ONE-WAY COMMUNICATION COMPLEXITY AND THE NEČIPORUK LOWER BOUND ON FORMULA SIZE

HARTMUT KLAUCK†

Abstract. In this paper the Nečiporuk method for proving lower bounds on the size of Boolean formulas is reformulated in terms of one-way communication complexity. We investigate the settings of probabilistic formulas, nondeterministic formulas, and quantum formulas. In all cases we can use results about one-way communication complexity to prove lower bounds on formula size. The main results regarding formula size are as follows: We show a polynomial size gap between probabilistic/quantum and deterministic formulas, a near-quadratic gap between the sizes of nondeterministic formulas with limited access to nondeterministic bits and nondeterministic formulas with access to slightly more such bits, and a near-quadratic lower bound on quantum formula size. Furthermore we give a polynomial separation between the sizes of quantum formulas with and without multiple read random inputs. The lower bound methods for quantum and probabilistic formulas employ a variant of the Nečiporuk bound in terms of the Vapnik–Chervonenkis dimension. To establish our lower bounds we show optimal separations between one-way and two-way protocols for limited nondeterministic and quantum communication complexity, and we show that zero-error quantum one-way communication complexity asymptotically equals deterministic one-way communication complexity for total functions.

Key words. formula size, communication complexity, quantum computing, limited nondeterminism, lower bounds, computational complexity

AMS subject classifications. 68Q17, 68Q10, 81P68, 03D15

DOI. 10.1137/S009753970140004X

1. Introduction. One of the most important goals of complexity theory is to prove lower bounds on the size of Boolean circuits computing some explicit functions. Currently only linear lower bounds for this complexity measure are known. It is well known that superlinear lower bounds are provable, however, if we restrict the circuits to fan-out 1, i.e., if we consider Boolean formulas. The best known technique for providing these is due to Nečiporuk [32]; see also the survey by Boppana and Sipser [7]. It applies to Boolean formulas with arbitrary gates of fan-in 2. For other methods applying to circuits over a less general basis of gates, see again [7]. The largest lower bounds provable with Nečiporuk’s method are of the order $\Theta(n^2/\log n)$.

The complexity measure of formula size is not only interesting because formulas are restricted circuits, which are easier to handle in lower bounds, but also because the logarithm of the formula size is asymptotically equivalent to the circuit depth.

It has become customary to consider randomized algorithms as a standard model of computation. While randomization can be eliminated quite efficiently using the nonuniformity of circuits, randomized circuits are sometimes simpler to describe and
more concise than deterministic circuits. It is natural to ask whether we can prove lower bounds for the size of randomized formulas.

More generally, we like to consider different modes of computation other than randomization. First we are interested in nondeterministic formulas. It turns out that general nondeterministic formulas are as powerful as nondeterministic circuits and thus are intractable for lower bounds with current techniques. But this construction relies heavily on a large consumption of nondeterministic bits guessed by the simulating formula; in other words, such a simulation drastically increases the length of proofs involved in nondeterministic computation. So we can ask whether the size of formulas with a limited number of nondeterministic guesses can be lower bounded, in the spirit of research on limited nondeterminism (for a survey of this topic see [14]).

Finally, we are interested in quantum computing. The model of quantum formulas was introduced by Yao in [41]. He gave a superlinear lower bound for quantum formulas computing the MAJORITY function. Later Roychowdhury and Vatan [37] proved that the classical Nečiporuk bound divided by \( \log n \) applies to quantum formulas and showed a lower bound of the order \( \Omega(n^2 / \log^2 n) \) for an explicit function. They also showed that quantum formulas can actually be simulated quite efficiently by classical Boolean circuits.

The outline of this paper is the following. First we observe that the Nečiporuk method can be defined in terms of one-way communication complexity. While this observation is not relevant for deterministic computations, it becomes useful if we consider other modes of computation. First we consider probabilistic formulas. We derive a variation of the Nečiporuk bound in terms of randomized communication complexity and, using results due to Kremer, Nisan, and Ron [26], a combinatorial variant involving the Vapnik–Chervonenkis (VC) dimension. Applying this lower bound we show a near-quadratic lower bound for probabilistic formula size (Corollary 3.7).

We also exhibit a function for which probabilistic formulas are smaller by a factor of \( \sqrt{n} \) than deterministic formulas and even Las Vegas (zero-error) formulas (Corollary 3.13). This is shown to be the maximal such gap provable under the condition that the lower bound for deterministic formulas is given by the Nečiporuk method. Furthermore we observe that the standard Nečiporuk bound asymptotically also works for Las Vegas formulas.

We then introduce communication complexity type Nečiporuk methods for nondeterministic formulas and for quantum formulas. To apply these generalizations we have to provide lower bounds for one-way communication complexity with limited nondeterminism and for quantum one-way communication complexity. Since the communication problems we investigate are asymmetric (i.e., Bob receives much fewer inputs than Alice), our results show optimal separations between one- and two-round communication complexity for limited nondeterministic and for quantum communication complexity. Such separations have been known previously for deterministic and probabilistic protocols; see [26, 36]. In the quantum case such a separation is given in the equivalent scenario of quantum random access codes in the work of Ambainis et al. [2]. In the case of limited nondeterminism such a separation was unknown prior to this work.

In the nondeterministic case we give a specific combinatorial argument for the communication lower bound (Theorem 5.5). In the quantum case we give a general lower bound method based on the VC dimension (Theorem 5.9), which can also be extended to the case where the players share prior entanglement, as an application of the ideas in [2]. The generalization to protocols with entanglement has also been observed by Nayak in his thesis [31]. Furthermore we show that exact and Las Vegas
quantum one-way communication complexity are never much smaller than deterministic one-way communication complexity for total functions (Theorems 5.11 and 5.12), generalizing a theorem of Hromkovič and Schnitger [19].

Then we are ready to give Nečiporuk-type lower bound methods for nondeterministic formulas and quantum formulas. In the nondeterministic case we show that for an explicit function there is a threshold on the amount of nondeterminism needed for efficient formulas; i.e., a near-quadratic size gap occurs between formulas allowed to make a certain amount of nondeterministic guesses and formulas allowed a logarithmic factor more. The threshold is polynomial in the input length (Theorem 6.4).

For quantum formulas (in Corollary 6.11) we show a lower bound of $\Omega(n^2 / \log n)$, improving by a logarithmic factor on the best previously known bound due to Roychowdhuri and Vatan [37]. More importantly, our bound also applies to a more general model of quantum formulas, which are, e.g., allowed to access multiple read random variables. This feature makes these generalized quantum formulas a proper generalization of both quantum formulas and probabilistic formulas. It turns out that we can give a $\Omega(\sqrt{n} / \log n)$ separation between formulas with multiple read random variables and without this option, even if the former are classical and the latter are quantum (Corollary 6.6). Thus quantum formulas as defined by Yao are not capable of efficiently simulating classical probabilistic formulas. We show that the VC-dimension variant of the Nečiporuk bound holds for generalized quantum formulas and the standard Nečiporuk bound holds for generalized quantum Las Vegas formulas (Theorem 6.10).

The organization of the paper is as follows: In section 2 we describe some preliminaries regarding the VC dimension, classical communication complexity, and Boolean circuits. In section 3 we expose the basic lower bound approach and apply the idea to probabilistic formulas. In section 4 we give more background on quantum computing and information theory. In section 5 we give the lower bounds for nondeterministic and quantum one-way communication complexity. In section 6 we derive our results for nondeterministic and quantum formulas and apply those bounds. In section 7 we give some conclusions.

2. Preliminaries.

2.1. The VC dimension. We start with a useful combinatorial concept [39], the Vapnik-Chervonenkis dimension. This will be employed to derive lower bounds for one-way communication complexity and then to give generalizations of the Nečiporuk lower bound on formula size.

**Definition 2.1.** A set $S$ is shattered by a set of Boolean functions $\mathcal{F}$, if for all $R \subseteq S$ there is a function $f \in \mathcal{F}$, so that for all $x \in S$: $f(x) = 1 \iff x \in R$.

The size of the largest set shattered by $\mathcal{F}$ is called the VC dimension $\text{VC}(\mathcal{F})$ of $\mathcal{F}$.

The following fact [39] will be useful.

**Fact 2.2.** Let $\mathcal{F}$ be a set of Boolean functions $f : X \rightarrow \{0, 1\}$. Then

$$2^{\text{VC}(\mathcal{F})} \leq |\mathcal{F}| \leq (|X| + 1)^{\text{VC}(\mathcal{F})}.$$  

2.2. One-way communication complexity. We now define the model of one-way communication complexity, first described by Yao [40]. Our discussion of this model will be informal; for a more formal treatment of the material and additional background information we refer to the excellent monograph by Kushilevitz and Nisan [27].
Definition 2.3. Let \( f : X \times Y \rightarrow \{0, 1\} \) be a function. Two players Alice and Bob with unrestricted computational power receive inputs \( x \in X, y \in Y \) to the function.

Alice sends a binary encoded message to Bob, who then computes the function value. The complexity of a protocol is the worst case length of the message sent (over all inputs).

The deterministic one-way communication complexity of \( f \), denoted \( D(f) \), is the complexity of an optimal deterministic protocol computing \( f \).

In the case Bob sends one message and Alice announces the result, we use the notation \( D^B(f) \).

The communication matrix of a function \( f \) is the matrix \( M \), with \( M(x, y) = f(x, y) \) for all inputs \( x, y \).

We will consider different modes of acceptance for communication protocols. Let us begin with nondeterminism.

Definition 2.4. In a nondeterministic one-way protocol for a Boolean function \( f : X \times Y \rightarrow \{0, 1\} \) Alice first guesses nondeterministically a sequence of \( s \) bits. Then she sends a message to Bob, depending on the sequence and her own input. Bob computes the function value. Note that the guessed sequence is known only to Alice. In such a protocol an input is accepted, if there is at least one sequence of \( s \) bits, which leads to the output "1" when guessed by Alice. All other inputs are defined as rejected. \( f \) is computed by the protocol, if all input pairs are accepted/rejected correctly by the nondeterministic protocol.

The complexity of a nondeterministic one-way protocol with \( s \) nondeterministic bits is the length of the longest message used.

The nondeterministic communication complexity \( N(f) \) is the complexity of an optimal one-way protocol for \( f \) using arbitrarily many nondeterministic bits.

\( N_s(f) \) denotes the complexity of an optimal nondeterministic protocol for \( f \), which uses at most \( s \) private nondeterministic bits for every input.

Note that if we do not restrict the number of nondeterministic bits, then nondeterministic protocols with more than one round of communication can be simulated without loss: Alice guesses a dialogue and sends this dialogue if it is consistent with her input; Bob checks the same with his input and outputs 1 if this is implied by the dialogue.

While nondeterministic communication is a theoretically motivated model, probabilistic communication is the most powerful realistic model of communication besides quantum mechanical models.

Definition 2.5. In a probabilistic protocol with private random coins Alice and Bob each possess a source of independent random bits that can be used to obtain an arbitrary number of random bits under the uniform distribution. The players are allowed to access that source and communicate depending on their inputs and the random bits they read. We distinguish the following modes of acceptance:

1. In a Las Vegas protocol the players are not allowed to err. They may, however, give up without an output with some probability \( \epsilon \). The complexity of a one-way protocol is the worst case length of a message used by the protocol; the Las Vegas complexity of a function \( f \) is the complexity of an optimal Las Vegas protocol computing \( f \) and is denoted \( R_{0, \epsilon}(f) \).

2. In a probabilistic protocol with bounded error \( \epsilon \) the output has to be correct with probability at least \( 1 - \epsilon \). The complexity of a protocol is the worst case length of the message sent (over all inputs and the random guesses); the complexity of a function is the complexity of an optimal protocol computing
that function and is denoted $R_\epsilon(f)$. For $\epsilon = 1/3$ the notation is abbreviated to $R(f)$.

3. A bounded error protocol has one-sided error, if inputs with $f(x_A, x_B) = 0$ are rejected with certainty.

We also consider probabilistic communication with public randomness. Here the players have access to a shared source of random bits without communicating. This means that both players can read the $i$th bit produced by the random source and thus establish a shared random bit string. Complexity in this model is denoted $R_{\text{pub}}$, with acceptance defined as above.

The difference between probabilistic communication complexity with public and with private random bits is actually only an additive $O(\log n)$ as shown by Newman [33] via an argument based on the nonuniformity of the model.

The following communication problems are frequently considered in the literature about communication complexity.

**Definition 2.6** (disjointness problem). $\text{DISJ}_n(x_1 \ldots x_n, y_1 \ldots y_n) = 1 \iff \forall i : \neg x_i \lor \neg y_i$. The function accepts, if the two sets described by the inputs are disjoint.

**Index function:**

$I\text{X}_n(x_1 \ldots x_n, y_1 \ldots y_n) = 1 \iff x_y = 1$.

The deterministic one-way communication complexity of a function can be characterized as follows. Let $\text{row}(f)$ be the number of different rows in the communication matrix of $f$. Note that in the communication matrix the rows are associated to the inputs of the sender, Alice.

**Fact 2.7.** $D(f) = \lceil \log \text{row}(f) \rceil$.

It is relatively easy to estimate the deterministic one-way communication complexity using this fact. As an example consider the index function; note that obviously $D^B(I\text{X}_n) = \log n$. It is easy to see with Fact 2.7 that $D(I\text{X}_n) = n$, since there are $2^n$ different rows in the communication matrix of $I\text{X}_n$. Kremer, Nisan, and Ron [26] show that also $R_{\text{pub}}(I\text{X}_n) = \Omega(n)$ holds. A bound with a tight constant factor has been obtained by Ambainis et al. [2] using information theory. Similar results were also given by Katz and Trevisan [20].

