On multi-partition communication complexity

Pavol Ďuriš, Juraj Hromkovič, Stasys Jukna, Martin Sauerhoff, Georg Schnitger

Lehrstuhl für Informatik I, RWTH Aachen, Ahornstraße 55, 52074 Aachen, Germany
Department of Computer Science, Swiss Federal Institute of Technology ETH Zürich, ETH Zentrum RZ F2, CH-8092 Zurich, Switzerland
Institute for Mathematics and Informatics, Akademijos 4, LT-2600 Vilnius, Lithuania
Fachbereich Informatik, Lehrstuhl 2, Universität Dortmund, 44221 Dortmund, Germany
Fachbereich Informatik, Johann Wolfgang Goethe-Universität Frankfurt, Robert-Mayer-Straße 11–15, 60054 Frankfurt am Main, Germany

Received 2 January 2003; revised 5 May 2004
Available online 3 August 2004

Abstract

We study $k$-partition communication protocols, an extension of the standard two-party best-partition model to $k$ input partitions. The main results are as follows.

1. A strong explicit hierarchy on the degree of non-obliviousness is established by proving that, using $k + 1$ partitions instead of $k$ may decrease the communication complexity from $\Theta(n)$ to $\Theta(\log k)$.
2. Certain linear codes are hard for $k$-partition protocols even when $k$ may be exponentially large (in the input size). On the other hand, one can show that all characteristic functions of linear codes are easy for randomized OBDDs.
3. It is proved that there are subfunctions of the triangle-freeness function and the function $\oplus$ $\text{CLIQUE}_3$, that are hard for multi-partition protocols. As an application, strongly exponential lower bounds on the size of nondeterministic read-once branching programs for these functions are obtained, solving an open problem of Razborov [Proceedings of the eighth FCT, LNCS 529, Springer, 1991, pp. 47–60].

© 2004 Elsevier Inc. All rights reserved.

Corresponding author. Fax: +49-231-755-2047.
E-mail addresses: jh@i1.informatik.rwth-aachen.de (J. Hromkovič), martin.sauerhoff@udo.edu (M. Sauerhoff), georg@thi.informatik.uni-frankfurt.de (G. Schnitger).
1. Introduction

One of the hardest tasks in theoretical computer science is to prove nontrivial lower bounds on the amount of computational resources needed to solve explicit computing problems. For many models of computation we observe the phenomenon that the border between oblivious and non-oblivious variants corresponds to the border between “easy” and “hard” for proving lower bounds. We call a model of computation oblivious if it may access its input bits in an order that may depend only on the input length but not on the actual input itself, and non-oblivious if this is not the case.

A nice illustration of this connection between non-obliviousness and hardness of proving lower bounds is provided by finite automata. Clearly, one-way finite automata are an oblivious model of computation and there is no problem in proving tight, large lower bounds on their size (the number of states) and so, for instance, to obtain an exponential gap between determinism and nondeterminism for some specific regular languages. In contrast, two-way finite automata are non-oblivious and one is so far not able to prove satisfying lower bounds on their size. In particular, proving an exponential gap between the sizes of two-way deterministic and two-way non-deterministic finite automata is a long-standing open problem [23]. In 1979, Sipser restricted two-way finite automata to so-called sweeping automata and proved an exponential gap between determinism and nondeterminism for this restricted model [26]. But the crucial point is that sweeping automata are an oblivious version of two-way automata and this kind of obliviousness can exponentially increase the number of states [18] (i.e., the lower bound proof technique for sweeping automata cannot successfully be used for general two-way finite automata). As a further source of examples illustrating the relationship between non-obliviousness and hardness of proving lower bounds we mention the area of branching programs. More details on this model will be given in the next section. For a thorough introduction we refer the reader to the monograph [27].

The above facts show that, to get better lower bound techniques for non-oblivious models of computation, it is worthwhile to study the dependence of computational complexity on the degree of non-obliviousness allowed in the model under consideration. In this paper, we follow this line of research for two-party communication protocols. The main reason for considering this model is the simplicity of its description and the fact that communication complexity has become one of the most successful instruments in proving lower bounds on other fundamental complexity measures in the last 20 years (see, e.g. [9,10,16] for surveys). Moreover, the standard models of deterministic, nondeterministic, and randomized two-party communication protocols are well understood and one has developed a powerful mathematical machinery for estimating the communication complexity of specific problems.

In the following, we summarize the definitions of deterministic and nondeterministic two-party communication protocols in the form required here. Let \( f \) be a boolean function defined on a set \( X \) of \( n \) boolean variables and let \( \Pi = (X_1, X_2) \) be a partition of \( X \), i.e., \( X_1 \cup X_2 = X \) and \( X_1 \cap X_2 = \emptyset \). A deterministic two-party communication protocol \( P \) for \( f \) with respect to \( \Pi \) is an algorithm by
which two players, called Alice and Bob, can evaluate \( f \) as follows. At the beginning of the computation, Alice obtains an input \( x : X_1 \rightarrow \{0, 1\} \) and Bob an input \( y : X_2 \rightarrow \{0, 1\} \). Then the players communicate according to \( P \) by alternatingly exchanging messages. The message computed by a player at some stage of the protocol may be viewed as a function of his or her respective input and all the previously exchanged messages. The players may use unbounded resources to compute their messages. The message sent by the last player is the output of the protocol, which has to agree with \( f(x, y) \). The cost of \( P \) on input \( (x, y) \) is the total number of bits exchanged during the computation on \( (x, y) \). The cost of \( P \) is the maximum of the cost of \( P \) on \( (x, y) \) over all inputs \( (x, y) \in \{0, 1\}^{\vert X_1 \vert} \times \{0, 1\}^{\vert X_2 \vert} \). The communication complexity of \( f \) with respect to \( \Pi \), \( cc_\Pi(f) \), is the minimum cost of a two-party protocol \( P \) for \( f \) with respect to \( \Pi \). Finally, we may also allow to adaptively choose the partition \( \Pi \) from a restricted class of partitions. For a constant \( \beta > 0 \) call the partition \( \Pi \) \( \beta \)-balanced if \( \vert X_1 \vert, \vert X_2 \vert \geq \lceil \beta n \rceil \) and just balanced if it is \( (1/2) \)-balanced. We define the \( (\text{best}-\text{partition}) \) communication complexity of \( f \), \( cc(f) \) as the minimum of \( cc_\Pi(f) \) over all balanced partitions \( \Pi \).

A nondeterministic protocol allows each player to access a (private) string of nondeterministic bits as an additional input. Such a protocol computes the function \( f \) if there is an assignment to the nondeterministic bits such that the protocol outputs 1 if and only if \( f(x, y) = 1 \). The complexity of a nondeterministic protocol \( P \) is the maximum of the number of exchanged bits taken over all inputs, including the nondeterministic bits. The nondeterministic communication complexity of \( f \) with respect to \( \Pi \), \( ncc_\Pi(f) \), and the \( (\text{best}-\text{partition}) \) nondeterministic communication complexity of \( f \), \( ncc(f) \), are defined analogously to the deterministic case. For the following, it is important to mention an alternative, combinatorial characterization of nondeterministic communication complexity. For a partition \( \Pi = (X_1, X_2) \) of the input variables, a \( (\text{combinatorial}) \) rectangle (with respect to \( \Pi \)) is a function \( r : \{0, 1\}^n \rightarrow \{0, 1\} \) that can be written as \( r = r^{(1)} \land r^{(2)} \), where the functions \( r^{(1)}, r^{(2)} : \{0, 1\}^n \rightarrow \{0, 1\} \) only depend on the variables in \( X_1 \) and \( X_2 \), respectively. A collection of such rectangles \( r_1, \ldots, r_t \) with respect to \( \Pi \) is said to form a rectangle cover with respect to \( \Pi \) of a boolean function \( f \) defined on \( X \) if \( f = r_1 \lor \cdots \lor r_t \). It is a well-known fact [9,16] that each nondeterministic communication protocol \( P \) for \( f \) with respect to a partition \( \Pi \) using \( m \) bits of communication yields a rectangle cover of \( f \) with respect to \( \Pi \) with \( 2^m \) rectangles and vice versa. In particular, \( ncc_\Pi(f) \) is equal to the logarithm (rounded up) of the minimum number of rectangles in a rectangle cover of \( f \) with respect to \( \Pi \).

We may regard two-party communication protocols as an oblivious model because they work with a fixed partition of the set of input variables for all inputs. Thus it is not surprising that a straightforward application of communication complexity for proving lower bounds only works for oblivious models of computation. As an example, we mention the situation for branching programs, where the first exponential lower bounds on the size using communication complexity have been for the oblivious variant of the model (Alon and Maass [3], see [12] for a generalized variant of their approach and [27] for a more detailed history of results). As an important step on the way to lower bounds for more general variants of branching programs, Okolnishnikova [20] and Borodin et al. [6] succeeded in deriving exponential lower bounds on the size of the non-oblivious models of deterministic and nondeterministic syntactic read-\( k \) branching programs, respectively. From the perspective of communication complexity theory, their approach leads to protocols that may use several different input partitions. More precisely, the idea is that such a protocol is allowed to choose nondeterministically between \( k \) different subprotocols according to the standard definition which may each use a different partition of the inputs. Then the number \( k \) is a natural
measure for the degree of non-obliviousness allowed in the model. If \( f \) is the function we want to compute, we require that for each input \( x \) there is a subprotocol that outputs 1 for this input if and only if \( f(x) = 1 \). This model has been introduced in [11], where the subprotocols were deterministic. Here we allow the subprotocols even to be nondeterministic and arrive at the following formal definition.

**Definition 1.** Let \( f \) be a boolean function defined on a set \( X \) of boolean variables, and let \( k \) be a positive integer. Let \( \Pi_1, \ldots, \Pi_k \) be partitions of \( X \). A \( k \)-partition protocol \( P \) for \( f \) with respect to \( \Pi_1, \ldots, \Pi_k \) is a collection of \( k \) nondeterministic protocols \( P_1, \ldots, P_k \) with \( f = f_1 \lor \cdots \lor f_k \), where the protocol \( P_i \) uses the partition \( \Pi_i \) and computes the function \( f_i \). Let \( c_i \) be the number of rectangles in the rectangle cover of \( f_i \) induced by \( P_i \). Then the complexity of \( P \) is defined as \( \lceil \log \sum_{i=1}^{k} c_i \rceil \). The \( k \)-partition communication complexity of \( f \), \( k\text{-pcc}(f) \), is the minimum of the complexity of a \( k \)-partition protocol for \( f \) with respect to \( \Pi_1, \ldots, \Pi_k \) taken over all collections \( \Pi_1, \ldots, \Pi_k \) of balanced partitions. The multi-partition communication complexity of \( f \) is \( \text{mpcc}(f) = \min_{k \in \mathbb{N}} k\text{-pcc}(f) \).

The paper of Borodin et al. [6] implicitly contains the first nontrivial lower bounds on multi-partition communication complexity. They considered the so-called clique-only function on \( n = \binom{m}{2} \) variables checking whether a graph on \( m \) vertices consists of an \( m/2 \)-clique and \( m/2 \) isolated vertices and proved that this function requires multi-partition communication complexity at least \( \Omega(n^{1/2}) \). Furthermore, they obtained a linear lower bound on the multi-partition communication complexity for functions checking whether the inner product with respect to generalized Fourier transform matrices is equal to zero. The latter bound was in fact even for a generalization of multi-partition protocols with covers of the input variables that do not overlap too much instead of partitions (see [15], we do not treat this model here), and thus allowed to obtain even exponential lower bounds on the size of syntactic read-\( k \) branching programs for not too large \( k \).

The goal of this paper is to study the influence of the degree of non-obliviousness measured in terms of the number of partitions \( k \) on the \( k \)-partition communication complexity (more precisely, we compare \( k\text{-pcc}(f) \) and \( k'\text{-pcc}(f) \) for \( k < k' \)) to prove new lower bounds on the fundamental measure \( \text{mpcc}(f) \), and to apply these results to branching programs. Our main results are as follows.

