

# Approximating Boolean Functions by OBDDs

Andre Gronemeier\*

Lehrstuhl Informatik 2  
Universität Dortmund, Germany

**Abstract.** In learning theory and genetic programming, OBDDs are used to represent approximations of Boolean functions. This motivates the investigation of the OBDD complexity of approximating Boolean functions with respect to given distributions on the inputs. We present a new type of reduction for one-round communication problems that is suitable for approximations. Using this new type of reduction, we prove the following results on OBDD approximations of Boolean functions:

1. We show that OBDDs approximating the well-known hidden weighted bit function for uniformly distributed inputs with constant error have size  $2^{\Theta(n^{1/4})}$ , improving a previously known result.
2. We prove that for every variable order  $\pi$  the approximation of some output bits of integer multiplication with constant error requires  $\pi$ -OBDDs of exponential size.

## 1 Introduction

Branching programs (BPs), also called binary decision diagrams (BDDs), are both a theoretical model for nonuniform sequential computation and a data structure for Boolean functions in applications like symbolic verification and other CAD problems. Especially restricted BP types like OBDDs have good algorithmic properties [1] and proof methods for strong lower bounds on the size of restricted BPs for concrete functions have been developed [2].

**Definition 1.** *A deterministic branching program (BP) or binary decision diagram (BDD) on the variable set  $X = \{x_1, \dots, x_n\}$  is a directed acyclic graph with one source and two sinks. The sinks are labeled by the constants 0 and 1, respectively, interior nodes are labeled by variables from  $X$  and have two outgoing edges labeled by the constants 0 and 1. The BP  $G$  computes a function  $G : \{0, 1\}^n \rightarrow \{0, 1\}$  defined on  $X$  in the following way: For an input  $a \in \{0, 1\}^n$  the output  $G(a)$  is defined as the label of the sink which is reached from the source of the graph by following the edge labeled by  $a_i$  for nodes labeled by  $x_i$ . The size  $|G|$  of a BP  $G$  is the number of its nodes. Let  $\pi$  be a permutation on the set  $\{1, \dots, n\}$ . A  $\pi$ -OBDD is a BP where  $\pi(i) < \pi(j)$  for each edge leading from a node labeled by  $x_i$  to a node labeled by  $x_j$ . In this context  $\pi$  is called a variable order. A  $\pi$ -OBDD for some unspecified variable order is simply called OBDD.*

---

\* Supported by DFG grant SA 1053/1-1

In some applications, e.g. learning theory and genetic programming, OBDDs are used to represent approximations of Boolean functions [3]. Motivated by these applications, Krause, Savický and Wegener [4] started investigating lower bounds on the size of OBDDs approximating Boolean functions.

**Definition 2.** *Let  $\mu$  be a probability distribution on  $\{0, 1\}^n$ . A  $\pi$ -OBDD approximates  $f$  with error  $\varepsilon$  with respect to  $\mu$  if  $\text{Prob}_\mu(G(x) \neq f(x)) \leq \varepsilon$ . The  $\pi$ -OBDD complexity  $\pi\text{-OBDD}_\varepsilon(f_\mu)$  of approximating  $f$  with error  $\varepsilon$  with respect to  $\mu$  is the size of a smallest  $\pi$ -OBDD which approximates  $f$  with error  $\varepsilon$  with respect to  $\mu$ . The OBDD complexity  $\text{OBDD}_\varepsilon(f_\mu)$  of approximating  $f$  with error  $\varepsilon$  with respect to  $\mu$  is  $\min_\pi \{\pi\text{-OBDD}_\varepsilon(f_\mu)\}$ . In the above notation  $\mu$  is omitted if it describes the uniform distribution.*

Although approximations of Boolean functions have been studied to prove lower bounds on the size of randomized OBDDs by Yao's min-max principle [5], these results only show lower bounds for very specific input distributions. There are few results on the OBDD complexity of approximations for given input distributions, especially important distributions like the uniform distribution [4, 6–8]. In [7] Bollig, Sauerhoff and Wegener ask how the known lower bound techniques for the exact case can be adopted to work also for approximations. In this paper, we define an appropriate type of reduction for approximation problems (Sec. 3). Then these reductions are used to obtain new lower bounds on the size of OBDDs approximating the hidden weighted bit function and integer multiplication with respect to the uniform distribution (Sec. 4).

## 2 Survey of the Results

Bryant's hidden weighted bit function [9] is a well-known benchmark function in the BP literature.

**Definition 3.** *For a vector  $x \in \{0, 1\}^n$  let  $\|x\|$  denote the number of ones in  $x$ . On the input  $x = (x_1, \dots, x_n)$  the hidden weighted bit function  $\text{HWB}_n : \{0, 1\}^n \rightarrow \{0, 1\}$  is defined by  $\text{HWB}(x) := x_{\|x\|}$  where  $x_0 := 0$ .*

While  $\text{HWB}_n$  is simple for many restricted BP types only slightly more general than OBDDs, e.g. FBDDs and  $k$ -OBDDs [10], nonetheless OBDDs computing  $\text{HWB}_n$  have size exponential in  $n$  [9]. So the HWB-function exposes a specific weakness of OBDDs. Bollig, Sauerhoff and Wegener [7] have shown that even approximations of  $\text{HWB}_n$  with constant error  $\varepsilon \in ]0, \frac{1}{2}[$  with respect to the uniform distribution require OBDDs of size at least  $2^{\Omega(n^{1/6-\delta})}$  for arbitrary constants  $\delta > 0$ . Here this lower bound is improved and a matching upper bound is shown.

**Theorem 1.** *For every constant  $\varepsilon \in ]0, \frac{1}{2}[$  the OBDD complexity of approximating  $\text{HWB}_n$  with error  $\varepsilon$  with respect to the uniform distribution is  $2^{\Theta(n^{1/4})}$ .*

Multiplication is one of the basic arithmetic functions. Naturally, the BP-complexity of multiplication has been investigated.

**Definition 4.** For a vector  $x = (x_{n-1}, \dots, x_0) \in \{0, 1\}^n$  let  $(x)_2$  denote the interpretation of  $x$  as a binary number. Then  $\text{MUL}_{i,n} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$  is the function that maps the inputs  $x, y \in \{0, 1\}^n$  to the  $i$ -th bit of the binary representation of  $(x)_2 \cdot (y)_2$ .

In his pioneering paper [1] Bryant was the first to investigate the OBDD complexity of integer multiplication. Later he proved that the computation of the middle bit of integer multiplication requires OBDDs of exponential size [9]. Since then this bound has been improved [11] and exponential lower bounds on the size of various BP types computing the middle bit of integer multiplication have been shown, e.g. for randomized OBDDs [12], FBDDs [13, 14] and read- $k$  BPs and linear length multiway BPs [8]. Surprisingly, approximating the middle bit of integer multiplication with respect to the uniform distribution is easy even for OBDDs [8]. Complementing this result, we show that no variable order is suitable for the approximation of all output bits: For each variable order there is an output bit which requires exponential OBDD size.

**Theorem 2.** Let  $\varepsilon \in ]0, \frac{1}{2}[$  be a constant. For each sequence  $\pi_n$  of variable orders there exists a sequence  $i_n \in \{0, \dots, n-1\}$  of output bits of integer multiplication with  $\pi_n$ -OBDD $_{\varepsilon}(\text{MUL}_{i_n,n}) = 2^{\Omega(n)}$ .

These results are proved using a new type of reduction for 1-round communication problems which is described in the following section. The proofs are generalizations of proofs for analogous theorems for exact computations. So we think that these reductions answer, at least partially, the aforementioned question raised in [7].