A general lower bound method for probabilistic one-way communication complexity is shown in [26].

We consider the VC dimension for functions as follows.

**Definition 2.8.** For a function $f : X \times Y \to \{0, 1\}$ let $\mathcal{F} = \{g|\exists x \in X : \forall y \in Y : g(y) = f(x, y)\}$. Then define $VC(f) = VC(\mathcal{F})$.

**Fact 2.9.** $R_{\text{pub}}(f) = \Omega(VC(f))$.

In section 5.2 we will generalize this result to quantum one-way protocols.

With the above definition $[\log |\mathcal{F}|] = D(f)$. Then $VC(f) \leq D(f) \leq [\log(|Y| + 1) \cdot VC(f)]$ due to Fact 2.2.

Las Vegas communication can be quadratically more efficient than deterministic communication in many-round protocols for total functions [27]. For one-way protocols the situation is different as shown by Hromkovič and Schnitger [19].

**Fact 2.10.** For all total functions $f$: $R_{0,1/2}^\text{pub}(f) \geq D(f)/2$.

We will also generalize this result to quantum communication in section 5.2.

**2.3. Circuits and formulas.** We now define the models of Boolean circuits and formulas. Note that we do not consider questions of uniformity of families of such circuits. For the definition of a Boolean circuit we refer to [7]. We consider circuits with fan-in 2. While it is well known that almost all $f : \{0, 1\}^n \to \{0, 1\}$ need circuit
size $\Theta(2^n/n)$ (see, e.g., [7]), superlinear lower bounds for explicit functions are known only for restricted models of circuits.

**Definition 2.11.** A (deterministic) Boolean formula is a Boolean circuit with fan-in 2 and fan-out 1. The Boolean inputs may be read arbitrarily often, the gates are arbitrary, and constants 0,1 may be read.

The size (or length) of a deterministic Boolean formula is the number of its non-constant leaves.

It is possible to show that for Boolean functions the logarithm of the formula size is linearly related to the optimal circuit depth (see [7]).

Probabilistic formulas have been considered in [38, 6, 12] with the purpose of constructing efficient (deterministic) monotone formulas for the majority function in a probabilistic manner.

The standard model of a probabilistic formula is a probability distribution on deterministic formulas. Since such a distribution can give some positive probability to all formulas of the given size, this is not a compact representation of a Boolean function. Hence we consider the following model of probabilistic formulas: “Fair” probabilistic formulas are formulas that read input variables plus additional random variables. The model mentioned before will be called “strong” probabilistic formulas.

**Definition 2.12.** A fair probabilistic formula is a Boolean formula, which works on input variables and additional random variables $r_1, \ldots, r_m$; a strong probabilistic formula is a probability distribution $F$ on deterministic Boolean formulas. Fair (resp., strong) probabilistic formulas $F$ compute a Boolean function $f$ with bounded error, if

$$\Pr[F(x) \neq f(x)] \leq 1/3.$$

Fair (resp., strong) probabilistic formulas $F$ are one-sided error formulas for $f$ (i.e., have one-sided error), if

$$\Pr[F(x) = 0 | f(x) = 1] \leq 1/2 \quad \text{and} \quad \Pr[F(x) = 1 | f(x) = 0] = 0.$$

A Las Vegas formula consists of two Boolean formulas. One formula computes the output; the other (verifying) formula indicates whether the computation of the first can be trusted or not. Both work on the same inputs. There are four different outputs, of which two are interpreted as “?” (the verifying formula rejects), and the other as 0 (resp., 1). A Las Vegas formula $F$ computes $f$, if the outputs 0 and 1 are always correct, and

$$\Pr[F(x) = ?] \leq 1/2.$$

The size of a fair probabilistic formula is the number of its nonconstant leaves; the size of a strong probabilistic formula is the expected size of a deterministic formula according to $F$.

It is easy to see that one can decrease the error probability to arbitrarily small constants, while increasing the size by a constant factor; therefore, we will sometimes allow different error probabilities.

A strong probabilistic formula $F$ can be transformed into a deterministic formula. For one-sided error formulas this increases the size by a factor of $O(n)$: Choose $O(n)$ formulas randomly according to $F$ and connect them by an OR gate. An application of the Chernov inequality proves that the error probability is so small that no errors are possible anymore. Strong formulas with bounded (two-sided) error are derandomized by picking $O(n)$ formulas and connecting them by an approximative majority function. That function outputs 1 on $n$ Boolean variables if at least $2n/3$ have the value 1 and outputs 0 if at most $n/3$ variables have the value 1. An approximative majority
function can be computed by a deterministic formula of size \( O(n^2) \); see [38, 6]. Thus the size increases by a factor of \( O(n^2) \).

Let us remark that strong probabilistic formulas may have sublinear length; this is impossible for fair probabilistic formulas depending on all inputs. As an example, the approximative majority function may be computed by a strong probabilistic formula through picking a random input and outputting its value.

We will later also consider nondeterministic formulas.

**Definition 2.13.** A nondeterministic formula with \( s \) nondeterministic bits is a formula with additional input variables \( a_1, \ldots, a_s \). The formula accepts an input \( x \), if there is a setting of the variables \( a \), so that \((a, x)\) is accepted.

### 3. The general lower bound method and probabilistic formulas.

There are some well known results giving lower bounds for the length of Boolean formulas. The method of Neˇciporuk [32, 7] remains the one giving the largest lower bounds among those methods working for formulas in which all fan-in 2 functions are allowed as gates. For other methods see [7] and [3]: a characterization for formula size with gates AND, OR, NOT using the communication complexity of a certain game is also known (see [27]). For such formulas the largest known lower bound is a near-cubic bound due to H˚astad [15].

Let us first give the standard definition of the Neˇciporuk bound.

Let \( f \) be a Boolean function on the \( n \) variables in \( X = \{x_1, \ldots, x_n\} \). For a subset \( S \subseteq X \) let a subfunction on \( S \) be a function induced by \( f \) by assigning Boolean values to the variables in \( X - S \). The set of all subfunctions on \( S \) is called the set of \( S \) subfunctions of \( f \).

**Fact 3.1 (Neˇciporuk).** Let \( f \) be a Boolean function on \( n \) variables. Let \( S_1, \ldots, S_k \) be a partition of the variables and \( s_i \) the number of \( S_i \) subfunctions on \( f \). Then every deterministic Boolean formulas for \( f \) has size at least

\[
\frac{1}{4} \sum_{i=1}^{k} \log s_i.
\]

It is easy to see that the Neˇciporuk function \((1/4) \sum_{i=1}^{k} \log s_i\) is never larger than \( n^2 / \log n \).

**Definition 3.2.** The function “indirect storage access” (ISA) is defined as follows: There are three blocks of inputs \( U, X, Y \), with \(|U| = \log n - \log \log n \), \(|X| = |Y| = n \). \( U \) addresses a block of length \( \log n \) in \( X \), which addresses a bit in \( Y \). This bit is the output; thus ISA(\( U, X, Y \)) = \( Y_{X_{U}} \).

The following is proved, e.g., in [7, 42].

**Fact 3.3.** Every deterministic formula for ISA has size \( \Omega(n^2 / \log n) \).

There is a deterministic formula for ISA with size \( O(n^2 / \log n) \).

1We are now going to generalize the Neˇciporuk method to probabilistic formulas and later to nondeterministic and quantum formulas. We will use a simple connection to one-way communication complexity and use the guidance obtained by this connection to give lower bounds from lower bounds in communication complexity. In the case of probabilistic formulas we will employ the VC dimension to give lower bounds. Informally speaking we will replace the log of the size of the set of subfunctions by the VC dimension of that set and get a lower bound for probabilistic formulas.

Our lower bounds are valid in the model of strong probabilistic formulas. Corollary 3.7 shows that even strong probabilistic formulas with a two-sided error do not help to decrease the size of formulas for ISA. All upper bounds will be given for fair formulas.
We are going to show that the (standard) Nečiporuk is at most a factor of \(O(\sqrt{n})\) larger than the probabilistic formula size for total functions. Thus the maximal gap we can show using the best known general lower bound method is limited.

On the other hand we describe a Boolean function, for which fair probabilistic formulas with a one-sided error are a factor \(\Theta(\sqrt{n})\) smaller than Las Vegas formulas, as well as a similar gap between one-sided error formulas and two-sided error formulas. The lower bound on Las Vegas formulas uses the new observation that the standard Nečiporuk bound asymptotically also works for Las Vegas formulas.

### 3.1. Lower bounds for probabilistic formulas

We now derive a Nečiporuk-type bound with one-way communication.

**Definition 3.4.** Let \(f\) be a Boolean function of \(n\) Boolean inputs, and let \(y_1 \ldots y_k\) be a partition of the input variables.

We consider \(k\) communication problems for \(i = 1, \ldots, k\). Player Bob receives all inputs in \(y_i\); player Alice receives all other inputs. The deterministic one-way communication complexity of \(f\) under this partition of inputs is called \(D(f_i)\). The public coin bounded error one-way communication complexity of \(f\) under this partition of inputs is called \(R^{pub}(f_i)\).

The probabilistic Nečiporuk function is \((1/4) \sum_i R^{pub}(f_i)\).

It is easy to see that \((1/4) \sum_i D(f_i)\) coincides with the standard Nečiporuk function and is therefore a lower bound for deterministic formula size due to Fact 3.1.

**Theorem 3.5.** The probabilistic Nečiporuk function is a lower bound for the size of strong probabilistic formulas with a bounded error.

**Proof.** We will show for every partition \(y_1, \ldots, y_k\) of the inputs how a strong probabilistic formula \(F\) can be simulated in the \(k\) communication games. Let \(F_i\) be the distribution over deterministic formulas on variables in \(y_i\) induced by picking a deterministic formula as in \(F\) and restricting to the subformula with all leaves labeled by variables in \(y_i\) and containing all paths from these to the root. We want to simulate the formula in game \(i\) so that the probabilistic one-way communication is bounded by the expected number of leaves in \(F_i\).

We are given a probabilistic formula \(F\). The players now pick a deterministic formula \(F'\) induced by \(F\) with their public random bits; player Alice knows all of the inputs except those in \(y_i\). This also fixes a subformula \(F'_i\) drawn from \(F_i\). Actually the players have access only to an arbitrarily large public random string, so the distributions \(F_i\) may be approximated only within arbitrary precision. This alters success probabilities by arbitrarily small values. We disregard these small changes in probability.

Let \(V_i\) contain the vertices in \(F'_i\), which have two predecessors in \(F'_i\), and let \(P_i\) contain all paths, which start in \(V_i\) or at a leaf, and which end in \(V_i\) or at the root, but contain no further vertices from \(V_i\). It suffices, if Alice sends two bits for each such path, which shows whether the last gate of the path computes 0, 1, \(g\), or \(\neg g\), for the function \(g\) computed by the first gate of the path. Then Bob can evaluate the formula alone.

There are at most \(2 |V_i| + 1\) paths as described, since the fan-in of the formula is 2. Thus the overall communication is \(4 |V_i| + 2\). The set of leaves \(L_i\) with variables from \(y_i\) has \(|V_i| + 1\) elements, and thus

\[
R^{pub}(f_i) \leq 4 |V_i| + 2 < 4 |L_i|,
\]

and \((1/4) \sum_i R^{pub}(f_i)\) is a lower bound for the length \(E[\sum_i |L_i|] = \sum_i E[|L_i|]\) of the probabilistic formula.
Let \( VC(f_i) \) denote the VC dimension of the communication problem \( f_i \). We call \( \sum_i VC(f_i) \) the VC–Nečiporuk function.

**Corollary 3.6.** The VC–Nečiporuk function is an asymptotical lower bound for the length of strong probabilistic formulas with a bounded error.

The standard Nečiporuk function is an asymptotical lower bound for the length of strong Las Vegas formulas for total functions.

**Proof.** Using Fact 2.9 the VC dimension is an asymptotical lower bound for the probabilistic public coin bounded error one-way communication complexity.

As in the proof of Theorem 3.5 we may simulate a Las Vegas formula by Las Vegas public coin one-way protocols. Using Fact 2.10 public coin Las Vegas one-way protocols for total functions can be only a constant factor more efficient than optimal deterministic one-way protocols.

According to Fact 3.3 the deterministic formula length of the ISA function from definition 3.2 is \( \Theta(n^2/\log n) \). We now employ our method to show a lower bound of the same order for strong bounded error probabilistic formulas. Thus ISA is an explicit function for which strong probabilism does not allow us to decrease the formula size significantly.

**Corollary 3.7.** Every strong probabilistic formula for the ISA function (with a bounded error) has length \( \Omega(n^2/\log n) \).

**Proof.** ISA has inputs \( Y, X, U \) and computes \( Y_{X_1} \). First we define a partition. We partition the inputs in \( X \) into \( n/\log n \) blocks containing \( \log n \) bits each; all other inputs are in one additional block. In a communication game Alice thus receives all inputs but those in one block of \( X \). Let \( S \) denote the set of possible values of the variables in that block. This set is shattered: Let \( R \subseteq S \) and \( R = \{r_1, \ldots, r_m\} \). Then set the pointer \( U \) to the block of inputs belonging to Bob, and set \( Y_i = 1 \iff i \in R \).

Thus the VC dimension of \( f_i \) is at least \( |S| = n \). Since there are \( n/\log n \) communication games, the result follows.