1. In [11], it was shown for an explicitly defined sequence of boolean functions \( f_n : \{0, 1\}^n \to \{0, 1\} \) that \( \text{ncc}(f_n) = 1\text{-pcc}(f_n) = \Theta(n) \), while \( 2\text{-pcc}(f_n) = O(1) \). In Section 3 (Theorem 8), we significantly extend this result by proving that for all polynomial-time computable functions \( k : \mathbb{N} \to \mathbb{N} \) there is a sequence of boolean functions \( f_{k,n} : \{0, 1\}^n \to \{0, 1\} \) whose corresponding language \( L = \bigcup_{n \in \mathbb{N}} f_{k,n}^{-1}(1) \) is contained in \( P \) and for which

\[
k(n)\text{-pcc}(f_{k,n}) = \Omega(n) \quad \text{and} \quad (k(n)+1)\text{-pcc}(f_{k,n}) = O(\log k(n)).
\]

In particular, the gap between the bounds is unbounded for constant \( k \) and still exponential for \( k(n) \) polynomial in \( n \). Thus, a small increase of the degree of non-obliviousness can result in a huge decrease of communication complexity.

2. In Section 4, we observe that an argument from [13,20] yields the lower bound \( \Omega(n^{1/2}) \) on the multi-partition communication complexity of the characteristic function of a BCH code of length \( n \) and designed distance \( d = 2t + 1 \) with \( t \approx n^{1/2} \) (Theorem 15). Furthermore, we show that the
characteristic function of a random linear code even requires linear multi-partition communication complexity (Theorem 18). On the other hand, the characteristic function of the complement of a linear code can be computed by small randomized OBDDs with arbitrarily small one-sided error (Theorem 20). Thus we obtain the apparently best known tradeoff between randomized and nondeterministic branching program complexity.

3. In Section 5, we consider the problem of determining whether a given graph on \( m \) vertices has no triangles. The corresponding triangle-freeness function \( A_n \) has \( n = \binom{m}{2} \) boolean variables, one for each potential edge, and accepts a given graph if and only if it has no triangles. We prove that there is a subfunction \( A'_n \) of \( A_n \) with \( \text{mpcc}(A'_n) = \Omega(n) \) (Theorem 22).

Although this result does not imply a lower bound on the multi-partition communication complexity of the triangle-freeness function \( A_n \) itself, it has an interesting consequence for nondeterministic read-once branching programs. Razborov ([22], Problem 11) asked whether a strongly exponential lower bound holds for the function \( \oplus \text{CLIQUE}_{3,n} \) on \( n = \binom{m}{2} \) variables that outputs the parity of the number of triangles in a graph on \( m \) vertices. In the case of deterministic read-once branching programs, such a lower bound for \( \oplus \text{CLIQUE}_{3,n} \) was proved by Ajtai et al. [2]. We solve this problem by proving that nondeterministic read-once branching programs for \( \oplus \text{CLIQUE}_{3,n} \) and for the triangle-freeness function \( A_n \) require size at least \( 2^{\Omega(n)} \). The only other strongly exponential lower bounds for nondeterministic read-once programs so far were proved for a class of functions based on quadratic forms in [4–6]. In the deterministic case, the celebrated result of Ajtai [1] gave a strongly exponential lower bound for a function similar to \( \oplus \text{CLIQUE}_{3,n} \) even for linear-length branching programs, which was subsequently improved by Beame, Saks, Sun, and Vee [5] to work also for the randomized case and slightly super-linear length.

Remark 1. Building on the results of this paper presented in the conference version, the following additional results have recently been achieved in [15]: (i) \( \text{mpcc}(A_n) = \Omega(n^{3/4}) \); (ii) \( k\text{-pcc} (A_n) = \Omega(n) \) provided that \( k \leq 2^{c\sqrt{n}} \) for a sufficiently small constant \( c > 0 \); and (iii) there is a constant \( c > 0 \) such that nondeterministic syntactic read-\( k \) branching programs detecting the absence of 4-cliques in a graph on \( m \) vertices require size at least \( 2^{\Omega(m^2/c^k)} \). Moreover, it has been shown that the lower bound on the multi-partition communication complexity of the triangle-freeness function remains true also for protocols that use \( \beta \)-balanced partitions, where \( \beta \) is any constant with \( 0 < \beta \leq 1/2 \).

The rest of the paper is organized as follows. In Section 2, we provide some further motivation why multi-partition communication complexity is a natural and fundamental measure by characterizing it combinatorially in terms of the size of rectangle covers and by discussing its relationship to usual nondeterministic communication complexity and to branching program complexity. In Sections 3, 4, and 5, we present the main contributions of the work in the order described above.

2. Relations between multi-partition communication complexity and other complexity measures

In this section, we discuss the relationship of multi-partition communication complexity to rectangle cover complexity, best-partition nondeterministic communication complexity, and to branching program complexity.
We start with a characterization of multi-partition communication complexity in terms of the number of rectangles needed to cover the ones of the considered function, in analogy to the standard model of nondeterministic communication complexity with respect to a single partition. We rely on this characterization for our lower bound proofs on multi-partition communication complexity.

Given a boolean function \( f \) defined on a set of variables \( X \), we define its \( ( \)multi-partition) rectangle complexity \( R(f) \) as the minimal number \( t \) for which there exist \( t \) rectangles \( r_1, r_2, \ldots, r_t \), which may each have its own balanced partition of the variables in \( X \), such that \( f = r_1 \lor r_2 \lor \cdots \lor r_t \). The \( k \)-partition rectangle complexity \( R_k(f) \) of \( f \) is the minimal number of rectangles needed to cover \( f \) under the restriction that these rectangles may use at most \( k \) different balanced partitions. Note that
\[
R_k(f) = \min_{f_1, f_2, \ldots, f_k} R_1(f_1) + R_1(f_2) + \cdots + R_1(f_k),
\]
where the minimum is taken over all \( k \)-tuples of boolean functions \( f_1, f_2, \ldots, f_k \) with \( f_1 \lor f_2 \lor \cdots \lor f_k = f \). Furthermore, \( R(f) = \min_k R_k(f) \). The definitions directly imply the following:

**Proposition 2.** For all boolean functions \( f \),
\[
\lceil \log R_k(f) \rceil = k - \text{pcc}(f) \quad \text{and} \quad \lceil \log R(f) \rceil = \text{mpcc}(f).
\]

This gives us the following obvious approach for proving lower bounds on multi-partition communication complexity.

**Proposition 3.** Let \( f \) be a boolean function defined on the variable set \( X \). Suppose that any rectangle \( r \) with respect to a balanced partition of \( X \) and with \( r \leq f \), i.e., with \( r^{-1}(1) \subseteq f^{-1}(1) \), accepts at most \( b \) inputs. Then \( \text{mpcc}(f) = \lceil \log R(f) \rceil \geq \lceil \log((f^{-1}(1))/b) \rceil \).

Proposition 2 includes the fact that \( \text{ncc}(f) = \lceil \log R_1(f) \rceil = 1 - \text{pcc}(f) \) as a special case. Apart from this, \( \text{ncc}(f) \) is also related to \( \text{mpcc}(f) \) in a deeper and somewhat surprising way which we describe now. We show that, analogously to \( \text{ncc}(f) \), the measure \( \text{mpcc}(f) \) can be characterized in terms of the rectangle size bound from communication complexity theory [16].

Let \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) be a boolean function, \( A \subseteq f^{-1}(1) \), and let \( \Pi \) be a partition of the variables of \( f \). Define the distribution \( \mu_A \) on \( \{0, 1\}^n \) by \( \mu_A(x) = \lvert A \rvert^{-1} \) if \( x \in A \), and \( \mu_A(x) = 0 \) otherwise. Define the rectangle size bound for \( f \) (with respect to \( A \) and \( \Pi \)) as
\[
B_{A, \Pi}^1(f) = \log(1/\max_r \mu_A(r^{-1}(1))),
\]
where the maximum extends over all rectangles \( r \) with respect to \( \Pi \) with \( r \leq f \).

We have \( \text{ncc}(f) = \max_{A \subseteq f^{-1}(1)} B_{A, \Pi}^1(f) + O(\log n) \) by the proof of Theorem 2.16 in [16], and consequently
\[
\text{ncc}(f) = \min_{\Pi} \max_{A \subseteq f^{-1}(1)} B_{A, \Pi}^1(f) + O(\log n),
\]
where the minimum extends over all balanced partitions \( \Pi \) of the variables of \( f \). A similar argument yields the following characterization of multi-partition communication complexity:
**Proposition 4.** For every boolean function \( f : \{0, 1\}^n \to \{0, 1\}, \)

\[
mpcc(f) = \max_{A \subseteq f^{-1}(1)} \min_{\Pi} B^I_{A,\Pi}(f) + O(\log n).
\]

**Proof.** Due to Proposition 2, it is sufficient to prove that

\[
R(f) \geq \max_{A \subseteq f^{-1}(1)} \min_{\Pi} B^I_{A,\Pi}(f) \quad \text{and} \quad R(f) = O(n) \cdot \max_{A \subseteq f^{-1}(1)} \min_{\Pi} B^I_{A,\Pi}(f).
\]

We first prove (1). Choose \( A \subseteq f^{-1}(1) \) arbitrarily. Let \( c = R(f) \). By averaging, there is a rectangle \( r_0 \leq f \) with respect to a balanced partition \( \Pi_0 \) of the variables of \( f \) such that \( |r_0^{-1}(1) \cap A| \geq |A|/c \).

Since \( B^I_{A,\Pi_0}(f) \) is the minimum of \( |A|/|r^{-1}(1) \cap A| \) over all rectangles \( r \leq f \) with respect to \( \Pi_0 \), it follows that \( B^I_{A,\Pi_0}(f) \leq |A|/|r_0^{-1}(1) \cap A| \leq c \). Hence, \( R(f) \leq \min \{ B^I_{A,\Pi_0}(f) \} \leq 2^{B^I_{A,\Pi}(f)} \leq 2^{B^I_{A,\Pi}(f)} \leq c \). Since \( A \subseteq f^{-1}(1) \) has been chosen arbitrarily, inequality (1) follows.

Now we prove (2). The proof is analogous to that of Theorem 2.16 in [16]. We choose a sequence of rectangles \( r_0, \ldots, r_{c-1} \) such that \( f = r_0 \lor \cdots \lor r_{c-1} \) by the greedy method. Let \( A_0 = f^{-1}(1) \). For \( i \geq 1 \), let \( A_i \) be the set of accepted inputs of \( f \) not covered by \( r_0, \ldots, r_{i-1} \). For \( i \geq 0 \) such that \( |A_i| \geq 1 \), choose \( r_i \) such that it has maximal measure \( \mu_{A_i} \) among rectangles \( r \leq f \) such that \( \mu_{A_i}(r^{-1}(1)) = \max_{r \leq f} \mu_{A_i}(r^{-1}(1)) \), where the maxima are taken over all balanced partitions \( \Pi \) of the input variables and all rectangles \( r \leq f \) according to \( \Pi \), respectively. Let \( B = \max_{A \subseteq f^{-1}(1)} \min_{\Pi} B^I_{A,\Pi}(f) \).

By the choice of \( r_i \),

\[
|A_{i+1}|/|A_i| = 1 - \mu_{A_i}(r_i^{-1}(1)) = 1 - \max_{\Pi} \mu_{A_i}(r_i^{-1}(1))
\]

\[
= 1 - 1/(\min_{\Pi} B^I_{A_i,\Pi}(f)) \leq 1 - 1/B.
\]

Since \( |A_0| \leq 2n \), it follows that \( |A_i| \leq 2^{2n}(1-1/B)^i \) for all \( i \geq 0 \). Using that \((1-1/B)^i \leq e^{-i/B}\), we get \( |A_i| < 1 \) for \( i > \ln(2^{2n}) \cdot B \). Thus there is a \( c = O(n) \cdot B \) such that \( f = r_0 \lor \cdots \lor r_{c-1} \) and we have proved inequality (2). \( \square \)

In the remainder of the section, we introduce the model of branching programs and some of its restricted variants that occur in this paper and discuss their relationship to multi-partition protocols.