### 3 Proof Methods

#### 3.1 Communication Complexity

We will use Yao's two-player communication complexity [15] to prove lower bounds on the OBDD-complexity of Boolean functions. A thorough introduction to communication complexity can be found in [16]. Here we are only interested in 1-round protocols. For the definition of reductions we extend the classical definition of randomized public coin 1-round protocols [16] by an oracle-function.

**Definition 5.** Let  $X, Y, X', Y', M$  and  $R$  be finite sets,  $g : X' \times Y' \rightarrow Z'$  be a function and  $\rho$  be a probability distribution on the set  $R$ . A randomized 1-round communication protocol  $P[g]$  with oracle  $g$  is a communication game between two players Alice and Bob. For an input  $(x, y) \in X \times Y$  and a random input  $r \in R$ , which is chosen with respect to distribution  $\rho$ , the output  $P[g](x, y, r)$  of the protocol is computed according to the following rules:

The input  $(x, y)$  is distributed among Alice and Bob. Alice gets the private input  $x$  and Bob gets the private input  $y$ . Both players have access to the public random input  $r$ . Then the following computation and communication steps are performed:

1. Alice computes a message  $m := P_A(x, r) \in M$  and sends  $m$  to Bob.
2. Alice computes her oracle input  $q_A := Q_A(x, r) \in X'$ .
3. Bob computes his oracle input  $q_B := Q_B(m, y, r) \in Y'$ .
4. Alice and Bob query the oracle  $g$  for the input  $(q_A, q_B)$ . Bob gets the oracle's output  $z' := g(q_A, q_B) \in Z'$ .
5. Bob computes the output  $P[g](x, y, r) := P_B(m, y, r, z') \in Z$ .

The protocol  $P[g]$  computes a function  $f : X \times Y \rightarrow Z$  with error  $\varepsilon$  if  $\text{Prob}_\rho(P[g](x, y, r) \neq f(x, y)) \leq \varepsilon$  for all inputs  $(x, y) \in X \times Y$ . The cost of the protocol is  $c(P[g]) := \lceil \log_2 |M| \rceil$ . If the output of the protocol does not depend on the oracle, then the corresponding part of the notation is omitted. A deterministic 1-round communication protocol is a randomized protocol, as defined above, where the output of the protocol does not depend on the random input  $r$ .

Given a probability distribution on the input set  $X \times Y$  of a communication problem  $f : X \times Y \rightarrow Z$ , one can define approximations of functions by deterministic 1-round communication protocols. This is done analogously to approximations by OBDDs. Note that the cost of a protocol for a given Boolean function may strongly depend on how the input is distributed among the players.

**Definition 6.** Let  $f$  be a Boolean function on the variable set  $X = \{x_1, \dots, x_n\}$ . Each partition  $\Pi$  of the set  $X$  into two sets  $X_A$  and  $X_B$  defines a corresponding communication problem  $\Pi$ - $f$  where Alice receives the variables from  $X_A$  while Bob receives the variables from  $X_B$ . Given a probability distribution  $\mu$  on the inputs of  $f$ , let  $D_\varepsilon^{A \rightarrow B}(\Pi$ - $f_\mu)$  denote the cost of a cheapest deterministic 1-round communication protocol  $P$  that approximates  $\Pi$ - $f$  with error  $\varepsilon$  with respect to  $\mu$ .

Communication complexity has been used by many authors to prove lower bounds on the size of BPs computing Boolean functions (see [2]). Bollig, Sauerhoff and Wegener [7] observed that the same proof method can be applied to approximations of Boolean functions by BPs.

**Theorem 3.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function on the variable set  $X = \{x_1, \dots, x_n\}$  and let  $\mu$  be a probability distribution on the inputs of  $f$ . Further let  $\Pi = (X_A, X_B)$  be a partition of  $X$  into two sets and let  $\pi$  be a variable order on  $X$  that satisfies  $\pi(a) < \pi(b)$  for all  $x_a \in X_A$  and  $x_b \in X_B$ . Then  $\pi$ -OBDD $_\varepsilon(f_\mu) \geq 2^{D_\varepsilon^{A \rightarrow B}(\Pi$ - $f_\mu)}$ .

A partition  $\Pi = (X_A, X_B)$  of  $X$  is called  $\alpha$ -balanced if  $|X_A| = \lfloor \alpha n \rfloor$ . Then  $\text{OBDD}_\varepsilon(f_\mu) \geq \min\{2^{D_\varepsilon^{A \rightarrow B}(\Pi$ - $f_\mu)} \mid \Pi \text{ is } \alpha\text{-balanced}\}$  for any  $\alpha \in [0, 1]$ .

Since a similar theorem is proved in [7], a proof of Theorem 3 is not necessary.

### 3.2 Randomized Rectangular Reductions

In communication complexity theory the relative complexity of problems is usually investigated with *rectangular reductions* [17]: A rectangular reduction from a problem  $f : X \times Y \rightarrow Z$  to a problem  $g : X' \times Y' \rightarrow Z'$  is a pair of functions

$\rho_X : X \rightarrow X'$  and  $\rho_Y : Y \rightarrow Y'$  with the property  $g(\rho_X(x), \rho_Y(y)) = f(x, y)$ . If  $f$  is reducible to  $g$  in this way, then lower bounds on the communication complexity of  $f$  imply lower bounds on the communication complexity of  $g$ . Unfortunately, this method does not work well for approximations of Boolean functions with respect to arbitrary input distributions: For a fixed reduction the input distribution on  $X \times Y$  uniquely defines the input distribution on  $X' \times Y'$ . This complicates the proof of lower bounds for approximations of  $g$  with respect to given distributions on  $X' \times Y'$ . Here we try to solve this problem by randomizing the reduction and by allowing additional communication.

**Definition 7.** Let  $f_\mu : X \times Y \rightarrow Z$  and  $g_{\mu'} : X' \times Y' \rightarrow Z'$  be functions with probability distributions  $\mu$  and  $\mu'$  on their finite input sets  $X \times Y$  and  $X' \times Y'$ . The function  $f_\mu$  is  $\text{RD}(\varepsilon, k)$ -reducible to  $g_{\mu'}$ , written  $f_\mu \leq_{\text{RD}}^{\varepsilon, k} g_{\mu'}$ , if there is a randomized 1-round communication protocol  $P[g]$  with oracle  $g$  which has the following properties: Let the oracle inputs  $q_A$  and  $q_B$  be defined as in Definition 5. Then

1.  $c(P[g]) \leq k$ ,
2.  $\text{Prob}_{\mu, \rho}((q_A, q_B) = (x', y')) = \mu'(x', y')$  for every  $(x', y') \in X' \times Y'$  and
3.  $\text{Prob}_{\mu, \rho}(P[g](x, y, r) \neq f(x, y)) \leq \varepsilon$ .

Note that the R in RD-reduction is an abbreviation for ‘rectangular’ while D stands for ‘distributional’. The purpose of the extensions in Definition 7 compared to the simpler rectangular reductions is to control the probability distribution on the oracle inputs. The following theorem shows how  $\text{RD}(\varepsilon, k)$ -reductions can be used to prove lower bounds on the 1-round communication complexity of approximations.

**Theorem 4.** Let  $f_\mu : X \times Y \rightarrow Z$  and  $g_{\mu'} : X' \times Y' \rightarrow Z'$  be functions with probability distributions  $\mu$  and  $\mu'$  on their finite input sets. If  $f_\mu \leq_{\text{RD}}^{\varepsilon, k} g_{\mu'}$ , then  $D_{\varepsilon + \varepsilon'}^{\text{A} \rightarrow \text{B}}(f_\mu) \leq D_{\varepsilon'}^{\text{A} \rightarrow \text{B}}(g_{\mu'}) + k$  for any constant  $\varepsilon' \in ]0, \frac{1}{2}[$ .