The next result would be trivial for deterministic or for fair probabilistic formulas, but strong probabilistic formulas can compute functions depending on all inputs in sublinear size. Consider, e.g., the approximate majority function. This partial function can be computed by a strong probabilistic formula of length 1 by picking a random input variable. For total functions on the other hand we have the following.

**Corollary 3.8.** Every strong probabilistic formula which computes a total function depending on \( n \) variables has length \( \Omega(n) \).

**Proof.** We partition the inputs into \( n \) blocks containing one variable each. In a communication game Alice thus receives \( n - 1 \) variables, and Bob receives 1 variable. Since the function depends on both Alice’s and Bob’s inputs, the deterministic communication complexity is at least 1. If the probabilistic one-way communication were 0, the error would be 1/2, and thus the protocol would not compute correctly.

Fact 2.2 shows that for a function \( f : X \times Y \rightarrow \{0,1\} \) it is true that \( D(f) \leq \lceil VC(f) \cdot \log(|Y| + 1) \rceil \). This leads to the following.

**Theorem 3.9.** For all total functions \( f : \{0,1\}^n \rightarrow \{0,1\} \) having a strong probabilistic formula of length \( s \) and for all partitions of the inputs of \( f \):

\[
\frac{\sum D(f_i)}{s} = O(\sqrt{n}).
\]

**Proof.** Obviously \( D(f_i) \leq n \) for all \( i \). Since a partition of the inputs can contain at most \( \sqrt{n} \) blocks with more than \( \sqrt{n} \) variables, these contribute at most \( n\sqrt{n} \) to the Nečiporuk function \( \sum D(f_i) \). All smaller blocks satisfy \( D(f_i) \leq [\sqrt{n} \cdot VC(f_i)] \).
Thus overall \( \sum D(f_i) \leq O(\sqrt{n}(n + \sum VC(f_i))) = O(\sqrt{n}s) \), with Corollary 3.8 and Theorem 3.5.

If a total function has an efficient (say, linear length) probabilistic formula, then the Nečiporuk method does not give near-quadratic lower bounds for the deterministic formula size.

3.2. A function for which probabilism helps. We now describe a function for which one-sided error probabilism helps as much as we can possibly show under the constraint that the lower bound for deterministic formulas is given using the Nečiporuk method. We find such a complexity gap even between strong Las Vegas formulas and fair one-sided error formulas.

**Definition 3.10.** The matrix product function \( MP \) receives two \( n \times n \)-matrices \( T^{(1)}, T^{(2)} \) over \( \mathbb{Z}_2 \) as input and accepts if and only if their product is not the all zero matrix.

**Theorem 3.11.** The \( MP \) function can be computed by a fair one-sided error formula of length \( O(n^2) \).

**Proof.** We use a fingerprinting technique similar to the one used in matrix product verification [30] but adapted to be computable by a formula. First we construct a vector as a fingerprint for each matrix using some random input variables. Then we multiply the fingerprints and obtain a bit. This bit is always zero if the matrix product is zero; otherwise, it is 1 with probability 1/4. Thus we obtain a one-sided error formula.

Let \( r^{(1)}, r^{(2)} \) be random strings of \( n \) bits each. The fingerprints are defined as

\[
F^{(1)}[k] = \bigoplus_{i=1}^{n} r^{(1)}[i]T^{(1)}[i, k] \quad \text{and} \quad F^{(2)}[k] = \bigoplus_{j=1}^{n} T^{(2)}[k, j]r^{(2)}[j].
\]

Then let

\[
b = \bigoplus_{k=1}^{n} F^{(1)}[k] \land F^{(2)}[k].
\]

Obviously \( b \) can be computed by a formula of linear length.

Assume \( T^{(1)}T^{(2)} = 0 \). Then \( b = r^{(1)}T^{(1)}r^{(2)} = 0 \) for all \( r^{(1)} \) and \( r^{(2)} \).

If on the other hand \( T^{(1)}T^{(2)} \neq 0 \), then \( i, j \) exist such that \( \bigoplus_{k=1}^{n} T^{(1)}[i, k]T^{(2)}[k, j] = 1 \). Fix all random bits except \( r^{(1)}[i] \) and \( r^{(2)}[j] \) arbitrarily. Note that

\[
b = \bigoplus_{i,j=1}^{n} \left( r^{(1)}[i]r^{(2)}[j] \cdot \bigoplus_{k=1}^{n} T^{(1)}[i, k]T^{(2)}[k, j] \right).
\]

Regardless of how the values of sums for other \( i, j \) look, one of the values of \( r^{(1)}[i] \) and \( r^{(2)}[j] \) yields the result \( b = 1 \); this happens with probability 1/4.

**Theorem 3.12.** For the \( MP \) function a lower bound of \( \Omega(n^3) \) holds for the length of strong Las Vegas formulas.

**Proof.** We use the Nečiporuk method. First the partition of the inputs has to be defined. There are \( n \) blocks \( b_j \) with the bits \( T^{(2)}(i, j) \) for \( i = 1, \ldots, n \) plus one block for the remaining inputs. Then Alice receives all inputs except \( n \) bits in column \( j \) of the second matrix, i.e., \( T^{(2)}(:, j) \), which go to Bob. We show that \( MP \) has now one-way communication complexity \( \Omega(n^2) \). The Nečiporuk method then gives us a lower bound of \( \Omega(n^3) \) for the length of deterministic and strong Las Vegas formulas. Without loss of generality (w.l.o.g.) assume Bob has the bits \( T^{(2)}(i, 1) \).
We construct a set of assignments to the input variables of Alice. Let $U$ be a subspace of $\mathbb{Z}_2^n$ and $T_U$ be a matrix, with $T_U x = 0 \iff x \in U$. For every $U$ we choose $T_U$ as $T^{(1)}$ and $T^{(2)}(i, j) = 0$ for all $i$ and for $j \geq 2$. If there are $2^{\Omega(n^2)}$ pairwise different subspaces, then we get that many different inputs. But these inputs correspond to different rows in the communication matrix, since all $T^{(1)}$ have different kernels. Thus with Corollary 3.6 the Las Vegas one-way communication is $\Omega(n^2)$.

To see that there are $2^{\Omega(n^2)}$ pairwise different subspaces of $\mathbb{Z}_2^n$, we count the subspaces with dimension at most $n/2$. There are $2^n$ vectors. There are $\binom{2^n}{n/2}$ possibilities to choose a set of $n/2$ pairwise different vectors. Each such set generates a subspace of dimension at most $n/2$. Each such subspace is generated by at most $\binom{2^{n/2}}{n/2}$ sets of $n/2$ pairwise different vectors from the subspace. Hence this number is an upper bound on the number of times a subspace is counted, and there are at least

$$\frac{\binom{2^n}{n/2}}{\binom{2^{n/2}}{n/2}} \geq 2^{\Omega(n^2)}$$

pairwise different subspaces of $\mathbb{Z}_2^n$.

**Corollary 3.13.** There is a function that can be computed by a fair one-sided error formula of length $O(N)$, while every strong Las Vegas formula needs length $\Omega(N^{3/2})$ for this task; i.e., there is a size gap of $\Omega(N^{1/2})$ between Las Vegas and one-sided error formulas.

There is also a size gap of $\Omega(N^{1/2})$ between one-sided error formulas and (two-sided) bounded error probabilistic formulas.

**Proof.** The first statement is proved in the previous theorems. For the second statement we consider the following function with four matrices as input. The function is the parity of the $MP$ function on the first two matrices and the complement of $MP$ on the other two matrices.

A fair probabilistic formula can compute the function obviously with length $O(n^2)$ following the construction in Theorem 3.11. Assume we have a one-sided error formula, then fix the first two input matrices once in a way so that their product is the 0 matrix and then so that their product is something else. In this way one gets one-sided error formulas for both $MP$ and its complement. Then one can use both formulas on the same input and combine their results to get a Las Vegas formula, which leads to the desired lower bound with Theorem 3.12.

For the construction of a Las Vegas formula let $F$ be the one-sided error formula for $MP$ and $G$ be the one-sided error formula for $\neg MP$. Then $F$ and $\neg G$ are formulas for $MP$, so that $F$ never erroneously accepts and is correct with probability $1/2$, and $\neg G$ never erroneously rejects and is correct with probability $1/2$. Assuming the function value is 0, then $F$ rejects. With probability $1/2$, $\neg G$ also rejects; otherwise, we may give up. Assuming the function value is 1, then $\neg G$ accepts. With probability $1/2$, $F$ also accepts; otherwise, we may give up. The other way around, if both formulas accept or both reject we can safely use this result, and this result comes up with probability $1/2$; the only other possible result is that $F$ rejects and $\neg G$ accepts, in which case we have to give up.

The formula described in the proof of Theorem 3.11 has the interesting property that each input is read exactly once, while the random inputs are read often. $MP$ cannot be computed by a deterministic formula reading the inputs only once, since this contradicts the size bound of Theorem 3.12. Later we will show that $MP$ cannot be computed substantially more efficiently by a fair probabilistic formula reading its
random inputs only once than by deterministic formulas. This follows from a lower bound for the size of such formulas given by the Nečiporuk function divided by $\log n$ (Corollary 6.7). Hence for the $MP$ function read-once random inputs are of little use.

4. Background on quantum computing and information. In this section we define more technical notions and describe results we will need. We start with information theory, then define the model of quantum formulas, and give results from quantum information theory. We also discuss programmable quantum gates. These results are used in the following section to give lower bounds for one-way communication complexity. Then we proceed to apply these to derive more formula size bounds.

4.1. Information theory. We now define a few notions from classical information theory; see, e.g., [10].

**Definition 4.1.** Let $X$ be a random variable with values in $S = \{x_1, \ldots, x_n\}$.

The entropy of $X$ is $H(X) = -\sum_{x \in S} Pr(X = x) \log Pr(X = x)$.

The entropy of $X$ given an event $E$ is $H(X | E) = -\sum_{x \in S} Pr(X = x | E) \log Pr(X = x | E)$.

The conditional entropy of $X$ given a random variable $Y$ is $H(X | Y) = \sum_y Pr(Y = y) H(X | Y = y)$, where the sum is over the values of $Y$. Note that $H(X | Y) = H(XY) - H(Y)$.

The information between $X$ and $Y$ is $I(X : Y) = H(X) - H(X | Y)$.

The conditional information between $X$ and $Y$, given $Z$, is $I(X : Y | Z) = H(XZ) - H(Z) - H(X | YZ)$.

For $\alpha \in [0, 1]$ we define $H(\alpha) = -\alpha \log \alpha - (1 - \alpha) \log (1 - \alpha)$.

All of the above definitions use the convention $0 \log 0 = 0$.

The following result is a simplified version of Fano's inequality; see [10].

**Fact 4.2.** If $X, Y$ are Boolean random variables with $Pr(X \neq Y) \leq \epsilon$, then $I(X : Y) \geq H(X) - H(\epsilon)$.

**Proof.** Let $Z = 1 \iff X = Y$ and $Z = 0 \iff X \neq Y$. Then $H(X | Y) = H(XY) - H(Y) = H(Y) - H(Y) \leq H(Z) \leq H(\epsilon)$. \qed

The next lemma is similar in the sense of a “Las Vegas variant.”

**Lemma 4.3.** Let $X$ be a random variable with a finite range of values $S$, and let $Y$ be a random variable with range $S \cup \{x_\gamma\}$, so that $Pr(Y = x | X = x) \geq 1 - \epsilon$ for all $x \in S$, $Pr(Y = x | X \neq x) = 0$ for all $x \neq x_\gamma$, and $Pr(Y = x_\gamma | X = x) \leq \epsilon$ for all $x \in S$. Then $I(X : Y) \geq (1 - \epsilon)H(X)$.

**Proof.** $I(X : Y) = H(X) - H(X | Y)$. Let $\delta = Pr(Y = x_\gamma) \leq \epsilon$, $\epsilon_x = Pr(Y = x_\gamma | X = x) \leq \epsilon$, $p_x = Pr(X = x)$.

\[
H(X | Y) \leq (1 - \delta)H(X | Y \neq x_\gamma) + \delta H(X | Y = x_\gamma)
\]

\[
= \delta H(X | Y = x_\gamma)
\]

\[
= -\delta \sum_x Pr(X = x | Y = x_\gamma) \log (Pr(X = x | Y = x_\gamma))
\]

\[
= -\delta \sum_x (\epsilon_x p_x / \delta) \log (\epsilon_x p_x / \delta)
\]

\[
\leq -\epsilon \sum_x p_x \log p_x + \delta \sum_x (\epsilon_x p_x / \delta) \log (\delta / \epsilon_x)
\]

\[
\leq \epsilon H(X) + \delta \log \sum_x p_x \text{ with Jensen’s inequality}
\]

\[
\leq \epsilon H(X). \quad \Box
\]
4.2. Quantum computation. We refer to [35] for a thorough introduction into the field. Let us briefly mention that pure quantum states are unit vectors in a Hilbert space written $|\psi\rangle$, inner products are denoted $\langle \psi | \phi \rangle$, and the standard norm is $|| \psi || = \sqrt{\langle \psi | \psi \rangle}$. Outer products $|\psi\rangle\langle \phi|$ are matrix valued.

In the space $\mathbb{C}^d$ we will consider not only the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ but also the Bell basis consisting of

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

The dynamics of a discrete time quantum system is described by unitary operations. We give some examples of such operations. A very useful operation is the Hadamard transform:

$$H^2 = \frac{1}{\sqrt{2}} \left( \begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array} \right).$$

Then $H_n = H_2 \otimes \cdots \otimes H_2$ is the $n$-wise tensor product of $H_2$.