**Definition 5.** A (deterministic) branching program on the variable set \( X = \{x_1, \ldots, x_n\} \) is a directed acyclic graph with a source and two sinks. The sinks are labeled by the constants 0 and 1, respectively. Each interior node is labeled by a variable from \( X \) and has two outgoing edges carrying labels 0 and 1, respectively. This graph computes a boolean function \( f \) defined on \( X \) as follows. To compute \( f(a) \) for some input \( a = (a_1, \ldots, a_n) \in \{0, 1\}^n \), start at the source. For an interior node labeled by \( x_i \), follow the edge labeled by \( a_i \) (this is called a test of the variable). Iterate this until a sink is reached, whose label gives the value \( f(a) \). For a fixed input \( a \), the sequence of nodes visited in this way is
The computation path for a uniquely determined and is called the computation path for \(a\). The size \(|G|\) of a BP \(G\) is the number of its nodes. The branching program size of a function \(f\) is the minimum size of a branching program that computes it.

The following variants of branching programs are important for this paper.

**Definition 6.**

- A branching program is called syntactic read-\(k\) if, for each variable \(x_i\), each of the paths in the branching program contains at most \(k\) nodes labeled by \(x_i\). For the case \(k = 1\) we use the name read-once branching program.
- An ordered binary decision diagram (OBDD) is a read-once branching program where on each computation path the variables are tested according to the same order.

We only remark that for the more general model of semantic read-\(k\) branching programs (not considered here) the restriction on the number of read accesses to the variables is required to hold only for all computation paths instead of all graph theoretical paths as above.

Nondeterministic branching programs and randomized branching programs are defined by allowing nodes labeled with variables from an additional set of nondeterministic or randomized variables, respectively. The value of these variables are chosen nondeterministically or by independent coin tosses, respectively. For randomized branching programs, acceptance with different types of error, e.g., one-sided and two-sided error, are defined as usual for Turing machines and communication protocols.

Multi-partition communication complexity allows to capture the essence of the technique of Borodin et al. [6] for proving lower bounds on the size of nondeterministic read-once branching programs. By the results in their paper, it follows that for every boolean function \(f\) nondeterministic read-once branching programs require size at least \(2^{	ext{mpcc}(f)} / 4\). This bound can slightly be improved by additional ideas from the paper [20] of Okolnischenikova to get:

**Proposition 7 ([6,20]).** For every boolean function \(f\) on \(n\) variables the size of a nondeterministic read-once branching program representing \(f\) is at least \(2^{	ext{mpcc}(f)} / (2n)\).

The above proposition may be generalized to syntactic read-\(k\) branching programs by considering generalized multi-partition protocols that work with covers of the input variables that do not overlap too much instead of partitions [15]. Since we do not prove any results for this case, we refrain from discussing the technical details.

### 3. A strong hierarchy on the degree of non-obliviousness

The goal of this section is to prove that allowing one more partition of the input variables can lead to an unbounded decrease of the communication complexity for explicitly defined functions. This represents the strongest possible influence of the degree of non-obliviousness on the complexity.
Theorem 8. For all polynomial-time computable functions \( k : \mathbb{N} \to \mathbb{N} \), there is a sequence of boolean functions \( f_{k,n} : \{0,1\}^n \to \{0,1\} \) such that the language \( L = \bigcup_{n \in \mathbb{N}} f_{k,n}^{-1}(1) \) is contained in \( P \),

\[
k(n)\text{-pcc} (f_{k,n}) = \Omega(n), \quad \text{and} \quad (k(n)+1)\text{-pcc} (f_{k,n}) = O(\log k(n)).
\]

Furthermore, the upper bound can even be achieved by using \((k(n)+1)\)-partition protocols where each subprotocol is deterministic.

Observe that, for any boolean function \( f \) and any \( k, k\text{-pcc} (f) \geq \lceil \log k \rceil \). Hence, the above statement is obviously true if \( k(n) = 2^{o(n)} \), since then \( k(n)\text{-pcc} (f) = \Theta(n) \) and \((k(n)+1)\text{-pcc} (f) = \Theta(n) \). We get a non-trivial gap as soon as \( k(n) = 2^{o(n)} \).

We first explain how the functions used in the proof of Theorem 8 are constructed. We start with some function \( h \) that is known to be hard for multi-partition protocols even if arbitrarily many \( \beta \)-balanced partitions are allowed, for a suitable constant \( \beta \) with \( 0 < \beta \leq 1/2 \). From \( h \) and a carefully chosen collection of partitions \( P = (\Pi_1, \ldots, \Pi_{k+1}) \) of the variables of \( h \), a new function is constructed that requires the evaluation of \( h \) on one half of each of the partitions in \( P \) and is thus easy for \((k+1)\)-partition protocols. Using the properties of \( P \), we then show that, on the other hand, any \( k \)-partition protocol for this function is forced to split the variables of \( h \) more or less evenly between the halves of its partitions and thus requires large complexity. More formally, we have the following definition.

Definition 9. Let \( k, \ell, \) and \( m \) be positive integers such that \( \lceil \log(k+1) \rceil \leq \ell \), and let \( h : \{0,1\}^m \to \{0,1\} \) be an arbitrary function. Let \( x = (x_1, \ldots, x_{2m}), y = (y_0, \ldots, y_{\ell-1}), \) and \( z = (z_0, \ldots, z_{\ell-1}) \) be vectors of boolean variables. Let \( P = (\Pi_1, \ldots, \Pi_{k+1}) \), where \( \Pi_i = (\Pi_{i,1}, \Pi_{i,2}) \) is a balanced partition of the variables in the vector \( x \). Let \( F_{h,\ell,P}(x, y, z) \) be the boolean function in \( 2(m+\ell) \) variables whose value on input \((x, y, z)\) is the value of \( h \) on the part of \( x \) corresponding to the first half of the \( i \)th partition \( \Pi_{i,1} \) if the number \( i \) represented in binary by \( y \) is contained in \( \{1,2,\ldots,k+1\} \), and let \( F_{h,\ell,P}(x, y, z) = 0 \) otherwise.

Observe that \( F_{h,\ell,P} \) does neither depend on the variables in \( x \) that only appear in the second halves of the partitions in \( P \) nor on the variables in \( z \). The latter are dummy variables only used for padding the input. It is obvious that, for any \( h \) and \( P, F_{h,\ell,P} \) has \((k+1)\)-partition protocols of small complexity:

Lemma 10. For any \( h \) and any collection \( P = (\Pi_1, \ldots, \Pi_{k+1}) \) of balanced partitions of the variables of \( h \), \((k+1)\text{-pcc} (F_{h,\ell,P}) = O(\log k) \). The upper bound is achieved by \((k+1)\)-partition protocols where each subprotocol is deterministic.

Proof. The protocol for \( F_{h,\ell,P} \) uses \( k+1 \) partitions which divide the \( x \)-vector of input variables between the two players according to the partitions in \( P \), and which give all \( y \)-variables to the first player and all \( z \)-variables to the second player. In the \( i \)th subprotocol, the first player outputs the value of \( h \) on the variables in the first half of the \( i \)th partition in \( P \) if \( i \) is the value represented by the \( y \)-variables, and 0 otherwise. The second player does nothing. The complexity of the whole protocol is obviously \( \lceil \log(2(k+1)) \rceil = \lceil \log(k+1) \rceil + 1 \). □
In the following, we describe the main combinatorial idea for the proof of the lower bound on the complexity of \((k + 1)\)-partition protocols for \(F_{h,k,P}\). If we can ensure that all the sets occurring as halves of partitions in \(P\) (where \(|P| = k + 1\)) are “very different,” then the partitions in \(P\) cannot be “approximated” by only \(k\) partitions, as the following lemma shows. For this, define the Hamming distance between two finite sets \(A, B\) by \(|A \cap B| + |\overline{A} \cap \overline{B}|\).

**Lemma 11.** Let \(D, m \geq 1\) be integers. Let \(A\) and \(B\) be families of subsets of \([1, \ldots, 2m]\) with \(|A| = m\) for all \(A \in A, D \leq d(A, A') \leq 2m - D\) for all different \(A, A' \in A\), and \(|B| - m| \leq D/4\) for all \(B \in B\). If \(|A| \geq |B| + 1\), then there exists an \(A_0 \in A\) such that for all \(B \in B\),

\[
|A_0 \cap B| \geq D/8 \quad \text{and} \quad |A_0 \cap \overline{B}| \geq D/8.
\]

**Proof.** We first show that there is an \(A_0 \in A\) such that \(D/2 \leq d(A_0, B) \leq 2m - D/2\) for all \(B \in B\). Assume to the contrary that for each \(A \in A\) there is a \(B \in B\) such that \(d(A, B) < D/2\) or \(d(\overline{A}, B) = 2m - d(A, B) < D/2\). Since \(|A| > |B| + 1\), the pigeonhole principle implies that there exists \(B \in B\) such that \(d(S_1, B) < D/2\) and \(d(S_2, B) < D/2\) for some \(S_1 \in \{A_1, \overline{A_1}\}, S_2 \in \{A_2, \overline{A_2}\}\) and \(A_1, A_2 \in A\). But then \(d(S_1, S_2) \leq d(S_1, B) + d(B, S_2) < D\), a contradiction to the hypothesis of the lemma.

For any two sets \(A\) and \(B\), we have \(d(A, B) = |A| + |B| - 2|A \cap B|\). Thus, for the above \(A_0\) and all \(B \in B\),

\[
|A_0 \cap B| = \frac{1}{2} \left( |A_0| + |B| - d(A_0, B) \right) \geq \frac{1}{2} \left( m + m - \frac{D}{4} - \left( 2m - \frac{D}{2} \right) \right) = \frac{D}{8}.
\]

Analogously, we get \(|A_0 \cap \overline{B}| \geq D/8\) for all \(B \in B\). \(\square\)

The next lemma shows how we apply the above combinatorial idea to multi-partition protocols to prove the lower bound in Theorem 8.

**Lemma 12.** Let \(k\) and \(m\) be positive integers. Let \(h\) be a boolean function in \(m\) variables and let \(P\) be a collection of \(k + 1\) balanced partitions of \(2m\) variables with the property that the Hamming distance between the first halves of the partitions is at least \(D\) and at most \(2m - D\) for some \(D = \varepsilon n, \varepsilon > 0\). For any positive integer \(\ell\) with \(\lceil \log(k + 1) \rceil \leq \ell \leq D/4\) let \(F = F_{h,k,P}\) be the function described in Definition 9. Then the \(k\)-partition communication complexity of \(h\) with \((\varepsilon/8)\)-balanced partitions does not exceed the \(k\)-partition communication complexity of \(F\).

Thus, the lemma implies a large lower bound on the \(k\)-partition communication complexity of \(F\) if we have a large lower bound on the complexity of multi-partition protocols for \(h\) with \(\beta\)-balanced partitions, \(\beta\) a suitable constant with \(0 < \beta \leq 1/2\).

**Proof.** Recall that \(F\) is defined on \(n = 2(m + \ell)\) variables in the vectors \(x, y, z\). Let \(x\) be split into halves \((x^{(0)}_{1}, x^{(1)}_{1}), \ldots, (x^{(0)}_{k+1}, x^{(1)}_{k+1})\) according to the partitions in \(P\). Let \(F^*\) be an optimal \(k\)-partition protocol for \(F\) according to some balanced partitions \(\Pi^*_1, \ldots, \Pi^*_k\) of the input variables of \(F\), where \(\Pi^*_i = (\Pi^{i,1}, \Pi^{i,2})\).
For \( i \in \{1, \ldots, k + 1\} \), let \( S_i \) and \( \overline{S_i} \) denote the sets of variables in \( x_i^{(1)} \) and \( x_i^{(2)} \), respectively. For \( i \in \{1, \ldots, k\} \), let \( T_i \) and \( \overline{T_i} \) be the sets of \( x \)-variables contained in \( \Pi_{i,1}^i \) and \( \Pi_{i,2}^i \), respectively. Since the number of the \( y \)- and \( z \)-variables together is \( 2\ell \) and \( \ell \leq D/4 \) by the hypothesis, the number of \( x \)-variables in each half of \( \Pi_{i,1}^i \) is at least \( n/2 - 2\ell = m - \ell \geq m - D/4 \). Hence, \(|T_i|, |\overline{T_i}| \geq m - D/4 \).