*Proof.* Let  $F[g]$  denote the randomized 1-round protocol with oracle  $g$  that proves  $f_\mu \leq_{\text{RD}}^{\varepsilon, k} g_{\mu'}$  according to Definition 7. By the definition of  $\text{RD}(\varepsilon, k)$ -reductions, the cost of  $F[g]$  is bounded by  $k$ . For every constant  $\varepsilon' \in ]0, \frac{1}{2}[$  there exists a deterministic 1-round protocol  $G$  which approximates  $g$  with error  $\varepsilon'$  with respect to  $\mu'$  whose cost is bounded by  $D_{\varepsilon'}^{\text{A} \rightarrow \text{B}}(g_{\mu'})$ . Then we can obtain a deterministic 1-round protocol  $P$  that approximates  $f$  with error  $\varepsilon + \varepsilon'$  with respect to  $\mu$  in the following way: The query of the  $g$ -oracle in  $F[g]$  is replaced by the execution of the protocol  $G$ . The resulting protocol  $P_{\text{rand}}$  is a randomized 1-round protocol that approximates  $f$ . By construction, the cost of  $P_{\text{rand}}$  is bounded by  $D_{\varepsilon'}^{\text{A} \rightarrow \text{B}}(g_{\mu'}) + k$ . The output  $P_{\text{rand}}(x, y, r)$  of the constructed protocol can be different from  $f(x, y)$  because of the following reasons:

- (1) The protocol  $F[g]$  computes the right output, but  $P_{\text{rand}}(x, y, r)$  is different from  $F[g](x, y, r)$ .
- (2) Even the protocol  $F[g]$  computes the wrong result for inputs  $x, y$  and  $r$ .

By Definition 7, the probability of (2) is bounded by  $\varepsilon$ . By the definition of the protocol  $G$  and by Definition 7, the probability of (1) is bounded by  $\varepsilon'$  because in this case the output of  $G$  differs from the oracle output. In all, the approximation error of protocol  $P_{\text{rand}}$  is  $\text{Prob}_{\mu,\rho}(P_{\text{rand}}(x,y,r) \neq f(x,y)) \leq \varepsilon + \varepsilon'$ . Then a well-known averaging argument [5] shows that for some fixed  $r^* \in R$   $\text{Prob}_{\mu}(P_{\text{rand}}(x,y,r^*) \neq f(x,y)) \leq \varepsilon + \varepsilon'$ . Replacing the random input  $r$  in protocol  $P_{\text{rand}}$  by the constant  $r^*$  yields the desired deterministic protocol  $P$  that approximates  $f$  with error  $\varepsilon + \varepsilon'$ .  $\square$

## 4 Proofs of the Main Results

Now we will prove the results from Section 2 by RD-reductions from the so-called index function for different distributions on the inputs.

**Definition 8.** For inputs  $(x_1, \dots, x_n) \in \{0, 1\}^n$  and  $i \in \{1, \dots, n\}$  the index function  $\text{IND}_n$  is defined by  $\text{IND}_n(x, i) := x_i$ .

The proofs are essentially randomized versions of proofs of similar theorems for exact computations. This underlines our claim that randomized reductions can be used to adopt the proof methods for exact computations to approximations.

### 4.1 Hidden Weighted Bit Function

Bollig, Sauerhoff and Wegener [7] observed that if Alice owns many variables with index close to  $\frac{n}{2}$ , then the hidden weighted bit function  $\text{HWB}_n$  for uniformly distributed inputs is very similar to the index function for uniformly distributed values of the data variables  $x$  and binomially distributed values of the index  $i$ .

**Definition 9.** Let  $\text{bin}(n)$  denote the product distribution of the uniform distribution on the set  $\{0, 1\}^n$  and the binomial distribution with parameters  $n - 1$  and  $\frac{1}{2}$ . Then for the inputs  $x = (x_0, \dots, x_{n-1}) \in \{0, 1\}^n$  and  $i \in \{0, \dots, n - 1\}$  the index function with respect to  $\text{bin}(n)$  is defined by  $\text{IND}_{\text{bin}(n)}(x, i) := x_i$ .

Compared to Definition 8, here the numbering of the indices is adjusted to the usual definition of the binomial distribution. Bollig *et al.* [7] also proved that  $D_{\varepsilon}^{\text{A} \rightarrow \text{B}}(\text{IND}_{\text{bin}(n+1)}) = \Omega(n^{\frac{1}{2} - \delta})$  for arbitrary constants  $\delta > 0$ . We will use a slightly improved version of this result.

**Theorem 5.**  $D_{\varepsilon}^{\text{A} \rightarrow \text{B}}(\text{IND}_{\text{bin}(n+1)}) = \Theta(n^{\frac{1}{2}})$  for every constant  $\varepsilon \in ]0, \frac{1}{2}[$ .

To prove this result, the proof from [7] only needs to be modified in a few places. Details are given in the Appendix. Now Theorem 5 and the similarity of  $\text{HWB}_n$  and  $\text{IND}_{\text{bin}(n)}$  stated above can be used to prove Theorem 1 by a randomized reduction.

*Proof (Theorem 1).* We claim that for every constant  $\varepsilon' \in ]0, \frac{1}{2}[$  there are constants  $\alpha \in ]0, 1[$ ,  $c_{\varepsilon'} \in ]0, \infty[$  and  $n_{\varepsilon'} \in \mathbb{N}$  such that for all  $n \geq n_{\varepsilon'}$  and all  $\alpha$ -balanced partitions  $\Pi$  of the input variables of  $\text{HWB}_n$

$$\text{IND}_{\text{bin}(\lfloor c_{\varepsilon'} n^{1/2} \rfloor)} \leq_{\text{RD}}^{\varepsilon', 0} \Pi\text{-HWB}_n . \quad (*)$$

Then, by Theorem 4,  $D_\varepsilon^{A \rightarrow B}(\Pi\text{-HWB}_n) \geq D_{\varepsilon + \varepsilon'}^{A \rightarrow B}(\text{IND}_{\text{bin}(\lfloor c_{\varepsilon'} n^{1/2} \rfloor)})$ . Since  $\varepsilon + \varepsilon' < \frac{1}{2}$  for  $\varepsilon' := \frac{1}{2}(\frac{1}{2} - \varepsilon)$ , the lower bound from Theorem 1 is implied by Theorem 5. We still have to show that (\*) holds.

Let  $n' := \lfloor c_{\varepsilon'} n^{1/2} \rfloor$  for the constant  $c_{\varepsilon'}$  which will be fixed later on. For the proof of (\*) we consider  $\alpha$ -balanced partitions  $\Pi = (X_A, X_B)$  of the input variables  $X = \{x_1, \dots, x_n\}$  of  $\text{HWB}_n$  for constants  $\alpha$  where  $|X_A| = n - n' + 1$  and  $|X_B| = n' - 1$ . We use the following randomized  $\Pi$ - $\text{HWB}_n$ -oracle protocol to show that  $\text{IND}_{\text{bin}(n')}$  is  $\text{RD}(\varepsilon', 0)$ -reducible to  $\Pi$ - $\text{HWB}_n$ :

Let  $(x', i) \in \{0, 1\}^{n'} \times \{0, \dots, n' - 1\}$  be the input of  $\text{IND}_{\text{bin}(n')}$ . The random inputs of the protocol are a random vector  $r \in \{0, 1\}^{n - 2n' + 1}$  and a random permutation  $\pi$  on the set  $\{1, \dots, n' - 1\}$ . Both random inputs are chosen with respect to the uniform distribution. The output of the protocol is computed according to the following rules:

1. Alice computes  $i_0 := \|x'\| + \|r\|$  and assigns the following values to the input variables  $X_A$  of her oracle input:
  - For  $x_k \in X_A$  with  $k = i_0 + j$  and  $j \in \{0, \dots, n' - 1\}$  Alice sets  $x_k := x'_j$ .
  - The remaining unassigned variables from  $X_A$  are assigned values from the input variables in  $x'$  and the random input  $r$  in an arbitrary but fixed order, such that each of the input bits from  $x'$  and  $r$  is used exactly once.
2. Bob computes  $b = (b_1, \dots, b_{n'-1}) := \pi\left(\sum_{j=1}^i e_j\right)$  where  $e_j$  denotes the  $j$ -th unit vector and  $\pi(v)$  denotes the permutation of the components of a vector  $v$  with respect to the permutation  $\pi$ . Then Bob assigns the values  $b_j$ ,  $j = 1, \dots, n' - 1$  to the variables  $X_B$  of his oracle input with respect to an arbitrary but fixed one-to-one mapping.
3. Alice and Bob query the  $\Pi$ - $\text{HWB}_n$ -oracle. Bob uses the result  $z$  as the output of the protocol.