The CNOT operation is defined by $CNOT : |x, y\rangle \rightarrow |x, x \oplus y\rangle$ on Boolean values $x, y$.

Furthermore measurements are fundamental operations. Measuring as well as tracing out subsystems leads to probabilistic mixtures of pure states.

**Definition 4.4.** An ensemble of pure states is a set $\{(p_i, |\phi_i\rangle)| 1 \leq i \leq k\}$. Here the $p_i$ are the probabilities of the pure states $|\phi_i\rangle$. Such an ensemble is called a mixed state.

The density matrix of a pure state $|\phi\rangle$ is the matrix $|\phi\rangle\langle \phi|$; the density matrix of a mixed state $\{(p_i, |\phi_i\rangle)| 1 \leq i \leq k\}$ is

$$\sum_{i=1}^{k} p_i |\phi_i\rangle\langle \phi_i|.$$ 

A density matrix is always Hermitian, positive semidefinite, and has trace 1. Thus a density matrix has nonnegative eigenvalues that sum to 1. The results of all measurements of a mixed state are determined by the density matrix.

A pure state in a Hilbert space $H = H_A \otimes H_B$ cannot in general be expressed as a tensor product of pure states in the subsystems.

**Definition 4.5.** A mixed state $\{(p_i, |\phi_i\rangle)| 1 \leq i \leq k\}$ in a Hilbert space $H_1 \otimes H_2$ is called separable if it has the same density matrix as a mixed state $\{(q_i, |\psi_i^1\rangle \otimes |\psi_i^2\rangle)| i = 1, \ldots, k'\}$ for pure states $|\psi_i^1\rangle$ from $H_1$ and $|\psi_i^2\rangle$ from $H_2$ with $\sum_i q_i = 1$ and $q_i \geq 0$. Otherwise, the state is called entangled.

Consider, e.g., the state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in $\mathbb{C}^2 \otimes \mathbb{C}^2$. The state is entangled and is usually called an EPR pair. This name refers to Einstein, Podolsky, and Rosen, who first considered such states [13].

Linear transformations on density matrices are called superoperators. Not all superoperators are physically allowed.

**Definition 4.6.** A superoperator $T$ is positive if it sends positive semidefinite Hermitian matrices to positive semidefinite Hermitian matrices. A superoperator is trace preserving if it maps matrices with trace 1 to matrices with trace 1.
A superoperator $T$ is completely positive if every superoperator $T \otimes I_F$ is positive, where $I_F$ is the identity superoperator on a finite dimensional extensional $F$ of the underlying Hilbert space.

A superoperator is physically allowed iff it is completely positive and trace preserving.

The following theorem characterizes physically allowed superoperators in terms of unitary operation, adding qubits, and tracing out [35].

**Fact 4.7.** The following statements are equivalent:
1. A superoperator $T$ sending density matrices over a Hilbert space $H_1$ to density matrices over a Hilbert space $H_2$ is physically allowed.
2. There is a Hilbert space $H_3$ with $\text{dim}(H_3) \leq \text{dim}(H_1)$ and a unitary map $U$, so that for all density matrices $\rho$ over $H_1$:

$$T\rho = \text{trace}_{H_1 \otimes H_3}[U(\rho \otimes |0_{H_3 \otimes H_2}\rangle \langle 0_{H_3 \otimes H_2}|)U^\dagger].$$

### 4.3. Quantum information theory

In this section we describe notions and results from quantum information theory.

**Definition 4.8.** The von Neumann entropy of a density matrix $\rho_X$ is $S(X) = S(\rho_X) = -\text{trace}(\rho_X \log \rho_X)$.

The conditional von Neumann entropy $S(X|Y)$ of a bipartite system with density matrix $\rho_{XY}$ is defined as $S(X|Y) = S(Y) - S(YX)$, where the state $\rho_Y$ of the $Y$ system is the result of a partial trace over $X$.

The von Neumann information between two parts of a bipartite system in a state $\rho_{XY}$ is $S(X : Y) = S(X) + S(Y) − S(XY)$ ($\rho_X$ and $\rho_Y$ are the results of partial traces).

The conditional von Neumann information of a system in state $\rho_{XYZ}$ is $S(X : Y|Z) = S(XZ) + S(YZ) - S(Z) - S(XYZ)$.

Let $\mathcal{E} = \{(p_i, \rho_i)\} | i = 1, \ldots, k$ be an ensemble of density matrices. The Holevo information of the ensemble is $\chi(\mathcal{E}) = S(\sum_i p_i \rho_i) - \sum_i p_i S(\rho_i)$.

The von Neumann entropy of a density matrix depends on the eigenvalues only, so it is invariant under unitary transformations. If the underlying Hilbert space has dimension $d$, then the von Neumann entropy of a density matrix is bounded by $\log d$.

A fundamental result is the so-called Holevo bound [16], which states an upper bound on the amount of classical information in a quantum state.

**Fact 4.9.** Let $X$ be a classical random variable with $\Pr(X = x) = p_x$. Assume for each $x$ that a quantum state with density matrix $\rho_x$ is prepared; i.e., there is an ensemble $\mathcal{E} = \{(p_x, \rho_x)| x = 0, \ldots, k\}$. Let $\rho_{XZ} = \sum_{x=0}^k p_x |x\rangle \langle x| \otimes \rho_x$. Let $Y$ be a classical random variable which indicates the result of a measurement on the quantum state with density matrix $\rho_Z = \sum_x p_x \rho_x$. Then

$$I(X : Y) \leq \chi(\mathcal{E}) = S(X : Z).$$

We will also need the following lemma.

**Lemma 4.10.** Let $\mathcal{E} = \{(p_x, \sigma_x)| x = 0, \ldots, k\}$ be an ensemble of density matrices, and let $\sigma = \sum_x p_x \sigma_x$ be the density matrix of the mixed state of the ensemble. Assume that there is an observable with possible measurement results $x$ and "?", so that for all $x$ measuring the observable on $\sigma_x$ yields $x$ with probability at least $1 - \epsilon$, the result "?" with probability at most $\epsilon$, and a result $x' \neq x$ with probability $0$; then

$$S(\sigma) \geq \sum_x p_x S(\sigma_x) + (1 - \epsilon)H(X), \quad \text{i.e., } \chi(\mathcal{E}) \geq (1 - \epsilon)H(X).$$
Proof. The proof proceeds similar to the information theoretic arguments in [2]. States $x$ of a classical random variable $X$ are coded as quantum states $\sigma_x$, where $x$ and $\sigma_x$ have probability $p_x$. The density matrix of the overall mixed state is $\sigma$ and has von Neumann entropy $S(\sigma)$. $\sigma$ corresponds to the “code” of a random $x$.

According to Holevo’s theorem (Fact 4.9) the information on $X$ one can access by measuring $\sigma$ with result $Y$ is bounded by $I(X : Y) \leq S(\sigma) - \sum_x p_x S(\sigma_x)$. But there is such a measurement as assumed in the lemma, and with Lemma 4.3 $I(X : Y) \geq (1 - \epsilon)H(X)$. Thus the lemma follows.

Not all of the relations that are valid in classical information theory hold in quantum information theory. The following fact states a notable exception, the so-called Araki–Lieb inequality and one of its consequences; see [35].

Fact 4.11. $S(XY) \geq |S(X) - S(Y)|$.

The reason for this behavior is entanglement.

Lemma 4.12. If $\sigma_{X|Y}$ is separable, then $S(XY) \geq S(X)$ and $S(X : Y) \leq S(X)$.

4.4. The quantum communication model. Now we define quantum one-way protocols.

Definition 4.13. In a two-player quantum one-way protocol players Alice and Bob each possess a private set of qubits. Some of the qubits are initialized to the Boolean inputs of the players; all other qubits are in some fixed basis state $|0\rangle$.

Alice then performs some quantum operation on her qubits and sends a set of these qubits to Bob. The latter action changes the possession of qubits rather than the global state. We can assume that Alice sends the same number of qubits for all inputs.

After Bob has received the qubits he can perform any quantum operation on the qubits in his possession, and afterwards he announces the result of the computation. The complexity of a protocol is the number of qubits sent.

In an exact quantum protocol the result has to be correct with certainty. $Q_E(f)$ is the minimal complexity of an exact quantum protocol for a function $f$.

In a bounded error protocol the output has to be correct with probability $1 - \epsilon$ (for $\epsilon > 0$). The bounded error quantum one-way communication complexity of a function $f$ is $Q_\epsilon(f)$, the minimal complexity of a bounded error quantum one-way protocol for $f$, and we set $Q(f) = Q_{1/3}(f)$.

Quantum Las Vegas protocols are defined in a manner similar to their probabilistic counterparts; the complexity measure notation is $Q_{0,\epsilon}(f)$.

Cleve and Buhrman [9] consider a different model of quantum communication: Before the start of the protocol Alice and Bob own a set of qubits whose state may be entangled but must be independent of the inputs. Then as above a quantum communication protocol is used. We use the superscript $\text{pub}$ to denote the complexity in this model.

It is possible to simulate the model with entangled qubits by allowing first an arbitrary finite communication independent of the inputs, followed by an ordinary protocol. By measuring distributed EPR pairs it is possible to simulate classical public randomness. The technique of superdense coding of [5] allows one in the model with prior entanglement to send $n$ bits of classical information with $\lceil n/2 \rceil$ qubits.

4.5. Quantum circuits and formulas. Besides quantum Turing machines quantum circuits [11] are a universal model of quantum computation (see [41]) and are generally easier to handle in descriptions of quantum algorithms. A more general model of quantum circuits in which superoperator gates work on density matrices is described in [1]. We begin with the basic model.
**Definition 4.14.** A unitary quantum gate with \( k \) inputs and \( k \) outputs is specified by a unitary operator \( U : \mathbb{C}^{2^k} \rightarrow \mathbb{C}^{2^k} \).

A quantum circuit consists of unitary quantum gates with \( O(1) \) inputs and outputs each, plus a set of inputs to the circuits, which are connected to an acyclic directed graph, in which the inputs are sources. Sources are labeled by Boolean constants or by input variables. Edges correspond to qubits; the circuit uses as many qubits as it has sources. One designated qubit is the output qubit. A quantum circuit computes a unitary transformation on the source qubits in the obvious way. In the end the output qubit is measured in the standard basis.

The size of a quantum circuit is the number of its gates; the depth is the length of the longest path from an input to the output.

A quantum circuit computes a function with a bounded error if it gives the right output with probability at least \( \frac{2}{3} \) for all inputs.

A quantum circuit computes a Boolean function with a one-sided error if it has a bounded error and furthermore never erroneously accepts.

A pair of quantum circuits computes a Boolean function \( f \) in the Las Vegas sense, if the first is a one-sided error circuit for \( f \) and the second is a one-sided error circuit for \( \neg f \).

A quantum circuit computes a function exactly if it makes no error.

The definition of Las Vegas circuits is motivated by the fact that we can easily verify the computation of a pair of one-sided error circuits for \( f \) and \( \neg f \) as in the classical case; see the proof of Corollary 3.13.

We are interested in restricted types of circuits, namely, quantum formulas [41].

**Definition 4.15.** A quantum formula is a quantum circuit with the following additional property: For each source there is at most one path connecting it to the output. The length or size of a quantum formula is the number of its sources.

Apart from the Boolean input variables a quantum formula is allowed to read Boolean constants only. There is only one final measurement. We also call the model from [41] pure quantum formulas. Compare also the definitions in [37].

In [1] a more general model of quantum circuits is studied, in which superoperators work on density matrices.

**Definition 4.16.** A superoperator gate \( g \) of order \((k,l)\) is a trace-preserving, completely positive map from the density matrices on \( k \) qubits to the density matrices on \( l \) qubits.

A quantum superoperator circuit is a directed acyclic graph with inner vertices marked by superoperator gates with fitting fan-in and fan-out. The sources are marked with input variables or Boolean constants. One gate is designated as the output.

A function is computed as follows. In the beginning the sources are each assigned a density matrix corresponding to the Boolean values determined by the input or by a constant. The Boolean value 0 corresponds to \(|0\rangle\langle0|\), 1 to \(|1\rangle\langle1|\). The overall state of the qubits involved is the tensor product of these density matrices.

Then the gates are applied in an arbitrary topological order. Applying a gate means applying the superoperator composed of the gate’s superoperator on the chosen qubits for the gate and the identity superoperator on the remaining qubits.

In the end the state of the output qubit is supposed to be a classical probability distribution on \(|0\rangle\) and \(|1\rangle\).

The following fact from [1] allows one to apply gates in an arbitrary topological ordering.

**Fact 4.17.** Let \( C \) be a quantum superoperator circuit, and let \( C_1 \) and \( C_2 \) be two sets of gates working on different sets of qubits. Then for all density matrices \( \rho \) on
the qubits in the circuit the result of $C_1$ applied to the result of $C_2$ on $\rho$ is the same as the result of $C_2$ applied to the result of $C_1$ on $\rho$.

Let two arbitrary topological orderings of the gates in a quantum superoperator circuit be given. The result of applying the gates in one ordering is the same as the result of applying the gates in the other ordering for any input density matrix.

One more aspect is interesting in the definition of quantum formulas: We want to allow quantum formulas to access multiple read random inputs, just as fair probabilistic formulas. Thus it is possible to simulate the latter model. Instead of random variables we allow the quantum formulas to read an arbitrary nonentangled state. A pure state on $k$ qubits is called nonentangled if it is the tensor product of $k$ states on one qubit each. A mixed state is nonentangled if it can be expressed as a probabilistic ensemble of nonentangled pure states. Note that a classical random variable read $k$ times can be modeled as $|1^k\rangle$ with probability 1/2 and $|0^k\rangle$ with probability 1/2.