We apply Lemma 11 to \( A = \{S_i \mid i = 1, \ldots, k + 1\} \) and \( B = \{T_i \mid i = 1, \ldots, k\} \). This yields an index \( i_0 \in \{1, \ldots, k + 1\} \) with \(|S_{i_0} \cap T_j|, |S_{i_0} \cap \overline{T_j}| \geq D/8 = (\varepsilon/8)m \) for all \( j = 1, \ldots, k \).

We construct the desired \( k \)-partition protocol \( P \) for \( h \) by setting variables to constants in the given protocol \( P^* \) for \( F \). Let \( F = P^*_1 \lor \cdots \lor P^*_k \), where \( P^*_i \) is the function computed by the \( i \)-th sub-protocol \( P^*_i \) of \( P^* \). We fix the \( y \)-variables such that \( y \) represents the value \( i_0 \). Furthermore, we fix the variables in \( S_{i_0} \) and the \( z \)-variables in an arbitrary way.

Let \( P \) and \( P_1, \ldots, P_k \) be the protocols obtained from \( P^* \) and \( P^*_1, \ldots, P^*_k \), respectively, by the above variable assignments. The new protocols only work on the \( m \) variables in \( S_{i_0} \), and we have \( P_1 \lor \cdots \lor P_k = h(x^{(1)}) \). By restricting the partitions \( \Pi_{i,1}^i, \ldots, \Pi_{i,2}^i \) to the remaining variables in \( S_{i_0} \), we obtain new partitions \( \Pi_{i,1}^i, \ldots, \Pi_{i,2}^i \), where \( \Pi_{i}^i = (\Pi^i_{i,1} , \Pi^i_{i,2}) \), such that \(|\Pi^i_{i,1}|, |\Pi^i_{i,2}| \geq (\varepsilon/8)m \) for all \( i = 1, \ldots, k \). Each \( P_i \) is a nondeterministic protocol according to \( \Pi_{i}^i \). Altogether, \( P \) is a protocol of the desired type for \( h \) defined on \( S_{i_0} \), and the complexity of \( P \) is bounded from above by the complexity of \( P^* \).

To get a collection of partitions for which we can apply Lemma 12, we rely on results from coding theory. We use the following definitions. A binary code of length \( n \) is a subset of \( \{0,1\}^n \). Such a code is called linear if it is even a subspace of \( \{0,1\}^n \) regarded as a vector space. For two vectors \( x, y \in \{0,1\}^n \), let \( d(x, y) \) denote the Hamming distance between \( x \) and \( y \). By the weight of \( x \in \{0,1\}^n \), denoted by \( w(x) \), we mean the number of ones in \( x \). Finally, for even \( n \) call a code \( C \) balanced if \( w(x) = n/2 \) for each \( x \in C \).

We identify balanced partitions \( \Pi_1, \ldots, \Pi_{k+1} \) of \( 2m \) variables with their characteristic vectors in \( \{0,1\}^{2m} \), where (say) a one indicates a variable from the first half and a zero a variable from the second half. A suitable collection of partitions is then described by a balanced code where the Hamming distance between two different codewords is neither too small nor too large. Furthermore, to make our argument work for a sufficiently large range of values for \( k \), we need a code with \( 2^{m/4} \) codewords. Finally, we have to make sure that the codewords can be efficiently enumerated to be able to argue that the function constructed from this code later on is efficiently computable. The next lemma provides codes satisfying all these requirements.

Lemma 13. Let \( d \geq 2 \) be an integer and let \( m = 2d(2d^2 - 1) \). Then there is a balanced code \( C \subseteq \{0,1\}^{2m} \) satisfying the following: (i) \( D \leq d(x, y) \leq 2m - D \) for all different \( x, y \in C \), with \( D = em \) for some constant \( e \) with \( 1/32 < e < 1 \); (ii) \( |C| \geq 2^{m/4} \); and (iii) there is a fixed order \( \pi \) of the codewords and an algorithm with running time polynomial in \( m \) that on input \( i \in \{1,2,\ldots,|C|\} \) computes the \( i \)-th codeword of \( C \) with respect to \( \pi \).

Proof. Our starting point are Justesen codes, which are a known family of good codes. We refer to [17] for a thorough treatment, but for easier reference also include a definition and the facts about these code used here in Appendix 5.2. Fix an integer \( d \geq 2 \) and let \( m = 2d(2d^2 - 1) \), \( N = 2d^2 - 1 \), and \( K = [N/2] \leq N - 1 \). Let \( J_d \subseteq \{0,1\}^m \) be the \([N,K]\)-Justesen code. This code has at least \( 2^m/4 \) code-
words and there is a constant $\varepsilon$ with $1/32 < \varepsilon < 1$ such that for sufficiently large $d$ each $x \in J_d$ has weight $w(x) \geq \varepsilon m$. Furthermore, following the proof of the lower bound on the weight, e.g., in [17], one can easily show an analogous upper bound, i.e., for sufficiently large $d$ and each $x \in J_d$, $w(x) \leq (1 - \varepsilon)m$. Since $J_d$ is a linear code, the minimum and maximum weight of codewords are equal to the minimum and maximum distance, respectively, of different codewords, and thus we have for all different $x, y \in J_d$ that $\varepsilon m \leq d(x, y) \leq (1 - \varepsilon)m$.

So far, the chosen code is not balanced. To rectify this problem, we double the length of the codewords and balance the codewords by padding them with ones. Let

$$C = \{(x, y) | x \in J_d, y \in \{0, 1\}^m \text{ with } w(y) = 2m - w(x) \} \subseteq \{0, 1\}^{2m}.$$  

Then $C$ is a balanced code with at least $2^{m/4}$ codewords that satisfies $\varepsilon m \leq d(x, y) \leq 2m - \varepsilon m$ for all different $x, y \in C$. Thus, all parameters are as required for the lemma.

Finally, we argue why the codewords of $C$ can be enumerated efficiently. Looking at the definition of $C$ and the definition of Justesen codes (see [17] or the Appendix 5.2), we see that each codeword of $C$ is obtained by first mapping a codeword of a Reed–Muller code of length $N = 2^d - 1 = O(m/\log m)$ to a codeword of $J_d$ of length $m = 2d(2^d - 1)$ and then mapping the latter to a codeword of $C$. Both mappings are one-to-one and computable in polynomial time in $m$. Since the underlying finite field of the Reed–Muller code has order $2^d = O(m/\log m)$, a primitive element of this field can be found in polynomial time in $m$ by exhaustive search. Given this, it is easy to enumerate the elements of the Reed–Muller code efficiently, e.g., using a generator matrix for this code (i.e., a matrix containing a basis of the code as its rows). □

To complete the construction of the functions $F_{h, l, p}$ for the proof of the lower bound, we still need an explicitly defined function $h$ which has large multi-partition communication complexity even if the given partitions are only $\beta$-balanced for a constant $\beta$. Linear lower bounds of this type, even for arbitrary constants $\beta$ with $0 < \beta < 1/2$, are provided, e.g., in [4,15]. In [4] this is proved for boolean functions based on quadratic forms with respect to generalized Fourier transform matrices and in [15] for the boolean function detecting the absence of 4-cliques in graphs. Now we are ready to complete the proof of Theorem 8.

**Proof of Theorem 8.** Recall that for $k(n) = 2^{\Theta(n)}$ the claim of the theorem is trivially true. Hence, it suffices to choose any constant $\alpha > 0$ and to show the result for all $k = k(n) \leq 2^{\alpha n}$. Choose $\alpha$ and $k$ such that $[\log (k + 1)] \leq n/2^{12}$.

We now define the functions $f_{k,n}$. We assume that $n$ is a sufficiently large, even integer (obviously, this can be done w.l.o.g. since the result can be extended also to odd $n$ by padding the input). Let $d = \lfloor \log n - \log \log n \rfloor - 3 \geq 2$ and $m = 2d(2^d - 1)$. Then $n/16 \leq m \leq n/4$. Let $r = \lfloor n/2 - (1 + 1/128)m \rfloor > 0$, $m' = m + r$, and $\ell = (1/2)(n - 2m') = n/2 - (m + r)$. Then $\ell \leq n/2 - (m + n/2 - (1 + 1/128)m) = m/128$ and $\ell \geq m/128 - 1 \geq m/256$. Let $C \subseteq \{0, 1\}^{2m}$ be the code obtained from Lemma 13. Define the new code $C' \subseteq \{0, 1\}^{2m'}$ by

$$C' = \{(x, y) | x \in C, y \in \{0, 1\}^{2r} \text{ with } w(y) = r \}.$$  

Then $C'$ is a balanced code with $D \leq d(x, y) \leq 2m' - D$, where $D = \varepsilon m$ and $\varepsilon$ is the constant from Lemma 13 with $1/32 < \varepsilon < 1$, and $|C'| \geq 2^{m/4} \geq 2^{n/64}$. Furthermore, using part (iii) of that lemma,
it follows that there is an order \( \pi \) on \( \{1, 2, \ldots, |C|\} \) such that the \( i \)-th codeword of \( C' \) with respect to \( \pi \) can be computed in time polynomial in \( n \).

Let \( h \) be a boolean function on \( m' \) variables from \([4,15]\) with multi-partition communication complexity \( \Omega(m') \) for \( \beta \)-balanced partitions, where \( \beta \) is an arbitrary constant with \( 0 < \beta \leq 1/2 \). Let \( c_1, \ldots, c_{k+1} \in \{0,1\}^{2m'} \) be the first \( k+1 \) codewords from \( C' \) with respect to \( \pi \); there are enough codewords since \( k+1 \leq 2^{n/212} \leq |C'| \). Define the collection of partitions \( P = (\Pi_1, \ldots, \Pi_{k+1}) \) of the variables \( \{x_1, \ldots, x_{2m'}\} \) with \( \Pi_i = (\Pi_{i,1}, \Pi_{i,2}), i = 1, \ldots, k+1, \) by \( \Pi_{i,1} = \{x_j | c_{i,j} = 1\} \) and \( \Pi_{i,2} = \{x_j | c_{i,j} = 0\} \). Let \( f_{k,n} = F_{h,\ell,P} \) be the function on \( n = 2(m' + \ell) \) variables obtained for the parameters \( h, \ell, \) and \( P \) according to Definition 9. We observe that \( \ell \geq m/256 \geq n/212 \geq \lceil \log(k+1) \rceil \). The number of \( y \)-variables is thus sufficiently large to encode the numbers \( 1, 2, \ldots, k+1 \). Using that the functions \( k \) and \( h \) as well as the \( i \)-th codeword \( c_i \) of \( C' \) can be computed in time polynomial in \( n \), it follows that the language corresponding to the sequence of functions \( f_{k,n}, n \in \mathbb{N} \), is in \( P \).

The upper bound in the theorem immediately follows from Lemma 10. For the lower bound, we apply Lemma 12. As required in the hypothesis of Lemma 12, we have \( \ell \leq m/128 \leq (\varepsilon/4)m \), where \( \varepsilon > 1/32 \) is the constant from Lemma 13. Due to the choice of \( h \), we know that the multi-partition communication complexity of this function with respect to \( (\varepsilon/8) \)-balanced partitions is linear in its input length \( m' = m + r = \Omega(n) \). By Lemma 12, this also implies that \( k\text{-pcc} (f_{k,n}) = \Omega(n) \). \( \Box \)

4. The multi-partition communication complexity of linear codes

In this section, we investigate the multi-partition communication complexity of the characteristic function of linear codes. Define the distance of a code as the minimum Hamming distance between any two different codewords belonging to this code. The following lemma is implicit in \([13,20]\), where a stronger version has been used to show that syntactic read-\( k \)-branching programs for the characteristic functions of certain linear codes require exponential size.