Obviously, the cost of the protocol is 0 since no communication takes place. To prove (\*) we have to show that the oracle inputs are uniformly distributed and that the approximation error of the protocol is bounded by  $\varepsilon'$ : By construction, the mapping from inputs  $x'$  and  $r$  to Alice's oracle inputs is a bijection. So Alice's oracle inputs are uniformly distributed because the inputs  $x'$  and  $r$  are uniformly distributed. A simple calculation shows that Bob's oracle inputs are uniformly distributed: Let  $v := \sum_{j=1}^i e_j$  for the input  $i$  of the index function. By the distribution of the input  $i$ , the probability of  $v$  having exactly  $i$  bits with value 1 is  $\binom{n'-1}{i} 2^{-(n'-1)}$ . The random permutation  $\pi$  maps  $v$  to a fixed vector  $b \in \{0, 1\}^{n'-1}$  that satisfies  $\|b\| = i$  with probability  $\binom{n'-1}{i}^{-1}$ . Multiplying both probabilities yields the uniform distribution.

Let  $x$  be the oracle input of the protocol for input  $(x', i)$  and random inputs  $r$  and  $\pi$ . The output of the protocol is  $z = x_{\|x\|} = x_{\|x'\| + \|r\| + \|b\|} = x_{i_0 + i}$ . By the assignment of the variables from  $X_A$  in the first step of the protocol, the output of the protocol is correct if  $x_{i_0 + i} \in X_A$ . Thus an error of the protocol implies  $x_{i_0 + i} \in X_B$ . We use the following estimates to bound the probability of this event

(see [18]): For any  $k \in \{0, \dots, n\}$

$$\binom{n}{k} 2^{-n} \leq \binom{n}{\lfloor \frac{n}{2} \rfloor} 2^{-n} = \left(\frac{\pi}{2}n\right)^{-\frac{1}{2}} \left(1 + O\left(\frac{1}{n}\right)\right).$$

The random variable  $i_0 + i$  is distributed with respect to the binomial distribution with parameters  $n$  and  $\frac{1}{2}$  since the random vectors  $x'$ ,  $r$  and  $b$  are uniformly distributed. Then, by the above inequalities and by the choice of  $|X_B|$ , we get

$$\begin{aligned} \text{Prob}(x_{i_0+i} \in X_B) &= \sum_{x_j \in X_B} \text{Prob}(i_0 + i = j) \\ &\leq |X_B| \left(\frac{\pi}{2}n\right)^{-\frac{1}{2}} \left(1 + O\left(\frac{1}{n}\right)\right) \\ &\leq c_{\varepsilon'} \left(\frac{\pi}{2}\right)^{-\frac{1}{2}} \left(1 + O\left(\frac{1}{n}\right)\right). \end{aligned}$$

If the constant  $c_{\varepsilon'}$  is chosen sufficiently small to satisfy  $c_{\varepsilon'} \left(\frac{\pi}{2}\right)^{-\frac{1}{2}} < \varepsilon'$ , then  $\text{Prob}(x_{i_0+i} \in X_B) \leq \varepsilon'$  for sufficiently large  $n$ . Fixing  $c_{\varepsilon'}$  and  $n_{\varepsilon'}$  to appropriate values completes the proof of (\*). The proof of the matching upper bound is contained in the Appendix.  $\square$

## 4.2 Integer Multiplication

We will prove Theorem 2 by a randomized reduction from the index function for uniformly distributed inputs to integer multiplication. Approximations of the index function for uniformly distributed inputs were studied by Kremer, Nisan and Ron [19] and Krause, Savický and Wegener [4]. In [7], the index function is used to prove lower bounds on the OBDD complexity of approximations. The following theorem is implicitly contained in [7].

**Theorem 6.** *The 1-round communication complexity of approximating  $\text{IND}_n$  with respect to the uniform distribution is  $D_{\varepsilon}^{\text{A} \rightarrow \text{B}}(\text{IND}_n) = \Theta(n)$  for every constant  $\varepsilon \in ]0, \frac{1}{2}[$ .*

To simplify the proof of Theorem 2, we first define some notation to translate between integers and the corresponding binary representation.

**Definition 10.** *Given a nonnegative integer  $x$ , let  $x_{[i]}$  denote the  $i$ -th bit of the binary representation of  $x$  and let  $x_{[i,j]}$  denote the vector  $(x_{[i]}, x_{[i-1]}, \dots, x_{[j]})$ . As usual, bits are counted from the least significant bit starting with 0.*

In the proof of Theorem 2 we will need a lemma about the distribution of the first  $n$  output bits of the product  $x \cdot y$  when the inputs  $x$  and  $y$  are chosen independently and uniformly at random from  $\mathbb{Z}_{2^n}$ .

**Lemma 1.** *Let  $\mathbb{Z}_{2^n} := \{0, 1, 2, \dots, 2^n - 1\}$  and let  $i, m \in \{0, \dots, n-1\}$  with  $i \geq 2m$ . If  $x$  and  $y$  are chosen independently and uniformly at random from  $\mathbb{Z}_{2^n}$ , then  $\text{Prob}((x \cdot y)_{[i, i-m+1]} = a) \leq 2^{-m+1}$  for every  $a \in \{0, 1\}^m$ .*

A similar result was shown by Dietzfelbinger *et al.* [20] in the analysis of a simple class of multiplicative universal hash functions. The proof of this lemma is contained in the Appendix. The following combinatorial lemma, which will be useful in the proof of Theorem 2, is also proved in the Appendix.

**Lemma 2.** *Given sets  $X = Y = \{0, \dots, n-1\}$  and subsets  $X_A \subseteq X, Y_A \subseteq Y$  with  $|X_A| + |Y_A| = n$ , let  $X_B := X \setminus X_A$  and  $Y_B := Y \setminus Y_A$ . Then for some  $i \in \{\lceil c \cdot n \rceil - 1, \dots, n-1\}$  there is a subset  $P \subseteq X_A \times Y_B$  or  $P \subseteq Y_A \times X_B$  with  $a + b = i$  for all  $(a, b) \in P$  and  $|P| \geq c \cdot n$  where  $c = 17 - \sqrt{17^2 - 1}$ .*

Now we can prove Theorem 2. Note that this proof is essentially a randomized version of Bryant's first result [1] on the OBDD complexity of multiplication. By randomized reductions we are able to extend Bryant's proof method to approximation problems.

*Proof (Theorem 2).* For notational convenience we sometimes do not distinguish vectors  $x \in \{0, 1\}^n$  from the corresponding integers  $(x)_2 \in \mathbb{N}$ . In both cases we simply write  $x$ . The meaning of the variables should be evident from the operators applied to the variables.