We restrict our definition to gates with fan-in 2; the set of quantum gates with fan-in 2 is known to be universal [4].

**Definition 4.18.** A generalized quantum formula is a quantum superoperator circuit with fan-out 1/fan-in 2 gates together with a fixed nonentangled mixed state. The sources of the circuit are either labeled by input variables or may access a qubit of the state. Each qubit of this state may be accessed only by one gate.

As proved in [1] Fact 4.7 implies that quantum superoperator circuits with constant fan-in are asymptotically as efficient as quantum circuits with constant fan-in. The same holds for quantum formulas. The essential difference between pure and generalized quantum formulas is the availability of multiple read random bits.

### 4.6. Programmable quantum gates.

For simulations of quantum mechanical formulas by communication protocols we will need a programmable quantum gate. Such a gate allows Alice to communicate a unitary operation as a program stored in some qubits to Bob, who then applies this operation to some of his qubits.

Formally we have to look for a unitary operator $G$, with

$$G(|d\rangle \otimes |P_U\rangle) = U(|d\rangle) \otimes |P'_U\rangle.$$  

Here $|P_U\rangle$ is the “code” of a unitary operator $U$ and $|P'_U\rangle$ some leftover of the code.

The bad news is that such a programmable gate does not exist, as proved in [34]. Note that in the classical case such gates are easy to construct.

**Fact 4.19.** If $N$ different unitary operators (pairwise different by more than a global phase) can be implemented by a programmable quantum gate, then the gate needs a program of length $\log N$.

Since there are infinitely many unitary operators on just one qubit there is no programmable qubit with finite program length implementing them all. The proof uses the fact that the gate works deterministically, and actually a probabilistic solution to the problem exists.

We now sketch a construction of Nielsen and Chuang [34]. For the sake of simplicity we describe just the construction for unitary operations on one qubit.

The program of a unitary operator $U$ is

$$|P_U\rangle = \frac{1}{\sqrt{2}}(|0\rangle U|0\rangle + |1\rangle U|1\rangle).$$

The gate receives as input $|d\rangle \otimes |P_U\rangle$. The gate then measures the first and second qubits in the basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$. Then the third qubit is used as a result.
For a state $|d\rangle = a|0\rangle + b|1\rangle$ the input to the gate is

$$\frac{|a|0\rangle U|0\rangle + |1\rangle U|1\rangle}{\sqrt{2}} = \frac{1}{2} \left( |\Phi^+\rangle (aU|0\rangle + bU|1\rangle) + |\Phi^-\rangle (aU|0\rangle - bU|1\rangle) + |\Psi^+\rangle (aU|1\rangle + bU|0\rangle) + |\Psi^-\rangle (aU|1\rangle - bU|0\rangle) \right).$$

Thus the measurement produces the correct state with probability $1/4$, and moreover the result of the measurement indicates whether the computation was done correctly. Also, given this measurement result we know exactly which unitary “error” operation has been applied before the desired operation. We now state Nielsen and Chuang’s result.

**Fact 4.20.** There is a probabilistic programmable quantum gate with $m$ input qubits for the state plus $2m$ input qubits for the program, which implements every unitary operation on $m$ qubits and succeeds with probability $1/2^m$. The result of a measurement done by the gate indicates whether the computation was done correctly and which unitary error operation has been performed.

Also note that it is easy to construct an approximate programmable quantum gate in the following sense. For any error parameter $\epsilon$ we may discretize the set of superoperators to a finite set so that for each superoperator $T$ there is an operator $T'$ from the finite set, such that for each density matrix $\rho$ we have that $T\rho$ is $\epsilon$-close to $T'\rho$. Then we can construct a gate that receives the classical description of one of these finitely many superoperators as a program.

5. One-way communication complexity: The nondeterministic and the quantum cases.

5.1. A lower bound for limited nondeterminism. In this section we investigate nondeterministic one-way communication with a limited number of nondeterministic bits. Analogous problems for many-round communication complexity have been addressed in [18], but in this section we again consider asymmetric problems, for which the one-way restriction is essential.

It is easy to see that if player Bob has $m$ input bits, then $m$ nondeterministic bits are the maximum player Alice needs. Since the nondeterministic communication complexity without any limitation on the number of available nondeterministic bits is at most $m$, Alice can just guess the communication and send it to Bob in case it is correct with respect to her input and leads to acceptance. Bob can then check the same for his input. Thus an optimal protocol can be simulated.

For the application to lower bounds on formula size we are again interested in functions with an asymmetric input partition; i.e., Alice receives much more inputs than Bob. For nontrivial results thus the number of nondeterministic bits must be smaller than the number of Bob’s inputs.

A second observation is that using $s$ nondeterministic bits can reduce the communication complexity from the deterministic one-way communication complexity $d$ to $d/2^s$ in the best case. If $s$ is sublogarithmic, strong lower bounds follow already from the deterministic lower bounds, e.g., $N_{c \log n}(\neg EQ) \geq n^{1-\epsilon}$, while $N_{\log n}(\neg EQ) = O(\log n)$. On the other hand, we have the following.

**Lemma 5.1.**

$$N_s(f) = c \Rightarrow N_c(f) \leq c.$$

**Proof.** In a protocol with communication $c$ at most $2^c$ different messages can be sent (for all inputs). To guess such a message $c$ nondeterministic bits are sufficient. □
Hence it is unnecessary for a nondeterministic protocol to use more nondeterministic bits than communication. We are interested in determining how large the difference between nondeterministic one-way communication complexity with \( s \) nondeterministic bits and unrestricted nondeterministic communication complexity may be. Therefore we consider the maximal such gap as a function \( G \).

**Corollary 5.2.** Let \( f : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function and \( G : \mathbb{N} \rightarrow \mathbb{N} \) a monotone increasing function, with \( N(f) = c \) and \( N_s(f) = G(c) \) for some \( s \).

Then \( N_{G^{-1}(n)}(f) \leq c \) and hence \( s \leq G^{-1}(n) \), where \( G^{-1}(x) = \min \{ y \mid G(y) \geq x \} \).

Proof. \( G(c) \leq n \) and hence \( c \leq G^{-1}(n) \). \( \Box \)

The range of values of \( s \) for which a gap \( G \) between \( N(f) \) and \( N_s(f) \) is possible is thus limited. If, e.g., an exponential difference \( G(x) = 2^x \) holds, then \( s \leq \log n \). If \( G(x) = r \cdot x \), then \( s \leq n/r \).

We now show a gap between nondeterministic one-way communication complexity with \( s \) nondeterministic bits and unlimited nondeterministic communication complexity. First we define the family of functions exhibiting this gap. Denote by \( \mathcal{P}(a,b) \) the set of size \( b \) subsets of a size \( a \) universe.

**Definition 5.3.** Let \( DI_{n,s} \) be the following Boolean function for \( 1 \leq s \leq n \):

\[
DI_{n,s}(x_1, \ldots, x_{n+1}) = 1 \iff \forall i : x_i \in \mathcal{P}(n^3, s) \\
\land \exists j : |\{ j \mid j \neq i ; x_i \cap x_j \neq \emptyset \}| \geq s.
\]

Note that the function has \( \Theta(sn \log n) \) input bits in a standard encoding. We consider the partition of inputs in which Bob receives the set \( x_{n+1} \) and Alice all other sets. The upper bounds in the following lemma are trivial, since Bob receives only \( O(s \log n) \) input bits.

**Lemma 5.4.**

\[
N_{O(s \log n)}(DI_{n,s}) = O(s \log n), \\
DI^{UB}(DI_{n,s}) = O(s \log n).
\]

The lower bound we present now results in a near optimal difference between nondeterministic (one-way) communication and limited nondeterministic one-way communication. Limited nondeterministic one-way communication has also been studied subsequently to this work in [17]. There a tradeoff between the consumption of nondeterministic bits and the one-way communication is demonstrated (i.e., with more nondeterminism the communication gradually decreases). Here we describe a fundamentally different phenomenon of a threshold type: Nondeterministic bits do not help much, until a certain number of them are available, when quite quickly the optimal complexity is attained. For more results of this type see [22].

**Theorem 5.5.** There is a constant \( \epsilon > 0 \), so that for \( s \leq n \)

\[
N_{\epsilon s}(DI_{n,s}) = \Omega(ns \log n).
\]

Proof. We have to show that all nondeterministic one-way protocols computing \( DI_{n,s} \) with \( \epsilon s \) nondeterministic bits need much communication.

A nondeterministic one-way protocol with \( \epsilon s \) nondeterministic bits and communication \( c \) induces a cover of the communication matrix with \( 2^{\epsilon s} \) Boolean matrices having the following properties: Each 1-entry of the communication matrix is a 1-entry in at least one of the Boolean matrices, no 0-entry of the communication matrix
is a 1-entry in any of the Boolean matrices, and furthermore the set of rows appearing in those matrices has size at most $2^c$. This set of matrices is obtained by fixing the nondeterministic bits and taking the communication matrices of the resulting deterministic protocols. Note that the rows of the communication matrices of the deterministic protocols correspond to messages. We will deduce the lower bound from the weaker property that each of the Boolean matrices covering the communication matrix uses at most $2^c$ different rows. This can be used to show that the lower bound holds even for protocols with limited, but public, nondeterminism.

We start by constructing a submatrix of the communication matrix with some useful properties and then show the theorem for this “easier” problem.

Partition the universe $\{1, \ldots, n^3\}$ in $n$ disjoint sets $U_1, \ldots, U_n$, with $|U_i| = n^2 = m$. Then choose vectors of size $s$ subsets of the universe, so that the $i$th subset is from $U_i$. Thus the $n$ subsets of a vector are pairwise disjoint. Now the protocol has to determine whether the set of Bob intersects nontrivially with $s$ of the sets given to Alice.

We restrict the set of inputs further. There are $\binom{m}{s}$ different size $s$ subsets of $U_i$. We choose a set of such subsets so that each pair of them has no more than $s/2$ common elements. To do so we start with any subset and remove all subsets having more than $s/2$ common elements with any already chosen subset. This process can be repeated until all size $s$ subsets are either chosen or discarded. We end with a set of size $s$ subsets of $U_i$, whose elements have pairwise no more than $s/2$ common elements. In every step at most $\binom{m}{s} \binom{s}{s/2} \binom{m}{s/2} \geq \frac{\binom{m}{s}}{2^{3s/2}}$ sets.

As described we draw Alice’s inputs as vectors of sets, where the set at position $i$ is drawn from the set of subsets of $U_i$ we have just constructed. These inputs are identified with the rows of the submatrix of the communication matrix. The columns of the submatrix are restricted to elements of $U_1 \cup \{\top\} \times \cdots \times U_n \cup \{\top\}$, for which $s$ positions are occupied; i.e., $n - s$ positions carry the extra symbol $\top$ which stands for “no element.” Call the constructed submatrix $M$.

Now assume there is a protocol computing the restricted problem with communication matrix $M$. Fixing the nondeterministic bits gives us a deterministic protocol. If a nondeterministic protocol uses $r$ nondeterministic bits, then there are $2^r$ such deterministic protocols, and at least one of them accepts a fraction of $1/2^r$ of the ones in $M$, where $r = \epsilon s$. We now show that such a matrix must have many different rows, which corresponds to a large amount of communication.

Each row of $M$ (being a vector of zeros and ones) also corresponds to a vector of $n$ sets, the associated input of Alice. A position $i$ is called a difference position for a pair of such sequences if they have different sets at position $i$. According to our construction these sets have no more than $s/2$ elements in common.

We say a set of rows has $k$ difference positions if there are $k$ positions $i_1, \ldots, i_k$, so that for each $i_j$ there are two rows in the set for which $i_j$ is a difference position.

We now show that each row of $M'$ containing “many” ones does not “fit” on many rows of $M$, i.e., contains ones these do not have. Since $M'$ has a one-sided error, the rows of $M'$ are either sparse or cover only a few rows of $M$. Observe that each row of $M$ has exactly $\binom{m}{s} s^s$ ones.
Lemma 5.6. Let $z$ be a row of $M'$, appearing several times in $M'$. The rows of $M$, in whose place in $M$ the row $z$ appears in $M'$, may have $\delta n$ difference positions. Then $z$ contains at most $2\binom{n}{s}s^s/2^{\delta s/6}$ ones.

Proof. Several rows of $M$ having $\delta n$ difference positions are given, and the ones of $z$ occur in all of these rows. Let $C$ be the set of $\binom{n}{s}s^s$ columns/sets being the ones in the first such row. All other columns are forbidden and may not be ones in $z$.

A column in $C$ is chosen randomly by choosing $s$ out of $n$ positions and then one of $s$ elements for each position. Let $k = \delta s$. We have to show an upper bound on the number of ones in $z$, and we analyze this number as the probability of getting a one when choosing a column in $C$. The probability of getting a one is at most the probability that the chosen positions have a nontrivial intersection with less than $k/2$ sets $U_i$ at difference positions $i$ (call this event $E$) plus the probability of getting a one under the condition of event $\overline{E}$, following the general formula $\text{Prob}(A) \leq \text{Prob}(A|E) + \text{Prob}(\overline{E})$.

We first count the columns in $C$, which have a nontrivial intersection with at most $k/2$ of the sets $U_i$ at difference positions $i$. Consider the slightly different experiment in which $s$ times independently one of $n$ positions is chosen; hence positions may be chosen more than one time. Now expected $\delta s = k$ difference positions are chosen. Applying Chernov’s inequality yields that, with probability at most

$$e^{-\frac{1}{2} - k} \leq 2^{-\delta s/6},$$

at most $k/2$ difference positions occur. When choosing a random column in $C$ instead, this probability is even smaller, since now positions are chosen without repetitions. Thus the columns in $C$, which “hit” less than $k/2$ difference positions, contribute at most $2^{\delta s/6}\binom{n}{s}s^s$ ones to $z$.