Lemma 14 ([13,20]). Let \( C \subseteq \{0,1\}^n \) be an arbitrary (not necessarily linear) code of distance \( 2t + 1 \) with characteristic function \( f_C \). Then

\[
mpcc(f_C) \geq \log \left( |C| \cdot \left( \frac{|n/2|}{t} \right)^2 \cdot 2^{-n} \right).
\]

For the sake of completeness, we include the easy proof of this lemma.

Proof. Let \( \Pi = (X_1,X_2) \) be any balanced partition of the \( n \) variables of \( f_C \). Let \( r = r^{(1)} \land r^{(2)} \) be a rectangle with respect to \( \Pi \) such that \( r \leq f_C \). By Proposition 3, it is sufficient to show that \( r^{-1}(1) \) cannot contain more than \( 2^n/B(t)^2 \) inputs in \( f_C^{-1}(1) = C \), where \( B(t) = \sum_{i=0}^{t} \binom{|n/2|}{i} \) is the number of vectors in the Hamming ball of radius \( t \) in \([0,1]^{n/2}\). This follows directly from the fact that any two different inputs in \( f_C^{-1}(1) \) must differ in at least \( d = 2t + 1 \) bits. If \( r(a,b) = 1 \) for any pair of assignments \( a, b \) to the variables in \( X_1 \) and \( X_2 \), respectively, then we can conclude for all inputs \( b' \neq b \)}}
of Hamming distance at most $d$ from $b$ that $f_C(a, b') = 0$ and thus (since $r \leq f_C$) also $r(a, b') = 0$. This implies that $|(r^{(2)})^{-1}(1)| \leq 2^{2^s} / B(i)$. Since we analogously get $|(r^{(1)})^{-1}(1)| \leq 2^{2^s} / B(i)$, we are done. □

To give an explicit example, we consider binary BCH codes with length $n = 2^m - 1$ and designed distance $d = 2t + 1$; such a code has at least $2^n / (n + 1)^t$ vectors and distance at least $d$ [17]. Let $BCH_n$ be the characteristic function of such a BCH code with $t = \lceil n^{1/2} \rceil$. Using Lemma 14, we obtain:

Theorem 15. Each multi-partition protocol for the characteristic function of $BCH_n$ has complexity at least $\Omega(n^{1/2})$.

Proof. Using Stirling’s formula, one can easily prove the following estimate for the binomial coefficients occurring in Lemma 14:

$$\binom{n/2}{t} = \frac{1}{e(2\pi)^{1/2} \cdot n^{1/2}} \cdot \left(\frac{e \cdot n^{1/2}}{2}\right)^{n/2} \cdot (1 + o(1)).$$

Thus, $(n/2)_t^2 \geq 2^{an^{1/2} \cdot n^{(1/2)n^{1/2}}}$, for some positive constant $a < \log(e/2)$ (where $\log(e/2) > 0.442$).

By Lemma 14, we obtain the following lower bound on the multi-partition communication complexity of the characteristic function of the considered BCH code:

$$\log\left(|C| \cdot \binom{n/2}{t} \cdot 2^{-n}\right) \geq \log\left(\frac{2^{an^{1/2} \cdot n^{n^{1/2}}}}{(n + 1)^{n^{1/2}}}\right) = \Omega(n^{1/2}).$$

Lemma 14 has the advantage of working for arbitrary codes, but is not strong enough to give linear lower bounds on the multi-partition communication complexity. However, for linear codes we can use the stronger argument explained in the following. A linear code $C \subseteq \{0, 1\}^n$ can be described by a boolean parity-check matrix $H$ of dimension $m \times n$, $m \leq n$ a suitable integer, which satisfies $H \cdot x \equiv 0 \mod 2$ if and only $x \in C$. Call a boolean $m \times n$ matrix $s$-good if each of its $m \times (n/2)$ submatrices has rank at least $s$.

Lemma 16. Let $C$ be a binary linear code with an $s$-good $m \times n$ parity-check matrix $H$ and characteristic function $f_C$. Then $\text{mpcc}(f_C) \geq 2^{2s-m}$.

Proof. Let $\Pi = (X_1, X_2)$ be a balanced partition of the $n$ variables of $f_C$. Let $r = r^{(1)} \land r^{(2)}$ be a rectangle with respect to $\Pi$ such that $r \leq f_C$. Since we have $|(f_C)^{-1}(1)| \geq 2^{n-m}$, it is sufficient to show that $r$ does not accept more than $2^{n-2s}$ inputs.

To prove this, let $H_1$ and $H_2$ be the $m \times (n/2)$ submatrices of $H$ corresponding to variables from $X_1$ and $X_2$. Hence, for assignments $a, b$ to $X_1$ and $X_2$, respectively, $f(a, b) = 1$ if and only if $H_1 \cdot a + H_2 \cdot b \equiv 0 \mod 2$, implying that $r(a, b) = 1$ if and only if $H_1 \cdot a \equiv H_2 \cdot b \mod 2$. If $b_0$ is fixed, then the vector $w_0 = H_2 \cdot b_0$ is fixed, and $r(a, b_0) = 1$ only if $a$ is a solution of $H_1 \cdot a \equiv w_0 \mod 2$. Due to the fact that $H$ is $s$-good, the matrix $H_1$ has rank at least $s$, and thus we have at most $2^{n/2-s}$ possible solutions $a$. Analogously, if $a_0$ is fixed, then the vector $w_1 = H_1 \cdot a_0$ is fixed and then $r(a_0, b) = 1$.
only if \( b \) is a solution of \( H_2 \cdot b \equiv w_1 \mod 2 \). Moreover, \( r(a_0, b_0) = 1 \) implies that for all pairs \((a, b)\) accepted by \( r \) we have the same column of free coefficients \( w_1 \equiv w_0 \mod 2 \). Thus, \( r \) accepts at most \( 2^{n/2-s} \cdot 2^{n/2-s} = 2^{n-2s} \) inputs.

To obtain a linear lower bound on multi-partition communication complexity by Lemma 16, we need a family of \( m \times n \) matrices that are \( s \)-good for \( s > \alpha m \) and a constant \( \alpha > 1/2 \). We have to leave it as an open problem to come up with an explicit construction of such a family and only show that random matrices have the required property with high probability.

**Proposition 17.** Let \( m \leq n/32 \). Let \( H \) be a random boolean \( m \times n \) matrix. Then \( H \) is \((m-1)\)-good with probability \( 1 - 2^{-\Omega(n)} \).

**Proof.** Let \( v_1, \ldots, v_n \in \{0, 1\}^m \) be vectors whose entries are determined by independent, fair coin tosses. Let \( H \) be the random boolean matrix with \( v_1, \ldots, v_n \) as column vectors. Our goal is to show that, with high probability, every subset of \( n/2 \) vectors from \( v_1, \ldots, v_n \) spans a space of dimension at least \( m - 1 \). This is not the case if and only if the following event happens:

\((*)\) There is a set \( I \subseteq \{1, \ldots, n\} \) with \( |I| = n/2 \) and vectors \( w_1, w_2 \in \{0, 1\}^m - \{0\} \) with \( w_1 \neq w_2 \) such that for all \( i \in I \), \( w_1^T \cdot v_i \equiv w_2^T \cdot v_i \equiv 0 \mod 2 \).

We show that \((*)\) occurs with exponentially small probability. Let \( w_1, w_2 \) be as described in \((*)\), and let \( X_i \) be the indicator random variable for the event that \( w_1^T \cdot v_i \equiv w_2^T \cdot v_i \equiv 0 \mod 2 \). Since \( E \left[ \sum X_i \right] = n/4 \), Chernoff’s inequality gives us that, for this pair of vectors \( w_1, w_2 \), the event \((*)\) happens with exponentially small probability: For \( \lambda = 1 \), we have

\[
\text{Prob} \left[ \sum_{i=1}^{n} X_i \geq (1 + \lambda) \cdot n/4 \right] \leq e^{-\lambda^2(n/4)/3} = e^{-n/12}.
\]

Since we have fewer than \( \left( \frac{2^m}{2} \right) \leq 2^m \leq 2^{n/16} \) pairs of non-zero vectors \( w_1, w_2 \), the event \((*)\) occurs with probability at most \( 2^{n/16} \cdot e^{-n/12} = 2^{-\Omega(n)} \). □

Combining the above proposition with Lemma 16, we obtain:

**Theorem 18.** With probability \( 1 - 2^{-\Omega(n)} \), the characteristic function of a random binary linear code of length \( n \) has multi-partition communication complexity \( \Omega(n) \).

In the remainder of the section, we derive upper bounds on the complexity the characteristic functions of linear codes. First, we observe that all linear codes have small randomized communication complexity even in the fixed-partition model.

**Proposition 19.** Let \( f_C \) be a characteristic function of a linear binary code of length \( n \). Then the two-party fixed-partition one-round bounded error communication complexity of \( f_C \) is \( O(1) \) with public coins and \( O(\log n) \) with private coins.
Proof. Checking whether a given input is accepted reduces to checking whether the two strings, obtained by Alice and Bob by multiplying the parts of the input they see with the corresponding parts of the parity-check matrix, are equal. Hence, if $H_1$ and $H_2$ are the parts of the parity-check matrix corresponding to the parts of the inputs string $(x, y)$ given to Alice and Bob, then testing whether $f_C(x, y) = 1$ is the same as testing the equality $H_1 \cdot x \equiv H_2 \cdot y \pmod{2}$ of two strings of length at most $n$. $\square$

The characteristic functions $f_C$ of linear codes are known to be hard for different models of branching programs, including nondeterministic syntactic read-$k$ branching programs [13] and $(1, k)$-branching programs [14] (the latter are deterministic branching programs where along each computation path at most $k$ variables are allowed to be tested more than once). On the other hand, the negation $\neg f_C$ is just an OR of at most $n$ scalar products of an input vector with the rows of the corresponding parity-check matrix. Hence, for every linear code, the characteristic function $\neg f_C$ of its complement has a small nondeterministic OBDD. Here we strengthen this observation to randomized OBDDs with one-sided error.

**Theorem 20.** Let $C \subseteq \{0, 1\}^n$ be a linear code and let $f_C$ be its characteristic function. Then, for every integer $r \geq 2$, $\neg f_C$ can be computed by a randomized OBDD of size $O(n^r)$ with one-sided error at most $2^{-r}$.

For the proof of the theorem, we need a technique to reduce the number of random bits that is originally due to Newman [19] and also appeared in different disguises in other papers (see, e.g. [7,8,19,24]). Although the main trick is quite simple, it is usually hidden behind the technical details of a particular model of computation. Since the argument may be of independent interest, it makes sense to formulate it as a separate combinatorial lemma about the average density of boolean matrices.

**Lemma 21.** Let $M,N$ be positive integers with $M = 2^{o(\sqrt{N})}$. Let $A$ be a boolean $M \times N$ matrix with the property that the average density, i.e., the average number of ones, in each row does not exceed $p$, $0 \leq p < 1$. Then, for every constant $\delta > 0$, there is a set $I \subseteq \{1, \ldots, N\}$ with $|I| = 3\lceil \log(2M/\delta^2) \rceil$ such that in the submatrix of $A$ consisting of the columns with index in $I$, each row has average density at most $p + \delta$.