We claim that for arbitrary constants  $\varepsilon' \in ]0, \frac{1}{2}[$  the following holds for constants  $c_{\varepsilon'} \in ]0, 1[$  and  $n_{\varepsilon'} \in \mathbb{N}$ : For every  $\frac{1}{2}$ -balanced partition  $\Pi$  of the input variables of  $\text{MUL}_{i,n}$  with  $n \geq n_{\varepsilon'}$  there exists an output bit  $i \in \{0, \dots, n-1\}$  with

$$\text{IND}_{\lfloor c_{\varepsilon'} n \rfloor} \stackrel{\varepsilon', O(1)}{\leq_{\text{RD}}} \Pi\text{-MUL}_{i,n} . \quad (*)$$

Then, by Theorem 4,  $D_{\varepsilon}^{\text{A} \rightarrow \text{B}}(\Pi\text{-MUL}_{i,n}) \geq D_{\varepsilon + \varepsilon'}^{\text{A} \rightarrow \text{B}}(\text{IND}_{\lfloor c_{\varepsilon'} n \rfloor}) - O(1)$ . Since  $\varepsilon + \varepsilon' < \frac{1}{2}$  for  $\varepsilon' := \frac{1}{2}(\frac{1}{2} - \varepsilon)$ , the lower bound from Theorem 2 is implied by Theorem 6. We still have to show that (\*) holds for every constant  $\varepsilon' \in ]0, \frac{1}{2}[$ .

For  $X = \{x_0, \dots, x_{n-1}\}$  and  $Y = \{y_0, \dots, y_{n-1}\}$  let  $X \cup Y$  be the set of the input variables of  $\text{MUL}_{i,n}$ . Given the  $\frac{1}{2}$ -balanced partition  $\Pi$  of  $X \cup Y$ , let  $X_A$  and  $Y_A$  denote the set of Alice's input variables from  $X$  and  $Y$ , respectively. Let  $X_B$  and  $Y_B$  be analogously defined for Bob's input variables. By Lemma 2, there exists a subset  $P \subseteq X_A \times Y_B$  (or  $P \subseteq Y_A \times X_B$ ) where  $|P| \geq (17 - \sqrt{17^2 - 1})n$  and  $k + l = i^*$  for some  $i^* \in \{0, \dots, n-1\}$  and all  $(x_k, y_l) \in P$  (or  $(y_k, x_l) \in P$ ). Let the output bit  $i$  in (\*) be defined by  $i := i^*$ . W.l.o.g. assume  $P \subseteq X_A \times Y_B$  and restrict  $P$  to a subset  $P' \subseteq P$  in the following way: Let  $m_{\varepsilon'} \in \mathbb{N}$  be a constant which will be fixed later on. If  $(x_k, y_l) \in P'$ , then  $k \geq 2m_{\varepsilon'}$  and for all  $j \in \{1, \dots, 2m_{\varepsilon'}\}$  no pair in  $P'$  contains the variable  $x_{k-j}$ . Obviously, this restriction can be satisfied by sets  $P'$  of size  $|P'| \geq \left\lfloor \frac{|P|}{2m_{\varepsilon'} + 1} \right\rfloor \geq \left\lfloor \frac{1}{2m_{\varepsilon'} + 1} (17 - \sqrt{17^2 - 1})n \right\rfloor$ . We fix the constant  $c_{\varepsilon'}$  to  $\frac{1}{2m_{\varepsilon'} + 1} (17 - \sqrt{17^2 - 1})$ . Let  $X'_A$  be the set of variables from  $X_A$  which are contained in pairs from  $P'$  and let  $Y'_B$  be the set of variables from  $Y_B$  which are contained in pairs from  $P'$ . Let  $n' := |P'|$ . Then, by construction,  $|X'_A| = |Y'_B| = n'$ . Define the function  $\lambda: \{1, \dots, n'\} \rightarrow \{0, \dots, n-1\}$  that maps the integers from  $\{1, \dots, n'\}$  to the indices of the variables from  $X'_A$  in ascending order. If all input variables from  $Y \setminus Y'_B$  have value 0, then Bob can choose  $x_{\lambda(k)}$  as the output of  $\text{MUL}_{i,n}$  by assigning the value 1 to  $y_{i-\lambda(k)}$  and assigning the value 0 to the remaining variables from  $Y'_B$ . Since we are interested

in uniformly distributed assignments of the variables from  $Y$ , we have to modify this strategy: Let  $x$  and  $y$  be the numbers that are multiplied. Instead of setting  $y := 2^{i-\lambda(k)}$  we choose  $y := r \pm 2^{i-\lambda(k)}$  for a random number  $r \in \mathbb{Z}_{2^n}$  and compute  $z := (x \cdot (r \pm 2^{i-\lambda(k)}))_{[i]}$ . Then, with some additional communication, we try to estimate  $(x \cdot 2^{i-\lambda(k)})_{[i]}$  from this value. The following protocol shows that  $\text{IND}_{n'}$  can be approximated with error  $\varepsilon'$  by a 1-round  $\Pi$ - $\text{MUL}_{i,n}$ -oracle protocol of cost  $O(1)$ :

Let  $x' = (x'_1, \dots, x'_{n'}) \in \{0, 1\}^{n'}$  and  $j \in \{1, \dots, n'\}$  be the inputs of  $\text{IND}_{n'}$ . The protocol uses the random inputs  $r_{X_A} \in \{0, 1\}^{|X_A| - n'}$ ,  $r_{X_B} \in \{0, 1\}^{|X_B|}$  and  $r_Y \in \{0, 1\}^n$ . The random inputs are chosen with respect to the uniform distribution. The protocol uses a  $\Pi$ - $\text{MUL}_{i,n}$ -oracle. Let  $x = (x_{n-1}, \dots, x_0)$  and  $y = (y_{n-1}, \dots, y_0)$  denote the numbers that are multiplied by the oracle. Alice and Bob compute the output of the protocol according to the following rules:

1. Alice computes her oracle input:
  - Alice sets  $(x_{\lambda(1)}, \dots, x_{\lambda(n')}) := (x'_1, \dots, x'_{n'})$ .
  - Variables from  $X_A \setminus X'_A$  are assigned values from  $r_{X_A}$  with respect to some arbitrary but fixed one-to-one mapping of the individual bits.
  - Variables from  $Y_A$  are assigned values from  $r_Y$  subject to the condition  $y = r_Y$ .
2. Bob computes his oracle input:
  - Variables from  $X_B$  are assigned values from  $r_{X_B}$  with respect to some arbitrary but fixed one-to-one mapping of the individual bits.
  - Bob chooses the assignment of the variables from  $Y_B$  that satisfies  $y = r_Y$ , but then he complements the variable  $y_{i-\lambda(j)}$ . Note that  $y = r_Y + 2^{i-\lambda(j)}$  if the value of  $y_{i-\lambda(j)}$  was 0 before it was complemented and  $y = r_Y - 2^{i-\lambda(j)}$  otherwise.
3. Alice knows the assignment of all variables from  $X$  because the variables from  $X_B$  are assigned values from  $r_{X_B}$  in some fixed way. Alice computes  $p = (p_{m_{\varepsilon'}}, \dots, p_0) := (x \cdot r_Y)_{[i, i-m_{\varepsilon}]}$  and sends  $p$  to Bob.
4. Alice and Bob query the oracle. Bob gets  $z := \text{MUL}_{i,n}(x, y)$ .
5. Bob computes the output of the protocol: Note that Bob knows the assignment of the variables  $x_{\lambda(j)-1}, \dots, x_{\lambda(j)-m_{\varepsilon'}}$  since, by the choice of  $P' \subseteq P$ , these variables are assigned values from the public random input  $r_{X_A}$  in a fixed way. If  $y = r_Y + 2^{i-\lambda(j)}$ , then Bob computes  $z' := ((0, x_{\lambda(j)-1}, \dots, x_{\lambda(j)-m_{\varepsilon'}})_2 + (p)_2)_{[m_{\varepsilon}]}$ , otherwise Bob computes  $z' := ((1, \bar{x}_{\lambda(j)-1}, \dots, \bar{x}_{\lambda(j)-m_{\varepsilon'}})_2 + (p)_2)_{[m_{\varepsilon}]}$ . Then Bob uses  $z \oplus z'$  as the output of the protocol.