Now consider the columns/sets in $C$, which intersect at least $k/2$ of the $U_i$ at difference positions $i$. Such a column/set fits on all of the rows, if the element at each position not bearing a $\top$ lies in the intersection of all sets in the rows at position $i$. At each difference position there are two rows, which hold different sets at that position, and those sets have no more than $s/2$ common elements.

Fix an arbitrary set of positions such that at least $k/2$ difference positions are included. The next step of choosing a column in $C$ consists of choosing one of $s$ elements for each position. But if a position is a difference position, then at most $s/2$ elements satisfy the condition of lying in the sets held by all of the rows at that position. Thus the probability of fitting on all of the rows is at most $2^{-k/2}$, and at most $2^{\delta s/6}\binom{n}{s}s^s$ such columns can be a one in $z$.

Overall only a fraction of $2^{-\delta s/6 + 1}$ of all columns in $C$ can be ones in $z$. \(\square\)

At least one-half of all ones in $M'$ lie in rows containing at least $\binom{m}{s}\delta n$ $s^s$ ones. Lemma 5.6 tells us that such a row fits only on a set of rows of $M$ having no more than $\delta n$ difference positions, where $r + 1 = \delta s/6 - 1$. Hence such a row can cover at most all of the ones in $\binom{m}{s}\delta n\binom{n}{s}s^s$ rows of $M$ and therefore only $\binom{m}{s}\delta n\binom{n}{s}s^s$ ones.

According to (5.1) at least $\frac{(m/s)^{m/2}\binom{n}{s}s^s}{\delta n \binom{n}{s}s^s 2^{3sn/2}2^{r+1}}$ ones are covered by such rows; hence

$$\frac{(m/s)^{m/2}\binom{n}{s}s^s}{\delta n \binom{n}{s}s^s 2^{3sn/2}2^{r+1}} \geq \frac{(m/s)^{m/2}}{(en/s)(3n/2)2^{3nsn/2}2^{r+1}} = 2^{\Omega(sn \log n)}$$

rows are necessary (for $\epsilon = 1/20$ and $n \geq s \geq 400$). \(\square\)
5.2. Quantum one-way communication. Our first goal in this section is to prove that the VC-dimension lower bound for randomized one-way protocols (Fact 2.9) can be extended to the quantum case. To achieve this we first prove a linear lower bound on the bounded error quantum communication complexity of the index function $IX_n$ and then describe a reduction from the index function $IX_d$ to any function with VC dimension $d$, thus transferring the lower bound. It is easy to see that $VC(IX_n) = n$, and thus the bounded error probabilistic one-way communication complexity is large for that function.

The problem of random access quantum coding has been considered in [2]. In a $n, m, \epsilon$-random access quantum code all Boolean $n$-bit words $x$ have to be mapped to states of $m$ qubits each, so that for $i = 1, \ldots, n$ there is an observable, so that measuring the quantum code with that observable yields the bit $x_i$ with probability $1 - \epsilon$. The quantum code is allowed to be a mixed state. The following is a result from [2].

**Fact 5.7.** For every $n, m, \epsilon$-random access quantum coding $m \geq (1 - H(\epsilon))n$.

It is easy to see that the problem of random access quantum coding is equivalent to the construction of a quantum one-way protocol for the index function. If there is such a protocol, then the messages can serve as mixed state codes, and if there is such a code, the codewords can be used as messages. We can thus deduce a lower bound for $IX_n$ in the model of one-way quantum communication complexity without prior entanglement.

We now give a proof that can also be adapted to the case of allowed prior entanglement. The proof follows Nayak’s idea, who also obtained the generalization to the case of prior entanglement in his thesis [31].

**Theorem 5.8.** $Q_\epsilon(IX_n) \geq (1 - H(\epsilon))n$.

**Proof.** Let $M$ be the register containing the message sent by Alice, and let $X$ be a register holding a uniformly random input to Alice. Then $\sigma_{X,M}$ denotes the state of Alice’s qubits directly before the message is sent. $\sigma_M$ is the state of a random message. Now every bit is decodable with probability $1 - \epsilon$, and thus $S(X_i : M) \geq 1 - H(\epsilon)$ for all $i$. To see this consider $S(X_i : M)$ as the Holevo information of the following ensemble:

$\sigma_{i,0} = \sum_{x, x_i = 0} \frac{1}{2^{n-1}} \sigma_M^{x}$

with probability $1/2$ and

$\sigma_{i,1} = \sum_{x, x_i = 1} \frac{1}{2^{n-1}} \sigma_M^{x}$

with probability $1/2$, where $\sigma_M^{x}$ is the density matrix of the message on input $x$. The information obtainable on $x_i$ by measuring $\sigma_M$ must be at $1 - H(\epsilon)$ due to Fano’s inequality (Fact 4.2), and thus the Holevo information of the ensemble is at least $1 - H(\epsilon)$; hence $S(X_i : M) \geq 1 - H(\epsilon)$.

But then $S(X : M) \geq (1 - H(\epsilon))n$ (since all $X_i$ are mutually independent).

$S(X : M) \leq S(M)$ using Lemma 4.12, since $X$ and $M$ are not entangled. Thus the number of qubits in $M$ is at least $(1 - H(\epsilon))n$.

Now we analyze the complexity of $IX_n$ in the one-way communication model with entanglement.
The density matrix of the state induced by a uniformly random input on $X$, the message $M$, and the qubits $E_A, E_B$ containing the prior entanglement in the possession of Alice and Bob is $\sigma_{X,M,E_A,E_B}$. Here $E_A$ contains those qubits of the entangled state Alice keeps; note that some of the entangled qubits will usually belong to $M$. Tracing out $X$ and $E_A$ we receive a state $\sigma_{M,E_B}$, which is accessible to Bob. Now every bit of the string in $X$ is decodable; thus $S(X_i : M|E_B) \geq 1 - H(\epsilon)$ for all $i$ as before. But then also $S(X : M|E_B) \geq (1 - H(\epsilon)n)$, since all of the $X_i$ are mutually independent.

$$S(X : M|E_B) = S(X : E_B) + S(X : M|E_B) \leq 2S(M)$$

by an application of the Araki–Lieb inequality; see Fact 4.11. Note that $S(X : E_B) = 0$. So the number of qubits in $M$ must be at least $1 - H(\epsilon)n/2$.

Note that the lower bound shows that two-round deterministic communication complexity can be exponentially smaller than one-way quantum communication complexity. For a more general quantum communication round hierarchy see [25].

**Theorem 5.9.** For all functions $f : Q_d(f) \geq (1 - H(\epsilon))\text{VC}(f)$ and $Q_e^{\text{pub}}(f) \geq (1 - H(\epsilon))\text{VC}(f)/2$.

**Proof.** We now describe a reduction from the index function to $f$. Assume $\text{VC}(f) = d$; i.e., there is a set $S = \{s_1, \ldots, s_d\}$ of inputs for Bob, which is shattered by the set of functions $f(x, \cdot)$. The reduction then goes from $IX_d$ to $f$.

For each $R \subseteq S$ let $c_R$ be the incidence vector of $R$ (having length $d$). $c_R$ is a possible input for Alice when computing the index function $IX_d$. For each $R$ choose some $x_R$, which separates this subset from the rest of $S$, i.e., so that $f(x_R, y) = 1$ for all $y \in R$ and $f(x_R, y) = 0$ for all $y \in S - R$.

Assume a protocol for $f$ is given. To compute the index function the players do the following. Alice maps $c_R$ to $x_R$. Bob’s inputs $i$ are mapped to the $s_i$. Then $f(x_R, s_i) = 1 \iff s_i \in R \iff c_R(i) = 1$.

In this manner a quantum protocol for $f$ must implicitly compute $IX_d$. According to Theorem 5.8 the lower bounds follow.

As an application of the previous theorem we get lower bounds for the disjointness problem in the model of quantum one-way communication complexity.

**Corollary 5.10.** $Q_e(\text{DISJ}_n) \geq (1 - H(\epsilon))n$.

$Q_e^{\text{pub}}(\text{DISJ}_n) \geq (1 - H(\epsilon))n/2$.

The first result has independently been obtained by Buhrman and de Wolf [8].

Theorem 5.8 and the following example show that the obtained lower bound method is not tight in general. There are functions for which an unbounded gap exists between the VC dimension and the quantum one-way communication complexity [24].

Now we turn to the exact and Las Vegas quantum one-way communication complexity. For classical one-way protocols it is known that Las Vegas communication complexity is at most a factor $1/2$ better than deterministic communication for total functions; see Fact 2.10.

**Theorem 5.11.** For all total functions $f$:

$Q_E(f) = D(f)$,

$Q_e^{\text{pub}}(f) \geq (1 - \epsilon)D(f)$.

**Proof.** Let $\text{row}(f)$ be the number of different rows in the communication matrix of $f(x, y)$. According to Fact 2.7, $D(f) = \lceil \log \text{row}(f) \rceil$. We assume in the following that the communication matrix consists of pairwise different rows only.

We will show that any Las Vegas one-way protocol which gives up with probability at most $\epsilon \geq 0$ for some function $f$ having $\text{row}(f) = R$ must use messages with von Neumann entropy at least $(1 - \epsilon)\log R$, when started on a uniformly random input. Inputs for Alice are identified with rows of the communication matrix. We then
conclude that the Hilbert space of the messages must have dimension at least $R^{1-\epsilon}$, and hence at least $(1-\epsilon) \log R$ qubits have to be sent. This gives us the second lower bound of the theorem. The upper bound of the first statement is trivial; the lower bound of the first statement follows by taking $\epsilon = 0$.

We now describe a process in which rows of the communication matrix are chosen randomly bit per bit. Let $p$ be the probability of having a 0 in column 1 (i.e., the number of 0s in column 1 divided by the number of rows). Then a 0 is chosen with probability $p$, a 1 with probability $1-p$. Afterwards the set of rows is partitioned into the set $I_0$ of rows starting with a 0 and the set $I_1$ of rows starting with a 1. When $x_1 = b$ is chosen, the process continues with $I_b$ and the next column.

Let $\rho_y$ be the density matrix of the following mixed state: The (possibly mixed) message corresponding to a row starting with $y$ is chosen uniformly over all such rows.

The probability that a 0 is chosen after $y$ is called $p_y$, and the number of different rows beginning with $y$ is called $\text{row}_y$.

We want to show via induction that $S(\rho_y) \geq (1-\epsilon) \log \text{row}_y$. Surely $S(\rho_y) \geq 0$ for all $y$.

Recall that Bob can determine the function value for an arbitrary column with the correctness guarantee of the protocol.

Then with Lemma 4.10, $S(\rho_y) \geq p_y S(\rho_{y0}) + (1-p_y) S(\rho_{y1}) + (1-\epsilon) H(p_y)$, and via induction

$$S(\rho_y) \geq p_y ((1-\epsilon) \log \text{row}_{y0})$$

$$+ (1-p_y)((1-\epsilon) \log \text{row}_{y1}) + (1-\epsilon) H(p_y)$$

$$= (1-\epsilon) p_y \log(p_y \text{row}_y)$$

$$+ (1-p_y) \log((1-p_y) \text{row}_y) + H(p_y)]$$

$$= (1-\epsilon) \log \text{row}_y.$$  

We conclude that $S(\rho) \geq (1-\epsilon) \log \text{row}(f)$ for the density matrix $\rho$ of a message to a uniformly random row. Hence the lower bound on the number of qubits holds. $\square$

We now again consider the model with prior entanglement.

**Theorem 5.12.** For all total functions $f$:

$$Q^\text{pub}_E(f) = \lceil D(f)/2 \rceil,$$

$$Q^\text{pub}_E(f) \geq D(f)(1-\epsilon)/2.$$

The upper bound follows from superdense coding [5]. Instead of the lower bounds of the theorem we prove a stronger statement. We consider an extended model of quantum one-way communication that will be useful later.

In a nonstandard one-way quantum protocol Alice and Bob are allowed to communicate in arbitrarily many rounds; i.e., they can exchange many messages. But Bob is not allowed to send Alice a message, so that the von Neumann information between Bob’s input and all of Alice’s qubits is larger than 0. The communication complexity of a protocol is the number of qubits sent by Alice in the worst case. The model is at least as powerful as the model with prior entanglement, since Bob may, e.g., generate some EPR pairs and send one qubit of each pair to Alice, and then Alice may send a message as in a protocol with prior entanglement.

**Lemma 5.13.** For all functions $f$ a nonstandard quantum one-way protocol with a bounded error must communicate at least $(1-H(\epsilon))VC(f)/2$ qubits from Alice to Bob.

For all total functions $f$ a nonstandard quantum one-way protocol

1. with exact acceptance must communicate at least $\lceil D(f)/2 \rceil$ qubits from Alice to Bob;
2. with Las Vegas acceptance and success probability $1 - \epsilon$ must communicate at least $(1 - \epsilon)D(f)/2$ qubits from Alice to Bob.

Proof. In this proof we always call the qubits available to Alice $P$ and the qubits available to Bob $Q$ for simplicity disregarding that these registers change during the course of the protocol. We assume that the inputs are in registers $X, Y$ and are never erased or changed in the protocol. Furthermore we assume that for all fixed values $x, y$ of the inputs the remaining global state is pure.

For the first statement it is again sufficient to investigate the complexity of the index function.