Proof. Let $\xi_1, \ldots, \xi_t$ be independent random variables which are uniformly distributed over $\{1, \ldots, N\}$, where $t = 3\lceil \log(2M/\delta^2) \rceil$. First, observe that with probability $1 - \binom{t}{2}/N = 1 - o(1)$, all $\xi_1, \ldots, \xi_t$ are distinct. Next, fix a row $x = (x_1, \ldots, x_N)$ of $A$ and consider the 0-1 random variables $X_i = x_{\xi_i}$, for $i = 1, \ldots, t$. We have $\text{Prob}[X_i = 1] \leq p$ for all $i$. By Chernoff bounds, the average density $(\sum_{i=1}^t X_i)/t$ of ones in $x$ exceeds $p + \delta$ with probability at most $e^{-\delta^2/3} \leq (2M)^{-\log e}$. Thus, with probability at least $1 - M \cdot (2M)^{-\log e}$, the restriction of each row of $A$ to the columns with indices $\xi_1, \ldots, \xi_t$ has density at most $p + \delta$. This probability is larger than 0 for all positive integers $M$. Altogether, the probability that the submatrix consisting of the columns with indices $\xi_1, \ldots, \xi_t$ has the claimed properties is larger than 0. $\square$
We can now prove the desired upper bound on the size of randomized OBDDs for the characteristic functions of linear codes.

**Proof of Theorem 20.** Let $H$ be a parity-check matrix of $C$ with dimension $m \times n$. Let $w$ be chosen uniformly at random from $\{0,1\}^n$. The essence of the construction is the simple fact that

$$w^\top Hx \equiv 0 \mod 2,$$

for $x \in C$,

whereas

$$\text{Prob}[w^\top Hx \not\equiv 0 \mod 2] = 1/2,$$

for $x \not\in C$.

We cannot use this representation of $f_C$ directly to construct a randomized OBDD, since this OBDD would require exponentially many randomized variables to randomly choose the vector $w$.

To reduce the required number of randomized variables, we apply Lemma 21. Choose the set of all $x \in \{0,1\}^n$ with $\neg f_C(x) = 1$, i.e., $x \not\in C$, as the row indices, and all vectors $w \in \{0,1\}^n$ as the column indices of the $(2^n - |C|) \times 2^n$ matrix $A = (a_{x,w})$. Let

$$a_{x,w} = \begin{cases} 1 & \text{if } w^\top Hx \not\equiv 0 \mod 2, \\ 0 & \text{otherwise.} \end{cases}$$

Then each row of $A$ has density $1/2$. For $M = 2^n - |C| \leq 2^n$ and each constant $\delta > 0$, the lemma gives us a set $W_\delta \subseteq \{0,1\}^n$ with

$$|W_\delta| = 3\left\lceil \log(2M/\delta^2) \right\rceil = O(n/\delta^2)$$

such that, for all $x$ with $\neg f_C(x) = 1$ and $w$ chosen uniformly at random from $W$, we have

$$\text{Prob}[w^\top Hx \not\equiv 0 \mod 2] \geq 1/2 - \delta.$$ 

Choose $\delta = 1/5$. Let $G$ be the randomized OBDD which starts with a tree on $\lceil \log |W_\delta| \rceil$ randomized variables at the top by which an element $w \in W_\delta$ chosen uniformly at random. At the leaf of the tree belonging to the vector $w$, append a deterministic sub-OBDD that checks whether $w^\top Hx \equiv 0 \mod 2$. By the above facts, this randomized OBDD computes $\neg f_C$ with one-sided error at most $7/10$. The size of $G$ is bounded by $O(n^2)$.

To decrease the error probability, we use probability amplification as described in [24]. We regard $G$ as a deterministic OBDD on all variables (deterministic and randomized ones). Applying the known efficient OBDD-algorithms (see, e.g. [27]), we obtain an OBDD $G'$ for the OR of $2r$ copies of $G$ with different sets of randomized variables. This OBDD $G'$ has one-sided error at most $(7/10)^{2r} < 2^{-r}$ and size $O(n^{4r})$. □

Apparently, this result gives the strongest known tradeoff between nondeterministic and randomized branching program complexity.
5. A lower bound for triangle-freeness

Let \( x = (x_{i,j})_{1 \leq i < j \leq m} \) be a vector of \( n = \binom{m}{2} \) boolean variables that are used to encode a graph \( G(x) \) on \( m \) vertices by setting \( x_{i,j} = 1 \) if the edge \( \{i, j\} \) is present and \( x_{i,j} = 0 \) otherwise. The triangle-freeness function \( \Delta_n \) is defined on \( x \) by \( \Delta_n(x) = 1 \) if \( G(x) \) contains a triangle and \( \Delta_n(x) = 0 \) otherwise. The function \( \oplus \text{CLIQUE}_3 \) has the same set of variables and on input \( x \) outputs the parity of the number of triangles in \( G(x) \). In this section, we prove the following result.

**Theorem 22.** There is a subfunction \( \Delta'_n \) of \( \Delta_n \) such that \( mpcc(\Delta'_n) = \Omega(n) \). The same holds also for \( \oplus \text{CLIQUE}_3 \).

This result is sufficient to prove that each nondeterministic read-once branching program detecting the triangle-freeness of a graph requires strongly exponential size. Since by assigning constants to some variables, we can only decrease the branching program size, the desired lower bound on the size of any nondeterministic read-once branching program computing \( \Delta_n \) follows directly from Theorem 22 and Proposition 7. We obtain the following main result which also answers Problem 11 of Razborov [22].

**Theorem 23.** Nondeterministic read-once branching programs for the triangle-freeness function \( \Delta_n \) as well as for \( \oplus \text{CLIQUE}_3 \) require size \( 2^{\Omega(n)} \).

In remainder of the section, we prove Theorem 22.

5.1. Statement and application of the main combinatorial Lemma

For simplicity, we concentrate on \( \Delta_n \) first. We handle \( \oplus \text{CLIQUE}_3 \) analogously later on. We observe that setting variables of \( \Delta_n \) to 0 or to 1 means that edges are forbidden or are required to be present. Each subfunction thus corresponds to a subfamily of all graphs on \( m \) vertices. We carefully choose such a subfamily of all graphs and prove that detecting the absence of triangles is already hard for this subfamily. We consider graphs on \( m \) vertices partitioned into sets \( U = \{1, \ldots, m/2\} \) and \( V = \{m/2 + 1, \ldots, m\} \) (w.l.o.g., assume that \( m \) is even). By a probabilistic argument, we choose triangle-free subgraphs \( G_U \) and \( G_V \) on the vertices in \( U \) and \( V \), respectively, and fix the variables of \( \Delta_n \) in the sets \( X_U = \{x_{i,j} \mid i, j \in U, i < j\} \) and \( X_V = \{x_{i,j} \mid i, j \in V, i < j\} \) accordingly. This yields the desired subfunction \( \Delta'_n \) that only depends on the variables in \( X_{U \times V} = \{x_{i,j} \mid i \in U, j \in V\} \). The number of remaining variables is still \( m^2/4 \) and thus linear in the input size.

For the following combinatorial arguments, it is rather inconvenient to argue about families of graphs or subfunctions. Instead, we look at the single graph on \( m \) vertices that is obtained as the union of \( G_U, G_V \) and the complete bipartite graph \( G_B = U \times V \). We then have to keep in mind that the edges in \( G_B \) in fact correspond to the variables of our subfunction. A multi-partition protocol for \( \Delta'_n \) works according to balanced partitions of the variables in \( X_{U \times V} \) which correspond to balanced partitions of the edges in \( G_B \).

A test is a pair of edges from \( G_B \) that form a triangle together with an edge from \( G_U \cup G_V \). Two tests are said to collide if a triangle can be formed by picking one edge from the first test, one edge from the second test, and an edge from \( G_U \cup G_V \). In particular, tests collide if they share an edge.
For a balanced partition \( \Pi \) of \( G_B \), call a test split by \( \Pi \) if its two edges belong to different halves of \( \Pi \). Ideally, we would like to ensure by the choice of the graphs \( G_U \) and \( G_V \) that for any balanced partition \( \Pi \) of \( G_B \) there is a large, collision-free set of tests that are split by \( \Pi \). Then the variables belonging to these tests could be fixed independently, and any multi-partition protocol for \( A'_n \) would require large complexity already to check that all these tests do not generate any triangle. We cannot obtain the desired properties for any balanced partition of \( G_B \), but surprisingly, we can still show something quite close to that.

**Lemma 24.** There exist triangle-free graphs \( G_U \) and \( G_V \) and constants \( \alpha, \beta > 0 \) such that for all balanced partitions \( \Pi_1, \ldots, \Pi_k \) of \( G_B = U \times V \) with \( k \leq 2^{am^2} \), the graph \( G = G_U \cup G_V \cup G_B \) has a set \( T \) of tests without collisions such that for each \( i \in \{1, \ldots, k\} \) there are at least \( \beta m^2 \) tests in \( T \) that are split by \( \Pi_i \).

The proof of this central combinatorial lemma is deferred to the next subsection. Here we show how it implies Theorem 22.

**Proof of Theorem 22.** We first present the proof for the subfunction \( A'_n \) of \( A_n \). Choose \( G_U \) and \( G_V \) according to Lemma 24 and let \( A''_n \) be the resulting subfunction on \( X_{U \times V} \). Let \( \alpha, \beta > 0 \) be the constants from the lemma. It is sufficient to prove that \( R_k(A''_n) = 2^\Theta(m^2) \) for \( k \) with

\[
\log k \leq \min\{am^2, (\beta/2)m^2\}.
\]

Let functions \( f_1, \ldots, f_k \) be given with \( A''_n = f_1 \lor \cdots \lor f_k \) and \( \sum_{i=1}^k R_i(f_i) = R_k(A''_n) \). Let \( \Pi_1, \ldots, \Pi_k \) be the partitions corresponding to optimal covers of \( f_1, \ldots, f_k \) by rectangles.

We construct a set \( A \) of hard 1-inputs for \( A''_n \) which will already require many rectangles to be covered according to the partitions \( \Pi_1, \ldots, \Pi_k \). Let \( T \) be the set of tests obtained by Lemma 24. For all inputs in \( A \), variables belonging to edges outside of \( T \) are fixed to 0. For each test in \( T \), we then choose exactly one edge and set the respective variable to 1, the second one is set to 0. Thus, the graph corresponding to an input in \( A \) has precisely one of the two edges of each test in \( T \), and two graphs differ only on edges in \( T \). Since the tests in \( T \) do not collide, the graphs are triangle-free and we obtain a total of \( 2^{|T|} \) graphs. Hence, \( |A| = 2^{|T|} \).

For \( i \in \{1, \ldots, k\} \), let \( A_i = (f_i)^{-1}(1) \cap A \). Since \( A_1 \cup \cdots \cup A_k = A \), there is at least one \( i \) with

\[
|A_i| \geq |A|/k = 2^{|T|}/k.
\]

By Lemma 24, there is a set \( T_i \subseteq T \) of tests with \( |T_i| \geq \beta m^2 \) that are split by the partition \( \Pi_i \). Since there are only \( 2^{|T|} \) assignments in \( A \) which differ in the variables belonging to tests in \( T - T_i \), there is at least one fixed assignment to these variables such that the subset \( B \) of inputs in \( A_i \) consistent with this assignment has size

\[
|B| \geq |A_i|/2^{|T|} \geq 2^{|T_i|}/k \geq 2^{(\beta/2)m^2}.
\]

The last inequality follows from our assumption that \( \log k \leq (\beta/2)m^2 \). Since all the inputs from \( B \) are accepted by \( f_i \), it remains to show that no rectangle \( r \leq f_i \) with the underlying partition \( \Pi_i \) can accept more than one input from \( B \). Assume that \((a, b)\) and \((a', b')\) are two different inputs in \( B \) accepted by \( r \). By the choice of \( B \), they differ in a test \( t = \{e_1, e_2\} \) which is split by \( \Pi_i \), i.e., whose edges belong to different halves of the partition \( \Pi_i \). By the definition of \( A \), exactly one of the two
edges \(e_1\) and \(e_2\) is present in each of the graphs belonging to \((a, b)\) and \((a', b')\), respectively, and these edges are different.

Now, if \(r(a, b) = 1\), then \(r(a, b') = 0\) or \(r(a', b) = 0\) because either the graph corresponding to \((a, b')\) or to \((a', b)\) will contain both edges \(e_1, e_2\), which, together with the corresponding edge of \(G_U\) or \(G_V\), forms a triangle. This is a contradiction to the fact that \(r\) is a rectangle. Altogether (assuming Lemma 24 holds), we have completed the proof of the lower bound for \(\Lambda_n^\#\).