By construction, the cost of the protocol is  $m_{\varepsilon'} + 1$ . Since the input  $x'$  and the random inputs  $r_{X_A}$ ,  $r_{X_B}$  and  $r_Y$  are uniformly distributed, by construction, the oracle inputs are also uniformly distributed. To prove (\*), we have to show that the approximation error of the protocol is bounded by  $\varepsilon'$ : The oracle of the protocol computes  $z = (x \cdot (r_Y \pm 2^{i-\lambda(j)}))_{[i]} = (x \cdot r_Y \pm x \cdot 2^{i-\lambda(j)})_{[i]}$  while we are interested in  $(x \cdot 2^{i-\lambda(j)})_{[i]}$ . For brevity, let  $a := x \cdot r_Y$  denote the first term

in the above sum and let  $b := x \cdot 2^{i-\lambda(j)}$  denote the second term. Since we are only interested in the  $i$ -th output bit of the multiplication, all computations concerning the multiplication can be done modulo  $2^{i+1}$ . We inspect the cases  $z = (a + b)_{[i]}$  and  $z = (a - b)_{[i]}$  separately.

*Case 1.*  $z = (a + b)_{[i]}$ : Obviously,  $a_{[i,0]} = 2^{i-m_{\varepsilon'}} \cdot a_{[i,i-m_{\varepsilon'}]} + a_{[i-m_{\varepsilon'}-1,0]}$  and  $b_{[i,0]} = 2^{i-m_{\varepsilon'}} \cdot b_{[i,i-m_{\varepsilon'}]} + b_{[i-m_{\varepsilon'}-1,0]}$ . Let  $c \in \{0, 1\}$  be the carry bit that is propagated into digit  $i - m_{\varepsilon'}$  when adding  $a_{[i-m_{\varepsilon'}-1,0]}$  and  $b_{[i-m_{\varepsilon'}-1,0]}$ . Then the following equation holds modulo  $2^{m_{\varepsilon'}+1}$ :

$$(a + b)_{[i,i-m_{\varepsilon'}]} = a_{[i,i-m_{\varepsilon'}]} + b_{[i,i-m_{\varepsilon'}]} + c. \quad (1)$$

Bob knows the value of  $a_{[i,i-m_{\varepsilon'}]}$  since it is equal to Alice's message  $p$ . Bob knows the value of  $b_{[i-1,i-m_{\varepsilon'}]}$  because, by the choice of the set  $P'$ , the variables  $x_{\lambda(j)-1}, \dots, x_{\lambda(j)-m_{\varepsilon'}}$  are assigned values from the public random input  $r_{X_A}$  in a fixed way. Additionally, Bob knows the oracle's output  $z = (a + b)_{[i]}$ . Now suppose that  $c = 0$  and let  $z'$  be defined like in the first case of the last step of the protocol. A simple calculation shows that in this case (1) holds for  $b_{[i,i-m_{\varepsilon'}]} = (z \oplus z', x_{\lambda(j)-1}, \dots, x_{\lambda(j)-m_{\varepsilon'}})$ . Thus, under the assumption  $c = 0$ , Bob computes the correct output  $x_{\lambda(j)} = z \oplus z'$ . If on the other hand the addition of  $c$  has an effect on  $(a + b)_{[i]}$ , then  $(a + b)_{[i-1,i-m_{\varepsilon'}]} = (0, \dots, 0)$  by the carry-rules of addition. Note that  $i \geq \lceil c_{\varepsilon'} n \rceil - 1$  by the choice of  $P$ . If we fix the constant  $n_{\varepsilon'}$  to a value that satisfies  $i \geq \lceil c_{\varepsilon'} n_{\varepsilon'} \rceil - 1 \geq 2m_{\varepsilon'} + 1$ , then  $\text{Prob}((x \cdot y)_{[i-1,i-m_{\varepsilon'}]} = (0, \dots, 0)) \leq 2^{-m_{\varepsilon'}+1}$  by Lemma 1. If we ignore the effect of  $c$  by fixing  $c$  to the constant 0, the approximation error of the protocol is increased by at most  $2^{-m_{\varepsilon'}+1}$ .

*Case 2.*  $z = (a - b)_{[i]}$ : Note that  $-b \bmod 2^{i+1}$  is the two's complement of  $b \bmod 2^{i+1}$ . Thus  $-b = -x \cdot 2^{i-\lambda(j)} = (\bar{x}_{\lambda(j)}, \dots, \bar{x}_0, 1, \dots, 1)_2 + 1$  holds modulo  $2^{i+1}$ . By the choice of  $P'$ , there are at least  $2m_{\varepsilon'}$  variables in  $X$  with an index smaller than  $\lambda(j)$ . If the addition of 1 has an effect on the value of  $(-b \bmod 2^{i+1})_{[i,i-m_{\varepsilon'}]}$ , then  $(\bar{x}_{\lambda(j)-m_{\varepsilon'}-1}, \dots, \bar{x}_{\lambda(j)-2m_{\varepsilon'}}) = (1, \dots, 1)$  must hold due to the carry-rules of addition. By the distribution on the assignments of the variables from  $X$ , the probability of this event is smaller than  $2^{-m_{\varepsilon'}+1}$ . Then, by assuming that  $(-b \bmod 2^{i+1})_{[i,i-m_{\varepsilon'}]} = (\bar{x}_{\lambda(j)}, \dots, \bar{x}_{\lambda(j)-m_{\varepsilon'}})$  and ignoring the effect of the addition of 1, the approximation error is increased by at most  $2^{-m_{\varepsilon'}+1}$ . Under this assumption we can proceed like in the case  $z = (a + b)_{[i]}$  increasing the approximation error by at most  $2^{-m_{\varepsilon'}+1}$  again. In all, the approximation error is bounded by  $2^{-m_{\varepsilon'}+2}$ . Fixing the constant  $m_{\varepsilon'}$  to an appropriate value yields  $2^{-m_{\varepsilon'}+2} \leq \varepsilon'$  which completes the proof.  $\square$

## Acknowledgements

I would like to thank Thorsten Bernholt, Martin Sauerhoff and Ingo Wegener for their valuable comments and advice on draft versions of this paper.