Let $\sigma_{XYPQ}$ be the state for random inputs in $X, Y$ for Alice and Bob, with qubits $P$ and $Q$ in the possession of Alice and Bob, respectively. Since Bob determines the result, it must be true that in the end of the protocol $S(X : YQ) \geq 1 - H(\epsilon)$, since the value $X_Y$ can be determined from Bob’s qubits with probability $1 - \epsilon$. It is always true in the protocol that $S(XP : Y) = 0$. Let $\rho_P^{X=x, Y=y}$ be the density matrix of $P$ for fixed inputs $X = x$ and $Y = y$. Then we have that for all $x, y, y'$:

$$\rho_P^{X=x, Y=y} = \rho_P^{X=x, Y=y'}.$$  

Thus the following fact from [29] and [28] tells us that all $y$ and corresponding states of $Q$ are “equivalent” from the perspective of Alice.

**FACT 5.14.** Assume $|\phi_1\rangle$ and $|\phi_2\rangle$ are pure states in a Hilbert space $H \otimes K$, so that $\operatorname{Tr}_K|\phi_1\rangle \langle \phi_1| = \operatorname{Tr}_K|\phi_2\rangle \langle \phi_2|$. Then there is a unitary transformation $U$ acting on $K$, so that $I \otimes U|\phi_1\rangle = |\phi_2\rangle$ (for the identity operator $I$ on $H$).

Thus there is a local unitary transformation applicable by Bob alone, so that $\rho_{yr}^{X=x, Y=y}$ can be changed to $\rho_{yr}^{X=x, Y=y'}$. Hence for all $i$ we have $S(QY : X_i) \geq 1 - H(\epsilon)$, and thus $S(X : QY) \geq (1 - H(\epsilon))n$.

In the beginning $S(X : QY) = 0$. Then the protocol proceeds w.l.o.g. so that each player applies a unitary transformation on his qubits and then sends a qubit to the other player. Since the information cannot increase by local operations, it is sufficient to analyze what happens if qubits are sent. When Bob sends a qubit to Alice, $S(X : QY)$ is not increased. When Alice sends a qubit to Bob, then $Q$ is augmented by a qubit $M$, and $S(X : QMY) \leq S(X : QY) + S(XQY : M) \leq S(X : QY) + 2S(M) \leq S(X : QY) + 2$ due to Fact 4.11. Thus the information can increase only when Alice sends a qubit and always by at most 2. The lower bound follows.

Now we turn to the second part. We consider the same situation as in the proof of Theorem 5.11. Let $\sigma_P^{rc}$ denote the density matrix of the qubits $P$ in Alice’s possession under the condition that the input row is $r$ and the input column is $c$. Clearly $\sigma_P^{rc}$ (containing also Bob’s qubits) is a purification of $\sigma_P^{rc'}$. Again $\sigma_P^{rc} = \sigma_P^{rc'}$ for all $r, c, c'$, and according to Fact 5.14 for all $c$ and all corresponding states of $Q$, it is true that Bob can switch locally between them. Hence it is possible for Bob to compute the function for an arbitrary column.

The probability of choosing a 0 after a prefix $y$ of a row is again called $p_y$, and the number of different rows beginning with $y$ is called $row_y$. $p_y$ contains the state of Bob’s qubits at the end of the protocol if a random row starting with $y$ is chosen uniformly (and some fixed column $c$ is chosen). Surely $S(p_y) \geq 0$ for all $y$. Since Bob can change his column (and the corresponding state of $Q$) by a local unitary transformation, he is able to compute the function for an arbitrary column, always with the success probability of the protocol, at the end. With Lemma 4.10 $S(p_y) \geq p_yS(p_y0) + (1 - p_y)S(p_y1) + (1 - \epsilon)H(p_y)$.
At the end of the protocol thus \( S(\sigma^n_Q) = S(\rho) \geq (1-\epsilon) \log \frac{\text{row}(f)}{s} + \sum_r \frac{1}{\text{out}(f)} S(\sigma^n_{Q_r}) \) for all \( c \). Thus the Holevo information of the ensemble, in which \( \rho_r = \sigma^n_{Q_r} \) is chosen with probability \( 1/\text{row}(f) \), is at least \((1-\epsilon) \log \text{row}(f)\). Let \( \sigma_{RQP} \) be the density matrix of rows, qubits of Alice and Bob. It follows that \( S(R:Q) \geq (1-\epsilon) \log \text{row}(f) \), and as before at least half that many qubits have to be sent from Alice to Bob. \( \square \)

6. More lower bounds on formula size.

6.1. Nondeterminism and formula size. Let us first mention that any nondeterministic circuit can easily be transformed into an equivalent nondeterministic formula without increasing size by more than a constant factor. To do so one simply guesses the values of all internal gates and then verifies that all of these guesses are correct and that the circuit accepts. This is a big AND over test involving \( O(1) \) variables, which can be implemented by a Boolean formula in conjunctive normal form. Hence large lower bounds for nondeterministic formula size are very hard to prove, since even nonlinear lower bounds for the size of deterministic circuits computing some explicit functions are unknown. We now show that formulas with limited nondeterminism are more approachable. We start by introducing a variant of the Nečiporuk method, this time in terms of nondeterministic communication:

**Definition 6.1.** Let \( f \) be a Boolean function with \( n \) input variables and \( y_1 \ldots y_k \) be a partition of the inputs in \( k \) blocks.

Player Bob receives the inputs in \( y_i \), and player Alice receives all other inputs. The nondeterministic one-way communication complexity of with \( s \) nondeterministic bits of \( f \) under this input partition is called \( N_s(f) \). Define the \( s \)-nondeterministic Nečiporuk function as \( \frac{1}{4} \sum_{i=1}^k N_s(f_i) \).

**Lemma 6.2.** The \( s \)-nondeterministic Nečiporuk function is a lower bound for the length of nondeterministic Boolean formulas with \( s \) nondeterministic bits.

The proof is analogous to the proof of Theorem 3.5. Again protocols simulate the formula in \( k \) communication games. This time Alice fixes the nondeterministic bits by herself, and no probability distribution on formulas is present.

We will apply the above methodology to the following language.

**Definition 6.3.** Let \( AD_{n,s} \) denote the following language (for \( 1 \leq s \leq n \)):

\[
AD_{n,s} = \{(x_1, \ldots, x_{n+1}) | \forall i : x_i \in P(n^3, s),
\]

\[
\quad x_i \text{ is written in sorted order}
\]

\[
\wedge \exists i : |\{j | j \neq i; x_i \cap x_j \neq \emptyset\}| \geq s).
\]

**Theorem 6.4.** Every nondeterministic formula with \( s \) nondeterministic bits for \( AD_{n,20s} \) has length at least \( \Omega(n^2 s \log n) \).

\( AD_{n,s} \) can be computed by a nondeterministic formula of length \( O(ns^2 \log n) \), which uses \( O(s \log n) \) nondeterministic bits (for \( s \geq \log n \)).

**Proof.** For the lower bound we use the methodology we have just described. We consider the \( n+1 \) partitions of the inputs, in which Bob receives the set \( x_i \) and Alice all other sets. The function they have to compute now is the function \( DI_{n,s} \) from Definition 5.3. In Theorem 5.5 a lower bound of \( \Omega(ns \log n) \) is shown for this problem; hence the length of the formula is \( \Omega(n \cdot ns \log n) \).

For the upper bound we proceed as follows: The formula guesses (in binary) a number \( i \) with \( 1 \leq i \leq n+1 \) and pairs \((j_1, w_1), \ldots, (j_s, w_s)\), where \( 1 \leq j_k \leq n+1 \) and \( 1 \leq w_k \leq n^3 \) for all \( k = 1, \ldots, s \). The number \( i \) indicates a set, and the pairs are witnesses that set \( i \) and set \( j_k \) intersect on element \( w_k \).
The formula does the following tests. First there is a test of whether all sets consist of $s$ sorted elements. For this $ns$ comparisons of the form $x^j_i < x^j_{i+1}$ suffice, which can be realized with $O(\log^2 n)$ gates each. Since $s \geq \log n$ overall $O(ns^2\log n)$ gates are enough.

The next test is whether $j_1 < \cdots < j_s$. This makes sure that witnesses for $s$ different sets have been guessed. Also $i \neq j_k$ for all $k$ must be tested.

Then the formula tests whether for all $1 \leq l \leq n$ the following holds: If $l = i$, then all guessed elements are in $x_l$; if $1 \leq l \leq n + 1$ and $1 \leq k \leq s$, the formula also tests, whether $l = j_k$ implies, that $w_k \in x_l$.

All of these tests can be done simultaneously by a formula of length $O(ns^2 \log n)$.

For $0 < \epsilon \leq 1/2$ let $s = n^{1-\epsilon}$; then the lower bound for limited nondeterministic formulas is $\Omega(N^{2-\epsilon}/\log^{1-\epsilon} N)$, with $N^{\epsilon}/\log^\epsilon N$ nondeterministic bits allowed. $O(N^{\epsilon}\log^{1-\epsilon} N)$ nondeterministic bits suffice to construct a formula having length $O(N^{1+\epsilon}/\log^\epsilon N)$. Hence the threshold for constructing an efficient formula is polynomially large, allowing an exponential number of computations on each input.

### 6.2. Quantum formulas

Now we derive the lower bound for generalized quantum formulas. Roychowdhury and Vatan [37] consider pure quantum formulas (recall these are quantum formulas which may not access multiply readable random bits). Their result is as follows.

**Fact 6.5.** Every pure quantum formula computing a function $f$ with a bounded error has length

$$\Omega \left( \sum_i D(f_i)/\log D(f_i) \right)$$

for the Nečiporuk function $\sum_i D(f_i)$; see Fact 3.1 and Definition 3.4.

Furthermore in [37] it is shown that pure quantum formulas can be simulated efficiently by deterministic circuits.

Now we know from section 3.2 that the Boolean function $MP$ with $O(n^2)$ inputs (the matrix product function) has fair probabilistic formulas of linear size $O(n^2)$, while the Nečiporuk bound is cubic (Theorems 3.11 and 3.12). Thus we get the following.

**Corollary 6.6.** There is a Boolean function $MP$ with $N$ inputs, which can be computed by fair one-sided error formulas of length $O(N)$, while every pure quantum formula with a bounded error for $MP$ has size $\Omega(N^{3/2}/\log N)$.

We conclude that pure quantum formulas are not a proper generalization of classical formulas. A fair probabilistic formula can be simulated efficiently by a generalized quantum formula, on the other hand. We now derive a lower bound method for generalized quantum formulas. First we give again a lower bound in terms of one-way communication complexity, and then we show that the VC–Nečiporuk bound is a lower bound, too.

This implies with Theorem 3.9 that the maximal difference between the sizes of deterministic formulas and generalized bounded error quantum formulas provable with the Nečiporuk method is at most $O(\sqrt{n})$.

But first let us conclude the following corollary, which states that fair probabilistic formulas reading their random bits only once are sometimes inefficient.

**Corollary 6.7.** The (standard) Nečiporuk function divided by $\log n$ is an asymptotical lower bound for the size for fair probabilistic formulas reading their random inputs only once.
Proof. We have to show that pure quantum formulas can simulate these special probabilistic formulas. For each random input we use two qubits in the state $|00\rangle$. These are transformed into the state $|\Phi^+\rangle$ by a Hadamard gate. One of the qubits is never used again; then the other qubit has the density matrix of a random bit. Then the probabilistic formula can be simulated. For the simulation of gates unitary transformations on three qubits are used. These get the usual inputs of the gate simulated plus one empty qubit as input, which after the application of the gate carries the output. These gates are easily constructed unitarily. According to [4] each 3-qubits gate can be composed of $O(1)$ unitary gates on two qubits only. 

We will need the following observation [1].

**Fact 6.8.** If the density matrix of two qubits in a circuit (with nonentangled inputs) is not the tensor product of their density matrices, then there is a gate so that both qubits are reachable on a path from that gate.

Since the above situation is impossible in a formula, the inputs to a gate are never entangled.

The first lower bound is stated in terms of one-way communication complexity. It is interesting that actually randomized complexity suffices for a lower bound on quantum formulas.

**Theorem 6.9.** Let $f$ be a Boolean function on $n$ inputs and $y_1 \ldots y_k$ a partition of the input variables in $k$ blocks. Player Bob knows the inputs in $y_i$, and player Alice knows all other inputs. The randomized (private coin) one-way communication complexity of $f$ (with a bounded error) under this input partition is called $R(f_i)$.

Every generalized quantum formula for $f$ with a bounded error has length

$$\Omega \left( \sum_i \frac{R(f_i)}{\log R(f_i)} \right).$$

Proof. For a given partition of the input we show how a generalized quantum formula $F$ can be simulated in the $k$ communication games, so that the randomized one-way communication in game $i$ is bounded by a function of the number of leaves in a subtree $F_i$ of $F$. $F_i$ contains exactly the variables belonging to Bob as leaves, and its root is the root of $F$. Furthermore $F_i$ contains all gates on paths from these leaves to the root. Note that the additional nonentangled mixed state which the formula may access is given to Alice.

$F$ is a tree of fan-in 2 fan-out 1 superoperators (recall that superoperators are not necessarily reversible). “Wires” between the gates carry one qubit each. $F_i$ is a formula that Bob wants to evaluate, the remaining parts of the formula $F$ belong to Alice, and she can easily compute the density matrices for all qubits on any wire in her part of the formula by a classical computation, as well as the density matrices for the qubits crossing to Bob's formula $F_i$. Note that none of the qubits on wires crossing to $F_i$ is entangled with another, so the state of these qubits is a probabilistic ensemble of pure nonentangled states. Hence Alice may fix a pure nonentangled state from this ensemble with a randomized choice.