Now we prove the result for \(\oplus \text{CLIQUE}_{3,n}\). We consider the subfunction \(\oplus \text{CLIQUE}_{3,n}^t\) which is obtained from \(\oplus \text{CLIQUE}_{3,n}\) in the same way as \(\Lambda^t_n\) from \(\Lambda_n\). Let \(t = |T|\). For \(x, y \in \{0, 1\}^t\) define

\[\text{IP}_t(x, y) = \sum_{i=1}^{t} x_i y_i \mod 2.\]

Define the set \(A\) of hard inputs for \(\oplus \text{CLIQUE}_{3,n}^t\) as follows: For all \((x, y) \in \text{IP}_t^{-1}(1)\), include the input obtained by setting variables outside of \(T\) to 0 and setting the two edge variables belonging to the \(i\)th test in \(T\) to \(x_i\) and \(y_i\), respectively. Then

\[A \subseteq \oplus \text{CLIQUE}_{3,n}^{-1}(1) \quad \text{and} \quad |A| = |\text{IP}_t^{-1}(1)| = 2^{2t-1} - 2^{t-1} \geq 2^{2t-2}.\]

Analogously to the proof for \(\Lambda_n\), we obtain a set \(A_i\) of inputs covered by the rectangles with respect to a single partition \(\Pi_i\) in a cover of \(\oplus \text{CLIQUE}_{3,n}\) such that \(|A_i| \geq |A|/k \geq 2^{2t-2}/k\). Furthermore, at least \(s \geq \beta m^2\) tests in \(T\) are split with respect to \(\Pi_i\). Since there are at most \(2^{2(t-s)}\) assignments to the variables belonging to tests that are not split by \(\Pi_i\), we get a set \(B\) of inputs in \(A_i\) that all agree on these variables with

\[|B| \geq |A_i|/2^{2(t-s)} \geq 2^{2t-2}/k.\]

The inputs in \(B\) are all accepted by \(\oplus \text{CLIQUE}_{3,n}\). Thus, the parts of the inputs in \(B\) fixing the variables that belong to the \(s\) tests split by \(\Pi_i\) are either all accepted by \(\text{IP}_s\) or are all accepted by \(\neg \text{IP}_s\). Let \(\text{IP}_s\) be defined on the variables \(x_1, \ldots, x_s, y_1, \ldots, y_s\) and let \(r\) be a rectangle with respect to the partition \(\Pi = (\{x_1, \ldots, x_s\}, \{y_1, \ldots, y_s\})\) with \(r \subseteq \text{IP}_s\) or \(r \subseteq \neg \text{IP}_s\). Then \(|r^{-1}(1)| \leq 2^s\) (see, e.g. [16]). This implies that also no rectangle \(r' \subseteq \oplus \text{CLIQUE}_{3,n}^t\) contains more than \(2^s\) inputs from \(B\). Thus, at least \(2^{s-2}/k \geq 2(\beta/2)m^2 - 2\) rectangles are needed to cover \(B\) and the desired lower bound for \(\oplus \text{CLIQUE}_{3,n}^t\) follows. Assuming that Lemma 24 holds, this completes the proof of the theorem. \(\square\)

5.2. Proof of the main combinatorial lemma (Lemma 24)

Recall that a test is a pair of edges in \(G_B = U \times V\) which form a triangle together with an edge in \(G_U\) or \(G_V\), and that a test is split by partition \(\Pi\) if its two edges lie in different halves of \(\Pi\). As the first step in the proof of Lemma 24, we choose the graphs \(G_U\) and \(G_V\). For this, we apply the following lemma.

**Lemma 25.** There exist graphs \(G_U\) and \(G_V\) such that:

(i) each of the graphs \(G_U\) and \(G_V\) has \(\Theta(m)\) edges, at most \(O(l)\) triangles, and at most \(O(m)\) paths of length 2 or 3; and
(ii) for every balanced partition \(\Pi\) of \(G_B = U \times V\), there are \(\Omega(m^2)\) tests which are split by \(\Pi\).
Proof. We prove the existence of the desired graphs by a probabilistic argument. In what follows, let $G_U$ ($G_V$) stand for the random graph on $U$ (respectively, on $V$) obtained by inserting the edges independently at random with probability $p = c/m$ each, for some constant $c > 0$ fixed below. We use Markov’s inequality to show that the graphs $G_U$ and $G_V$ have the properties described in part (i) of the lemma with probability at least $1/2$.

Let $G$ be a random graph on $m/2$ vertices where the edges are inserted independently at random with probability $p = c/m$. We claim that, with probability at least $3/4$, $G$ has $\Theta(m)$ edges, $O(1)$ triangles, and $O(1)$ paths of length 2 and 3.

(a) The expected number of edges in $G$ is $E = p \cdot (m/2)^2 = \Theta(m)$. Using Chernoff bounds, we get that the actual number of edges is smaller than $E/2$ or larger than $(3/2)E$ only with exponential small probability.

(b) The expected number of triangles in $G$ is $E = (m/3) \cdot p^3$. Hence, $G$ has more than $16 \cdot E$ triangles with probability less than $1/16$ by Markov’s inequality.

(c) The expected number of paths of length $k$ in $G$ is $E = (m/2^k) \cdot p^k$, and $G$ has more than $32 \cdot E$ paths of length $k$ with probability less than $1/32$. Thus the bound on the number of paths of length two and three is exceeded with probability at most $1/16$.

Altogether, the conjunction of (a), (b), and (c) holds with probability at least $1 - 3/16 > 3/4$. It follows that, with probability larger than $1/2$, both of the random graphs $G_U$ and $G_V$ have $\Theta(m)$ edges, $O(1)$ triangles, and $O(1)$ paths of length 2 and 3.

It remains to prove that, with probability larger than $1/2$, for every balanced partition of $U \times V$, there are at least $\Omega(m^2)$ tests split by this partition. Let $\Pi$ be such a balanced partition. The partition $\Pi$ distributes the edges in $U \times V$ to two sets of size $m^2/8$ each which are given to the players Alice and Bob. Call a vertex mixed if each of the two players has at least $(1/8) \cdot (m/2)$ edges from $U \times V$ incident to it.

Claim 1. There are $\Omega(m)$ mixed vertices in each of the sets $U$ and $V$.

Proof of Claim 1. We use essentially the same argument as Papadimitriou and Sipser [21]. W.l.o.g., assume that we have at most $\epsilon m$ mixed vertices in $V$, where $\epsilon > 0$ is a sufficiently small constant ($\epsilon < 1/112$ works fine). Call a vertex $v$ an $A$-vertex (respectively, $B$-vertex) if Alice (respectively, Bob) has at least $(7/8) \cdot (m/2)$ edges incident to $v$. Thus, vertices which are neither $A$-nor $B$-vertices are mixed. Observe first that the number of $A$-vertices as well as the number of $B$-vertices in each of the sets $U$ and $V$ is at most $b_{\text{max}} = (4/7) \cdot (m/2)$, since otherwise Alice or Bob would have more than $m^2/8$ edges. On the other hand, the number of $A$-vertices as well as the number of $B$-vertices in $U$ (in $V$) is bounded from below by $b_{\text{min}} = (3/7) \cdot (m/2) - \epsilon m$, since otherwise there would be more than $\epsilon m$ mixed vertices in $U$ (in $V$), contrary to the assumption.

Now more than half of the edges from $A$-vertices in $U$ to $B$-vertices in $V$ belong to Alice, because otherwise there will be an $A$-vertex $u \in U$ such that Alice has at most half of the edges from $u$ to $B$-vertices in $V$, and thus altogether at most

$$\frac{1}{2} \cdot b_{\text{max}} + |V| - b_{\text{min}} \leq \frac{1}{2} \cdot \frac{4}{7} \cdot \frac{m}{2} + \frac{m}{2} - \left( \frac{3}{7} \cdot \frac{m}{2} - \epsilon m \right) \leq \frac{6}{7} \cdot \frac{m}{2} + \epsilon m < \frac{7}{8} \cdot \frac{m}{2}$$
edges incident to \( u \). With the same reasoning, however, more than half of all edges from \( A \)-vertices in \( U \) to \( B \)-vertices in \( V \) belong to Bob. Contradiction. \( \square \)

For each mixed vertex \( u \in U \), let \( V_A(u) \) (\( V_B(u) \)) be the set of vertices \( v \in V \) for which Alice (respectively, Bob) has the edge \{\( u, v \}\}. Since \( u \) is mixed, \(|V_A(u)|,|V_B(u)| \geq (1/8) \cdot (m/2)\). Observe that each edge between \( V_A(u) \) and \( V_B(u) \) leads to a test split by the given partition \( \Pi \).

Claim 2. There is a constant \( c > 0 \) such that for the random graph \( G_V \) on \( m/2 \) vertices obtained by inserting edges independently at random with probability \( p = c/m \), the following event has probability larger than \( 1/2 \): For all pairs of disjoint sets \( S_1, S_2 \subseteq V \) of size at least \( m/16 \) each, the number of edges in \( G_V \) between \( S_1 \) and \( S_2 \) is at least \( p|S_1||S_2|/2 \).

Proof of Claim 2. The expected number of edges between fixed sets of vertices \( S_1 \) and \( S_2 \) is \( p|S_1||S_2|/2 \). By Chernoff bounds, the true number of edges is at least \( p|S_1||S_2|/2 \) with probability at least \( 1 - e^{-c''m} \), where the constant \( c'' > 0 \) can be adjusted by the choice of the constant \( c \) in the definition of \( p \). Since there are at most \( (2^{m/2})^2 = 2^m \) choices for the sets \( S_1, S_2 \subseteq V \), the probability of the described event is at least \( 1 - 2^m \cdot e^{-c''m} \), which is larger than \( 1/2 \) for appropriate \( c'' \). \( \square \)

Fix the constant \( c > 0 \) and \( p = c/m \) such that Claim 2 holds and let \( G_V \) be the resulting random graph. We apply the claim to the sets \( V_A(u), V_B(u) \subseteq V \), where \( u \in U \) is a mixed vertex. Due to Claim 2, the event that, for all balanced partitions \( \Pi \) and all \( \Omega(m) \) mixed vertices \( u \) with respect to \( \Pi \), the respective sets \( V_A(u) \) and \( V_B(u) \) are connected by at least \( p|V_A(u)||V_B(u)|/2 = \Omega(m) \) edges, has probability larger than \( 1/2 \). Thus, with probability larger than \( 1/2 \), for each balanced partition \( \Pi \) there are \( \Omega(m^2) \) tests split by \( \Pi \). This completes the proof of Lemma 25. (Observe that it does not matter whether we carry out the above argument for mixed vertices in \( U \) or \( V \).) \( \square \)

We apply Lemma 25 and fix graphs \( G_U \) and \( G_V \) with the described properties. Since there are only \( O(1) \) triangles, we can remove these triangles without destroying the other properties. Especially, we still have linearly many edges. By property (ii), this pair of graphs produces a set of \( \Omega(m^2) \) split tests for any balanced partition of \( G_B \).

Let \( T_0 \) be the set of all tests induced by \( G_U \) and \( G_V \), and let \( t = |T_0| \) be its size. Since both graphs \( G_U \) and \( G_V \) have \( \Theta(m) \) edges, \( t = \Omega(m^2) \). Using the properties of these graphs stated in Lemma 25 (i), we show that at most \( O(t) \) of all \( \binom{t}{2} \) pairs of tests in \( T_0 \) collide:

Lemma 26. There are at most \( O(t) \) pairs of colliding tests in \( T_0 \).

Proof. We prove the claim by case inspection of all possible situations in which tests may collide. Recall that a test is a pair of edges of the complete bipartite graph \( G_B = U \times V \) which together with an edge from \( G_U \) or \( G_V \) form a triangle. Thus, a test is described by a pair \( (e, v) \), where \( e \) is an edge in \( G_U \) (\( G_V \)) and a vertex \( v \in V \) (\( v \in W \), respectively).