## References

1. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers* **C-35** (1986) 677–691
2. Wegener, I.: *Branching Programs and Binary Decision Diagrams: Theory and Applications*. SIAM, Philadelphia, PA (2000)
3. Droste, S., Heutelbeck, D., Wegener, I.: Distributed hybrid genetic programming for learning boolean functions. In: *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Springer Verlag (2000) 181–190
4. Krause, M., Savický, P., Wegener, I.: Approximations by OBDDs and the variable ordering problem. In: *Proc. of 26th ICALP*. Number 1644 in LNCS, Springer (1999) 493–502
5. Yao, A.C.: Lower bounds by probabilistic arguments. In: *Proc. of 24th FOCS*. (1982) 420–428
6. Bollig, B., Wegener, I.: Approximability and nonapproximability by binary decision diagrams. *Electronic Colloquium on Computational Complexity (ECCC)* **7** (2000)
7. Bollig, B., Sauerhoff, M., Wegener, I.: On the nonapproximability of boolean functions by OBDDs and read-k-times branching programs. *Information and Computation* **178** (2002) 263–278
8. Sauerhoff, M., Woelfel, P.: Time-space tradeoff lower bounds for integer multiplication and graphs of arithmetic functions. In: *Proc. of 35th STOC*. (2003) 186–195
9. Bryant, R.E.: On the complexity of VLSI implementations and graph representations of boolean functions with application to integer multiplication. *IEEE Transactions on Computers* **40** (1991) 205–213
10. Bollig, B., Löbbing, M., Sauerhoff, M., Wegener, I.: On the complexity of the hidden weighted bit function for various BDD models. *RAIRO Theoretical Computer Science* **33** (1999) 103–115
11. Woelfel, P.: New bounds on the OBDD-size of integer multiplication via universal hashing. In: *Proc. of 18th STACS*. Volume 2010 of *Lecture Notes in Computer Science*, Springer (2001) 563–574
12. Ablayev, F.M., Karpinski, M.: A lower bound for integer multiplication on randomized read-once branching programs. *Proc. of CSIT '99*, Electronic Edition (1999)
13. Ponzio, S.: A lower bound for integer multiplication with read-once branching programs. In: *Proc. of 27th STOC*. (1995) 130–139
14. Bollig, B., Woelfel, P.: A read-once branching program lower bound of  $\Omega(2^{n/4})$  for integer multiplication using universal hashing. In: *Proc. of 33rd STOC*. (2001) 419–424
15. Yao, A.C.: Some complexity questions related to distributed computing. In: *Proc. of 11th STOC*. (1979) 209–213
16. Kushilevitz, E., Nisan, N.: *Communication Complexity*. Cambridge University Press (1997)
17. Babai, L., Frankl, P., Simon, J.: Complexity classes in communication complexity theory. In: *Proc. of 27th FOCS*, IEEE (1986) 337–347
18. Sedgewick, R., Flajolet, P.: *An Introduction to the Analysis of Algorithms*. Addison-Wesley (1996)
19. Kremer, I., Nisan, N., Ron, D.: On randomized one-round communication complexity. In: *Proc. of 27th STOC*. (1995) 596–605
20. Dietzfelbinger, M., Hagerup, T., Katajainen, J., Penttonen, M.: A reliable randomized algorithm for the closest pair problem. *Journal of Algorithms* **25** (1997) 19–51

## A Appendix

### A.1 Proof of Theorem 5

Theorem 5 is proved almost exactly like the slightly weaker result in [7]. We just choose some parameters differently and estimate the error more carefully. Therefore, we only sketch the main ideas of the proof. The following Lemma is an improved version of Lemma 16 from [7].

**Lemma 3.** *Let  $M := \{i \mid |i - n/2| < c \cdot n^{1/2}\}$  for an arbitrary constant  $c \in ]0, \infty[$ . Then, for every constant  $\delta > 1$ , the following property holds for an appropriate constant  $m_{c,\delta} \in \mathbb{N}$ : Let  $y$  be a random variable that is binomially distributed with parameters  $n$  and  $\frac{1}{2}$ . If the set  $M$  is partitioned into  $2m_{c,\delta}$  consecutive blocks  $B_j$ ,  $j = 1, \dots, 2m_{c,\delta}$  of length  $\frac{c}{m_{c,\delta}}n^{\frac{1}{2}}$ , then for all  $i \in B_j$*

$$\text{Prob}(y = i \mid y \in B_j) \leq \delta(1 + o(1)) \frac{1}{|B_j|} .$$

*Proof.* For the sake of readability, in the proof we neglect the fact that the block size  $\frac{c}{m_{c,\delta}}n^{\frac{1}{2}}$  should be an integer. A closer inspection of the proof should convince the reader that the proof remains valid for exact calculations. We will use the following approximation of  $\binom{n}{n/2-k}2^{-n}$  for  $k = O(n^{\frac{1}{2}})$  to prove the claim of the lemma (see [18]):

$$\binom{n}{n/2-k}2^{-n} = \frac{e^{-2k^2/n}}{\sqrt{\frac{\pi}{2}n}}(1 + o(1)) .$$

W.l.o.g. we assume that the elements from  $B_j$  are not larger than  $n/2$ . Let  $k' = n/2 - k$  and  $l' = n/2 - l$  be elements from  $B_j$  where  $0 \leq k \leq l$ . Then, by the definition of  $B_j$ , we get  $k \leq c \cdot n^{\frac{1}{2}}$  and  $l - k = k' - l' \leq \frac{c}{m_{c,\delta}}n^{\frac{1}{2}}$ . Since  $l^2 - k^2 = (l - k)^2 + 2k(l - k)$ , this implies  $l^2 - k^2 \leq \left(\frac{c^2}{m_{c,\delta}^2} + \frac{2c^2}{m_{c,\delta}}\right)n$ . Further, note that

$$\frac{\text{Prob}(y = k' \mid y \in B_j)}{\text{Prob}(y = l' \mid y \in B_j)} = \frac{\text{Prob}(y = k')}{\text{Prob}(y = l')} \geq 1 .$$

Using the estimates above, we get

$$\begin{aligned} \frac{\text{Prob}(y = k')}{\text{Prob}(y = l')} &= \frac{e^{-2k^2/n}}{\sqrt{\frac{\pi}{2}n}}(1 + o(1)) \cdot \frac{\sqrt{\frac{\pi}{2}n}}{e^{-2l^2/n}}(1 - o(1)) \\ &\leq e^{2(l^2 - k^2)/n}(1 + o(1)) \\ &\leq e^{\frac{2c^2}{m_{c,\delta}^2} + \frac{4c^2}{m_{c,\delta}}}(1 + o(1)) . \end{aligned}$$

For  $i \in B_j$ , the above inequality implies

$$\text{Prob}(y = i \mid y \in B_j) \leq \frac{1}{|B_j|} e^{\frac{2c^2}{m_{c,\delta}^2} + \frac{4c^2}{m_{c,\delta}}}(1 + o(1)) .$$

Since  $e^{\frac{2c^2}{m_{c,\delta}^2} + \frac{4c^2}{m_{c,\delta}}} \leq \delta$  for sufficiently large  $m_{c,\delta}$ , the lemma holds.  $\square$

Theorem 5 can be proved by replacing Lemma 16 in Theorem 17 from [7] by our improved Lemma 3.

*Proof (Theorem 5).* The upper bound is proved in [7]. For the proof of the lower bound let  $P$  be a deterministic 1-round protocol which approximates  $\text{IND}_{\text{bin}(n+1)}$  with error  $\varepsilon < \frac{1}{2}$ . Like in Lemma 3, let  $M := \{i \mid |i - n/2| < c \cdot n^{\frac{1}{2}}\}$  and partition  $M$  into  $2m_{c,\delta}$  blocks  $B_j$ ,  $j = 1, \dots, 2m_{c,\delta}$  of length  $\frac{c}{m_{c,\delta}}n^{\frac{1}{2}}$  for constants  $c$  and  $m_{c,\delta}$  which will be fixed later on. For inputs  $(x, i)$  of  $\text{IND}_{\text{bin}(n+1)}$  the value of  $i$  is binomially distributed with parameters  $n$  and  $\frac{1}{2}$ . Then, by Chernoff bounds,  $\text{Prob}(i \in M) \geq 1 - 2\exp(-c^2/2)$ . Thus, under the assumption  $i \in M$ , the error probability of  $P$  is bounded by  $\varepsilon_M := \varepsilon(1 - 2\exp(-c^2/2))^{-1}$ . Then, by a simple averaging argument, for at least one  $j$  the error of  $P$  under the assumption  $i \in B_j$  must be bounded by  $\varepsilon_M$  too. By Lemma 3, under the assumption  $i \in B_j$ , the values of  $i$  are almost uniformly distributed. Let  $n' := |B_j|$ . Then, exactly like in [7], there is a rectangular reduction from  $\text{IND}_{n'}$  to  $\text{IND}_{\text{bin}(n+1)}$  that approximates  $\text{IND}_{n'}$  with error

$$\varepsilon' := \varepsilon_M \cdot \delta (1 + o(1)) = \varepsilon \cdot \delta (1 - \exp(-c^2/2))^{-1} (1 + o(1))$$

where  $\delta > 1$  is the constant from Lemma 3. If  $\delta$  is chosen sufficiently small and  $c$  is chosen sufficiently large, then  $\varepsilon' < \frac{1}{2}$  for sufficiently large  $n$ . The constant  $m_{c,\delta}$  is chosen with respect to the choice of  $c$  and  $\delta$ . Then the claim of Theorem 5 is implied by the rectangular reduction and Theorem 6.  $\square$

## A.2 Proof of the Upper Bound from Theorem 1

It is a well-known fact that upper bounds on the  $\pi$ -OBDD complexity of Boolean functions can be proved by showing upper bounds on the amount of memory used by nonuniform algorithms for the function that read the input variables with respect to the variable order  $\pi$ . In the following proof, this method is used to show that the lower bound in Theorem 1 is tight.