In all communication games Bob evaluates the formula as far as possible without the help of Alice. By an argument as in other Nečiporuk methods (e.g., [7, 37] or the previous sections) it is sufficient to send a few bits from Alice to Bob to evaluate a path with the following property: All gates on the path have one input from Alice and one input from its predecessor, except for the first gate, which has one input from Alice and one (already known) input from Bob. With standard arguments the number of such paths is a lower bound on the number of leaves in the subformula; see section 3.1.
Hence we have to consider some path $g_1, \ldots, g_m$ in $F$, where $g_1$ has one input or a gate from Alice as predecessor and an input or gate from Bob as the other predecessor, and all gates $g_i$ have the previous gate $g_{i-1}$ and an input or gate from Alice’s part of the formula as predecessors. The density matrix of Bob’s input to $g_1$ is called $\rho$, and the density matrix of the other $m$ inputs is called $\sigma$. The circuit computing $\sigma$ works on different qubits than the circuit computing $\rho$.

Thus the density matrix of all inputs to the path is $\rho \otimes \sigma$; see Fact 6.8. The path maps $\rho \otimes \sigma$ with a superoperator $T$ to a density matrix $\mu$ on one qubit; alternatively we may view $\sigma$ as determining a superoperator $T_\sigma$ on one qubit that has to be applied to $\rho$. Now Alice can compute this superoperator by herself, classically.

Bob knows $\rho$. Bob wants to know the state $T_\sigma \rho$. Since this operator works on a single qubit only, it can be described within a precision $1/poly(k)$ by a constant size matrix containing numbers of size $O(\log k)$ for any integer $k$. Thus Alice may communicate $T_\sigma$ to Bob within this precision using $O(\log k)$ bits.

In this way Alice and Bob may evaluate the formula, and the error of the formula is changed only by $\text{size}_i/poly(k)$ compared to the error of the quantum formula, where $\text{size}_i$ denotes the number of gates in $F_i$. Thus choosing $k = poly(\text{size}_i)$ the communication is bounded $R(f_i) \leq O(\text{size}_i \log \text{size}_i)$. This implies $\text{size}_i \geq \Omega(R(f_i)/\log R(f_i))$. Summation over all $i$ yields the theorem. □

The above construction loses a logarithmic factor, but in the combinatorial bounds we actually apply, we can avoid this, by using quantum communication and the programmable quantum gate from Fact 4.20.

**Theorem 6.10.** The VC–Nečiporuk function is an asymptotical lower bound for the length of generalized quantum formulas with a bounded error.

The Nečiporuk function is an asymptotical lower bound for the length of generalized quantum Las Vegas formulas.

**Proof.** We proceed similar to the above construction, but Alice and Bob use quantum computers. Instead of communicating a superoperator in matrix form with some precision, we use the programmable quantum gate.

Alice and Bob cooperatively evaluate the formulas $F_i$ in a communication game as before. As before, for certain paths Alice wants to help Bob to apply a superoperator $T_\sigma$ on a state $\rho$ of his. Using Fact 4.7 we can assume that this is a unitary operator on $O(1)$ qubits (one of them $\rho$, the others blank) followed by throwing away all but one of the qubits.

This time Alice sends to Bob the program corresponding to the unitary operation in $T_\sigma$. Bob feeds this program into the programmable quantum gate, which tries to apply the transformation, and if this is successful, the formula evaluation can continue after discarding the unnecessary qubits. This happens with probability $\Omega(1)$. If Alice could get some notification from Bob saying whether the gate has operated successfully, and if not, what kind of error occurred, then Alice could send him another program that both undoes the error and the previous operator and then makes another attempt to compute the desired operator.

Note that the error that resulted by an application of the programmable quantum gate is determined by the classical measurement outcome resulting in its application. Furthermore this error can be described by a unitary transformation itself. If the error function is $E$, the desired is unitary is $U$, and the state it has to be applied to is $\rho$, then Bob now holds $UE\rho E^\dagger U^\dagger$. Once Alice knows $E$ (which is determined by Bob’s measurement outcome), Alice can produce a program for $UE^\dagger U^\dagger$. If Bob applies this transformation successfully, they are done; otherwise, they can iterate. Note that only an expected number of $O(1)$ such iterations
are necessary, and hence the expected quantum communication in this process is $O(1)$, too.

So the expected communication can be reduced to $O(size_i)$. But Alice needs some communication from Bob. Luckily this communication does not reveal any information about Bob’s input: Bob’s measurement outcomes are random numbers without correlation with his input.

So we consider the nonstandard one-way communication model from Lemma 5.13, in which Bob may talk to Alice but without revealing any information about his input. Using this model in the construction and letting Bob always ask explicitly for more programs reduces the communication in game $i$ to $O(size_i)$ in the expected sense.

With Lemma 5.13 we get the lower bounds for a bounded error and Las Vegas communication.

Now we can give a lower bound for ISA showing that even generalized quantum formulas compute the function not significantly more efficient than deterministic formulas.

**Corollary 6.11.** Every generalized quantum formula which computes ISA with a bounded error has length $\Omega(n^2/\log n)$.

Considering the matrix multiplication function $MP$, we get the following.

**Corollary 6.12.** There is a function, which can be computed by a generalized quantum formula with a bounded error as well as by a fair probabilistic formula with a bounded error, with size $O(N)$. Every generalized quantum Las Vegas formula needs size $\Omega(N^{3/2})$ for this task. Hence there is a size gap of $\Omega(N^{1/2})$ between the Las Vegas formula length and the length of the bounded error formulas.

Since the VC–Nečiporuk function is a lower bound for generalized quantum formulas, Theorem 3.9 implies that the maximal size gap between deterministic formulas and generalized quantum formulas with a bounded error provable by the (standard) Nečiporuk method is $O(\sqrt{n})$ for input length $n$. Such a gap actually already lies between generalized quantum Las Vegas formulas and fair probabilistic formulas with a bounded error.

7. **Conclusions.** In this paper we have derived lower bounds for the sizes of probabilistic, nondeterministic, and quantum formulas. These lower bounds follow the general approach of reinterpreting the Nečiporuk bound in terms of one-way communication complexity. This is nontrivial in the case of quantum formulas, where we had to use a programmable quantum gate. Nevertheless we have obtained the same combinatorial lower bound for quantum and probabilistic formulas based on the VC dimension.

Using the lower bound methods we have derived a general $\sqrt{n}$ gap between the bounded error and Las Vegas formula size. Another result is a threshold phenomenon for the amount of nondeterminism needed to compute a function, which gives a near-quadratic size gap for a polynomial threshold on the number of nondeterministic bits.

To derive our results we needed lower bounds for one-way communication complexity. These results give gaps between two-round and one-way communication complexity in these models. Those gaps have been generalized to round hierarchies for larger number of rounds in [22] and [25] for the nondeterministic and the quantum cases, respectively. Furthermore we have shown that quantum Las Vegas one-way protocols for total functions are not much more efficient than deterministic one-way protocols. The lower bounds for quantum one-way communication complexity are also useful to give lower bounds for quantum automata and for establishing that only bounded error quantum finite automata can be exponentially smaller than deterministic finite automata [23]. A generalization of the VC-dimension bound on quantum one-way communication complexity is given in [24].
The following problems remain open:

1. Give a better separation between deterministic and probabilistic/quantum formula size (see [21] for a candidate function).
2. Separate the size complexities of generalized quantum and probabilistic formulas for some function.
3. Investigate the power of quantum formulas that can access an entangled state as an additional input, thus introducing entanglement into the model.
4. Separate quantum and probabilistic one-way communication complexity for some total function or show that both are related.
5. Prove superquadratic lower bounds for formulas over the basis of all fan-in 2 gates.

Acknowledgments. The author thanks Gregor Gramlich for a talk on the Neˇciporuk method, which inspired this research, Georg Schnitger for stimulating discussions, and the anonymous referees for their detailed suggestions and pointing out an error in an earlier version.

REFERENCES

[1] D. Aharonov, A. Kitaev, and N. Nisan, *Quantum Circuits with Mixed States*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 20–30.

[2] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, *Quantum dense coding and quantum finite automata*, J. ACM, 49 (2002), pp. 496–511. Earlier versions appeared in STOC ’99 and FOCS ’99 (authored by Nayak alone).

[3] L. Babai, N. Nisan, and M. Szegedy, *Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs*, J. Comput. System Sci., 45 (1992), pp. 204–232.

[4] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A, 52 (1995), pp. 3457–3467.

[5] C. H. Bennett and S. J. Wiesner, *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states*, Phys. Rev. Lett., 69 (1992), pp. 2881–2884.

[6] R. B. Boppana, *Amplification of probabilistic Boolean formulas*, in Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, Portland, OR, 1985, pp. 20–29.

[7] R. B. Boppana and M. Sipser, *The Complexity of Finite Functions*, in Handbook of Theoretical Computer Science A, Elsevier, New York, 1990.

[8] H. Buhrman and R. de Wolf, *Communication complexity lower bounds by polynomials*, in Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, IL, 2001, pp. 120–130.

[9] R. Cleve and H. Buhrman, *Substituting quantum entanglement for communication*, Phys. Rev. A, 56 (1997), pp. 1201–1204.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Ser. Telecomm., 1991.

[11] D. Deutsch, *Quantum computational networks*, Proc. R. Soc. Lond. A, 425 (1989), pp. 73–90.

[12] M. Dubiner and U. Zwick, *Amplification by read-once formulas*, SIAM J. Comput., 26 (1997), pp. 15–38.

[13] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?*, Phys. Rev., 47 (1935), pp. 777–780.

[14] J. Goldsmith, M. A. Levy, and M. Mundhenk, *Limited nondeterminism*, SIGACT News, 27 (1996), pp. 20–29.

[15] J. H˚astad, *The shrinkage exponent of de Morgan formulas is 2*, SIAM J. Comput., 27 (1998), pp. 48–64.

[16] A.S. Holevo, *Some estimates on the information transmitted by quantum communication channels*, Proc. IEEE, 48 (1973), pp. 177–183.

[17] J. Hromkovic and M. Sauerhoff, *Tradeoffs between nondeterminism and complexity for communication protocols and branching programs*, in Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science, Lille, France, 2000, Lect. Notes Comput. Sci. 1770, Springer, New York, 2000, pp. 145–156.
[18] J. Hromkovič and G. Schnitger, Nondeterministic communication with a limited number of advice bits, SIAM J. Comput, 33 (2003), pp. 43–68.

[19] J. Hromkovič and G. Schnitger, On the power of Las Vegas for one-way communication complexity, OBDDs, and finite automata, Inform. Comput., 169 (2001), pp. 284–296.

[20] J. Katz and L. Trevisan, On the efficiency of local decoding procedures for error-correcting codes, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, Portland, OR, 2000, pp. 80–86.

[21] H. Klauck, On the size of probabilistic formulae, in Proceedings of the 8th International Symposium on Algorithms and Computation, Singapore, 1997, Lect. Notes Comput. Sci. 1350, Springer, 1997, pp. 243–252.

[22] H. Klauck, Lower bounds for quantum communication complexity, SIAM J. Comput., 37 (2007), pp. 20–46.

[23] H. Klauck, On quantum and probabilistic communication: Las Vegas and one-way protocols, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, Portland, OR, 2000, pp. 644–651.

[24] H. Klauck, Quantum communication complexity, in Proceedings of the Workshop on Boolean Functions and Applications at 27th ICALP, Geneva, Switzerland, 2000, pp. 241–252.

[25] H. Klauck, A. Nayak, A. Ta-Shma, and D. Zuckerman, Interaction in quantum communication and the complexity of set disjointness, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, Hersonissos, Greece, 2001, pp. 124–133.

[26] I. Kremer, N. Nisan, and D. Ron, On randomized one-round communication complexity, Comput. Complexity, 8 (1999), pp. 21–49.

[27] E. Kushilevitz and N. Nisan, Communication Complexity, Cambridge University Press, London, 1997.

[28] H. Lo and H. Chau, Why quantum bit commitment and ideal quantum coin tossing are impossible, Phys. D, 120 (1998), pp. 177–187.

[29] D. Mayers, Unconditionally secure quantum bit commitment is impossible, Phys. Rev. Lett., 78 (1997), pp. 3414–3417.

[30] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, London, 1995.

[31] A. Nayak, Lower Bounds for Quantum Computation and Communication, Ph.D. thesis, University of California, Berkeley, 1999.

[32] E. I. Nečiporuk, A Boolean function, Sov. Math. Dokl., 7 (1966), pp. 999–1000.

[33] I. Newman, Private vs. common random bits in communication complexity, Inform. Process. Lett., 39 (1991), pp. 67–71.

[34] M. A. Nielsen and I. Chuang, Programmable quantum gate arrays, Phys. Rev. Lett., 79 (1997), pp. 321–324.

[35] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, London, 2000.

[36] C. Papadimitriou and M. Sipser, Communication complexity, J. Comput. System Sci., 28, (1984), pp. 260–269.

[37] V. V. Vapnik and A. Y. Chervonenkis, On the uniform convergence of relative frequencies to their probabilities, Theory Probab. Appl., 16 (1971), pp. 264–280.

[38] A. C. Yao, Some complexity questions related to distributed computing, in Proceedings of the 11th Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1979, pp. 209–213.

[39] A. C. Yao, Quantum circuit complexity, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1993, pp. 352–361.

[40] U. Zwick, Boolean Circuit Complexity, lecture notes, Tel Aviv University, 1995.