Claim 1. Let \( (e_1, w_1) \) and \( (e_2, w_2) \) describe two colliding tests where \( e_1 \) and \( e_2 \) both belong to \( G_U \) (respectively, where both belong to \( G_V \)). Then at least one of the following conditions applies.
Fig. 1.

(a) \(\{w_1, w_2\}\) is an edge of \(G_V\) (respectively of \(G_U\)) and \(e_1\) and \(e_2\) belong to a \(G_U\)-path (respectively to a \(G_V\)-path) of length two;

(b) \(w_1 = w_2\) and \(e_1\) and \(e_2\) belong to a \(G_U\)-path (respectively to a \(G_V\)-path) of length two or three.

Proof of Claim 1. Assume first that a triangle is formed by picking a \(G_U\)-edge (respectively a \(G_V\)-edge) as the third edge. In this case the two edges in \(G_B\) originate from the same vertex in \(U\) (respectively, \(V\)) which has to be a common endpoint of \(e_1\) and \(e_2\). Thus \(e_1\) and \(e_2\) belong to a \(G_U\)-path (respectively \(G_V\)-path) of length two and \(\{w_1, w_2\}\) is the \(G_V\)-edge (respectively the \(G_U\)-edge) in question. (See Fig. 1A.)

Now assume that the triangle is formed by picking a \(G_U\)-edge (respectively a \(G_V\)-edge) \(e\). Thus the triangle consists of \(e\) and the two edges in \(G_B\): \(w_1 = w_2\) follows. If \(e_1\) and \(e_2\) do not share an endpoint, then \((e_1, e, e_2)\) is a \(G_U\)-path (respectively \(G_V\)-path) of length three (Fig. 1B1). Finally, if \(e_1\) and \(e_2\) share an endpoint, then \((e_1, e_2)\) is a \(G_U\)-path (respectively \(G_V\)-path) of length two (Fig. 1B2). □

Claim 2. Let \((e_1, w_1)\) and \((e_2, w_2)\) describe two colliding tests where \(e_1\) belongs to \(G_U\) and \(e_2\) belongs to \(G_V\) (the situation where \(e_1\) belongs to \(G_V\) and \(e_2\) to \(G_U\) is symmetric). Then at least one of the following conditions applies.

(c) \(w_1\) is an endpoint of \(e_2\) and \(w_2\) is an endpoint of \(e_1\);

(d) \(w_1\) is an endpoint of \(e_2\) and \(e_1\) belongs to a \(G_U\)-path of length two that begins in \(w_2\);

(e) \(w_2\) is an endpoint of \(e_1\) and \(e_2\) belongs to a \(G_V\)-path of length two that begins in \(w_1\).

Proof of Claim 2. There are essentially three different possible situations which are shown in Fig. 2. Obviously, this is exactly what is described in conditions (c)–(e). Condition (e) is symmetric to (d). □
We now estimate the number of colliding pairs of tests by using the above results and Lemma 25, part (i). We show that there are only $O(m^2)$ pairs of tests for which one of the conditions (a)–(e) applies. Since $t = O(m^2)$, this also proves that the number of colliding pairs is of order $O(t)$.

(a) There are only $O(m)$ edges $\{w_1, w_2\}$ in $G_U$ (respectively $G_V$) and $O(m)$ $G_U$-paths ($G_V$-paths) of length two.
(b) There are only $m/2$ vertices $w_1$ and $O(m)$ $G_U$-paths ($G_V$-paths) of length three.
(c) The number of collisions of this type is $2|G_U||G_V| = O(m^2)$, since there are $|G_U||G_V|$ choices for $e_1$ and $e_2$ and two ways to place the endpoints $w_1$ and $w_2$ for each of these choices.
(d) There are $O(m)$ $G_U$-paths of length two and $2|G_V|$ choices for the pair $(e_2, w_1)$.
(e) This is symmetric to (d).

This concludes the proof of Lemma 26. □

Recall that we already have a set of tests $T_0$ of size $t = \Theta(m^2)$ such that each balanced partition of $G_B = U \times V$ has $\Omega(m^2)$ split tests in $T_0$. To finish the proof of Lemma 24, it remains to find constants $\alpha, \beta > 0$ such that for each collection $\Pi_1, \ldots, \Pi_k$ of balanced partitions of $G_B$ with $k \leq 2^{\alpha m^2}$, there is a subset $T \subseteq T_0$ of tests with the following properties:

(i) There is no pair of tests from $T$ which collide; and
(ii) for each $i \in \{1, \ldots, k\}$ there are at least $\beta m^2$ tests in $T$ that are split by $\Pi_i$.

We again use a probabilistic construction. Let $T$ be a set of $u$ tests picked uniformly at random from the set $T_0$, where $u = \gamma t$ and $\gamma$ is a constant with $0 < \gamma < 1$ chosen later on.

**Lemma 27.** There is a constant $\alpha > 0$ such that for all $k \leq 2^{\alpha m^2}$ and for any collection $\Pi_1, \ldots, \Pi_k$ of balanced partitions of $G_B$ such that for each $i \in \{1, \ldots, k\}$ there is a set of at least $s = \Omega(m^2)$ tests in $T_0$ that are split by $\Pi_i$, the following is satisfied.

(i) With probability at least $1/2$, the set $T$ contains at most $O(\alpha^2 / t)$ pairs of colliding tests (where $t = |T_0|$ is the total number of tests).
(ii) With probability larger than $1/2$, for each $i \in \{1, \ldots, k\}$ there are at least $us/(2t)$ tests in $T$ that are split by $\Pi_i$.

**Proof.** Part (i): We define the collision graph to have tests as vertices and edges for each collision. Let $c$ be the number of edges in the collision graph. By Lemma 26, we know that $c = O(t)$.

Let $c_T$ be the number of edges in the subgraph of the collision graph induced by the randomly chosen set $T$. Since we pick tests uniformly at random, the expected number of edges is

$$E[c_T] = \frac{u(u - 1)}{t(t - 1)} \cdot c.$$

By Markov’s inequality, it follows that the actual number of edges is at most $2 \cdot E[c_T]$ with probability at least $1/2$. Hence, the number of pairs of colliding tests in $T$ is at most
2 \cdot \mathbb{E}[e_T] = O((u/t)^2 \cdot c) = O(u^2/t)

with probability at least 1/2.

Part (ii): Consider a fixed partition $\Pi_i$ and let $T_i$ be a set of $s = \Omega(m^2)$ tests in $T_0$ that are split by $\Pi_i$. Then the probability that $T$ contains a test from $T_i$ is $s/t = \Omega(m^2)$ the total number of tests. Thus the expected number of elements in $T \cap T_i$ for a randomly chosen set $T$ of $u$ tests is $u \cdot s/t$. Let $\lambda = 1/2$. By Chernoff bounds, it follows that

$$\mathbb{P}[|T \cap T_i| < (1 - \lambda) \cdot us/t] \leq e^{-\lambda^2(us/t)/2} = e^{-\Omega(u)}.$$ 

Hence, the probability that for each $i \in \{1, \ldots, k\}$ the set $T$ contains at least $(1 - \lambda) \cdot us/t = us/(2t)$ tests split by $\Pi_i$ is at least $1 - k \cdot 2^{-\Omega(u)}$. Since $u = \gamma t = \Theta(m^2)$, this probability is larger than 1/2 for $k \leq 2^{\alpha m^2}$ and $\alpha > 0$ sufficiently small. □

Let $k \leq 2^{\alpha m^2}$ with $\alpha > 0$ the constant from the above lemma. For $i \in \{1, \ldots, k\}$, let $\Pi_i$ be a balanced partitions of $G_B$ and let $T_i$ be a set of $s = \Omega(m^2)$ tests in $T_0$ split by $\Pi_i$. Lemma 27 yields the existence of a set $T \subseteq T_0$ with the following properties:

(i) $|T| = u = \gamma t$;
(ii) there are at most $\delta u^2/t$ pairs of tests in $T$ which collide, $\delta > 0$ some constant; and
(iii) for all $i = 1, \ldots, k$, $|T \cap T_i| \geq us/(2t)$.

By deleting at most $\delta u^2/t$ tests from $T$, we remove all collisions, obtaining a smaller set $T'$. For each $i$, the number of tests in $T'$ split by $\Pi_i$ is still

$$\frac{us}{2t} - \frac{\delta u^2}{t} = \frac{u}{t} \cdot \left(\frac{s}{2} - \delta u\right) = \gamma \cdot \left(\frac{s}{2} - \delta \gamma t\right).$$

Since this number is of order $\Omega(m^2)$ for $\gamma = s/(4\delta t) = O(1)$, there is a suitable constant $\beta > 0$ independent of the choice of the partitions $\Pi_1, \ldots, \Pi_k$ such that the number of tests in $T'$ split by $\Pi_i$ is at least $\beta m^2$. Altogether, we have completed the proof of Lemma 24.

Acknowledgment

We would like to thank Ingo Wegener for helpful discussions and comments on an earlier version of this paper and also gratefully acknowledge many valuable suggestions due to an anonymous referee.

Appendix A. Justesen Codes

For easier reference, we include a definition of Justesen codes and the main facts about these codes used in Section 3. Different from the main text we also consider non-binary codes. A (linear) code of length $n$ over $\mathbb{F}_q$, $q$ a prime power, is a subset (subspace) of $\mathbb{F}_q^n$. 
Definition 1. Let $d$ be a positive integer, $N = 2^d - 1$, and let $\alpha$ be a primitive element of $\mathbb{F}_{2^d}$. Let $K$ be an integer with $1 \leq K \leq N - 1$, and define $D = N - K + 1$. Let $R_{N,K}$ be the $[N,K]$-Reed–Solomon code which is the linear code of length $N$ over $\mathbb{F}_{2^d}$ specified by the parity-check matrix

$$H_{N,K} = \begin{pmatrix}
1 & \alpha & \alpha^2 & \cdots & \alpha^{N-1} \\
1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(N-1)} \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
1 & \alpha^{D-1} & \alpha^{(D-1)2} & \cdots & \alpha^{(D-1)(N-1)}
\end{pmatrix}.$$ 

For $x \in \mathbb{F}_{2^d}$ and $1 \leq i \leq N$, define $c_i(x) = (x, \alpha^i \cdot x)$. For $x = (x_1, \ldots, x_N) \in \mathbb{F}_{2^d}^N$, define $c(x) = (c_1(x_1), \ldots, c_N(x_N))$ and regard this as a vector from $(\mathbb{F}_2)^{2dN}$. The code $J_{N,K} \subseteq \mathbb{F}_{2^d}^{2dN}$ defined by $J_{N,K} = \{c(x) | x \in R_{N,K}\}$ is called $[N,K]$-Justesen code.

The code $R_{N,K}$ is known to have dimension $K$ and distance $D$ [17]. By the above definition, it follows that $J_{N,K}$ is linear and has dimension $mK$. In the main text, we have made use of the following general bounds on the weight (and thus the distance) of these codes.

Theorem 1 (Justesen). Let $d$ be a positive integer and let $0 < R < 1/2$. Let $N = 2^d - 1$, $m = 2dN = 2d(2^d - 1)$, and $K = \lceil R \cdot 2N \rceil \leq N - 1$. Then the Justesen code $J_{N,K}$ has at least $2^{Rm}$ codewords, and for each constant $\varepsilon > 0$, $d$ sufficiently large, and each $x \in J_{N,K}$,

$$\alpha m \leq w(x) \leq m - \alpha m, \text{ with } \alpha = (1 - \varepsilon)(1 - 2R)H^{-1}(1/2),$$

where $H(p) = -p \log p - (1 - p) \log(1-p)$ is the binary entropy function.

The lower bound on the weight of the codewords of a Justesen code stated above is standard in textbooks on coding theory, see, e.g. [17]. The upper bound follows along the same lines.

References


