. Impagliazzo, T. Pitassi and A. Urquhart, Upper and lower bounds for tree-like cutting planes proofs, *Proceedings of Logic in Computer Science*, (1994).

[JoLa77] N.D. Jones and W.T. Laaser, Complete problems for deterministic polynomial time, Theoretical Computer Science **3** (1977), pp. 105–117.

[Ju97] S. Jukna, Finite limits and monotone computations: the lower bound criterion, Preprint 1997.

[Ka88] M. Karchmer, Communication complexity: a new approach to circuit depth, ACM Doctoral dissertation award 1988, MIT Press (1989).

[KaRaWi91] M. Karchmer, R. Raz and A. Wigderson, On proving super-logarithmic depth lower bounds via the direct sum in communication complexity, *Proceedings of the 6th Annual Symposium on Structure in Complexity Theory*, (1991).

[KaWi88] M. Karchmer and A. Wigderson, Monotone circuits for connectivity require super-logarithmic depth, *Proc. of the 20th ACM Symp. on the Theory of Computing* (1988), pp. 539–550. Full version in: *SIAM J. on Disc. Math.* **3**, no. 2 (1990) pp. 255–265.

[Kr95] J. Krajicek, Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic, To appear in: *Journal of Symbolic Logic.*

[KuNi96] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press.

[LoNeNaWi95] L. Lovasz, I. Newman, M. Naor and A. Wigderson, Search problems in the decision tree model, *SIAM J. on Disc. Math.* Vol 8, (1995) pp. 119–132.

[Pu95] P. Pudlak, Lower bounds for resolution and cutting planes proofs and monotone computation, Preprint 1995.

[Ra85a] A. RAZBOROV, Lower bounds on the monotone complexity of some Boolean function, *Dolk. Akad. Nauk. SSSR* **281**(4) (1985), 598–607 (in Russian). English translation in: *Soviet Math. Dokl.* **31** (1985), 354–357.

[Ra85b] A. RAZBOROV, A lower bound on the monotone network complexity of the logical permanent, *Mat. Zametki* **37**(6) (1985), 887–900 (in Russian). English translation in: *Math. Notes* **37**(6)(1985), 485–493.

[Ra89] A. RAZBOROV, On the method of approximation, *Proc. of the 21th ACM Symp. on the Theory of Computing* (1989), pp. 167–176.

[Ra94] A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the R.A.N.*, 59(1) pp.201–224, 1995.

[RaRu93] A. Razborov and S. Rudich, Natural Proofs, *Proc. of the 26th ACM Symp. on the Theory of Computing* (1994), pp. 204–213.

[RaWi90] R. RAZ AND A. WIGDERSON, Monotone circuits for matching require linear depth, *Proc. of the 22th ACM Symp. on the Theory of Computing* (1990), pp. 287–292. Full version in: *J. of the Association for Computing Machinery* **39** (3), pp. 1992.736–744

[SiTs97] J. SIMON AND S.C. TSAI, A note on the bottleneck counting argument, Preprint 1997.

[Ta88] E. TARDOS, The gap between monotone and non-monotone circuit complexity is exponential, *Combinatorica* **8** (1988), 141–142.

[Ya79] A. YAO, Some complexity questions related to distributive computing, *Proc. of the 11st ACM Symp. on the Theory of Computing* (1979), pp. 209–213.

[Ya94] A. YAO, A lower bound for the monotone depth of connectivity, *Proc. of the 35th IEEE Symp. on the Foundations of Computer Science* (1994), pp. 302–308.