*Proof (Theorem 1, upper bound).* Let  $X = \{x_1, \dots, x_n\}$  be the set of the input variables of  $\text{HWB}_n$ . Given  $\varepsilon \in ]0, \frac{1}{2}[$ , let  $c_\varepsilon \in ]0, \infty[$  be a constant which will be fixed later on. Now define the sets

$$X_C := \{x_i \in X \mid |i - \frac{n}{2}| < c_\varepsilon n^{\frac{1}{2}}\} \quad \text{and} \quad X_T := X \setminus X_C.$$

We assume that  $|X_C| = 2c_\varepsilon n^{\frac{1}{2}}$  to simplify the proof. Exact calculations show that the proof remains valid for the exact value of  $|X_C|$ . Let  $x = (x_1, \dots, x_n)$  be the input of  $\text{HWB}_n$  and let

$$C(x) := \sum_{x_i \in X_C} x_i \quad \text{and} \quad T(x) := \sum_{x_i \in X_T} x_i.$$

Then  $\|x\| = C(x) + T(x)$  since  $(X_C, X_T)$  is a partition of  $X$ . We use the following algorithm to approximate  $\text{HWB}_n$  with respect to the uniform distribution:

1. Read the variables from  $X_T$  in some arbitrary fixed order and compute  $T(x)$ .
2. Let  $s_\varepsilon := c_\varepsilon(c_\varepsilon n^{\frac{1}{2}})^{\frac{1}{2}}$ . Read the variables from  $X_C$  in some arbitrary fixed order and compute  $C(x)$ . The variables  $x_i \in X_C$  with  $|i - (T(x) + c_\varepsilon n^{\frac{1}{2}})| < s_\varepsilon$  are stored in an array.
3. If  $x_{C(x)+T(x)}$  is among the variables stored in the array, then the stored value of  $x_{C(x)+T(x)}$  is used as output of the algorithm. Otherwise, the output is 0.

If the output of the algorithm is wrong, then at least one of the following conditions must hold:

- (1) The variable  $x_{\|x\|}$  is not contained in  $X_C$ , so  $|\|x\| - \frac{n}{2}| \geq c_\varepsilon n^{\frac{1}{2}}$ .
- (2) The variable  $x_{\|x\|}$  was not stored in the array in the second step of the algorithm because  $|\|x\| - (T(x) + c_\varepsilon n^{\frac{1}{2}})| \geq c_\varepsilon(c_\varepsilon n^{\frac{1}{2}})^{\frac{1}{2}}$ . Then  $|C(x) - c_\varepsilon n^{\frac{1}{2}}| \geq c_\varepsilon(c_\varepsilon n^{\frac{1}{2}})^{\frac{1}{2}}$  because  $\|x\| = C(x) + T(x)$ .

By Chernoff bounds, the probability of each of the above events is bounded by  $2 \exp(-c_\varepsilon^2/2)$ . In all, the approximation error of the algorithm is bounded by  $4 \exp(-c_\varepsilon^2/2)$  which is smaller than  $\varepsilon$  for sufficiently large  $c_\varepsilon$ . The algorithm reads the variables with respect to some fixed variable order  $\pi$  and uses  $O(n^{1/4})$  bits of memory. So the algorithm can be simulated by a  $\pi$ -OBDD of size  $2^{O(n^{1/4})}$  which completes the proof.  $\square$

### A.3 Proof of Lemma 1

The proof of Lemma 1 is similar to the proof in [20].

*Proof (Lemma 1).* Let  $\mathbb{Z}_{2^n}^* := \{1, 3, 5, \dots, 2^n - 1\}$ . Note that  $\mathbb{Z}_{2^n}^*$  is a group with respect to multiplication modulo  $2^n$ . Let

$$M_{k,l} := \{(x, y) \in \mathbb{Z}_{2^n}^2 \mid x = 2^k x', y = 2^l y', x', y' \text{ odd}\}.$$

Choosing  $(x, y)$  uniformly at random from  $M_{k,l}$  is equivalent to choosing independently and uniformly  $x'$  from  $\mathbb{Z}_{2^{n-k}}^*$  and  $y'$  from  $\mathbb{Z}_{2^{n-l}}^*$  and setting  $(x, y) := (2^k \cdot x', 2^l \cdot y')$ . Then, by the group properties of  $\mathbb{Z}_{2^{n-k-l}}^*$ , the value of  $(x' \cdot y')_{[n-k-l, 1]}$  is uniformly distributed implying that  $(x \cdot y)_{[n, k+l+1]} = (x' \cdot y' \cdot 2^{k+l})_{[n, k+l+1]}$  is uniformly distributed. Thus  $\text{Prob}((x \cdot y)_{[i, i-m+1]} = a \mid (x, y) \in M_{k,l}) = 2^{-m}$  for  $i \geq 2m$  and  $k+l \leq m$ . Let  $M := \bigcup_{k+l \leq m} M_{k,l}$ . Then  $\text{Prob}((x \cdot y)_{[i, i-m+1]} = a \mid (x, y) \in M) = 2^{-m}$  for  $i \geq 2m$  and obviously  $\text{Prob}((x \cdot y)_{[i, i-m+1]} = a \mid (x, y) \notin M) \leq 1$ . It is easy to show that  $\text{Prob}((x, y) \notin M) \leq 2^{-m}$ : If  $(x, y) \notin M$ , then  $x = 2^k x'$  and  $y = 2^l y'$  with  $k+l > m$  for odd  $x', y'$ . For each fixed pair  $k$  and  $l$ , the probability of this event is bounded by  $2^{-(k+l)} \leq 2^{-m}$ . Combining the above observations yields

$$\text{Prob}((x \cdot y)_{[i, i-m+1]} = a) \leq (1 - 2^{-m}) \cdot 2^{-m} + 2^{-m} \cdot 1 \leq 2^{-m+1}$$

which completes the proof.  $\square$

#### A.4 Proof of Lemma 2

Lemma 2 is used to find an output bit of multiplication which is hard to approximate by OBDDs. A similar result with less restrictive assumptions was used by Bryant [1] in the analogous result for exact computations.

*Proof (Lemma 2).* To prove the lemma, we estimate upper bounds on the value of  $c$  for different cases, such that the claim can be satisfied for some  $i$ . Combining the upper bounds on  $c$  from the separate cases will result in the constant  $c = 17 - \sqrt{17^2 - 1}$ . Note that the assumptions of the lemma imply  $|X_A| = |Y_B|$  and  $|Y_A| = |X_B|$ . Let  $\alpha \in ]0, 1[$  be a constant which is fixed later on.

*Case 1.*  $|X_A| \geq (1 + \alpha)\frac{n}{2}$  or  $|X_B| \geq (1 + \alpha)\frac{n}{2}$ : W.l.o.g. assume the first case is true and define the bijection  $f : X \rightarrow Y$  as  $f(x) := n - 1 - x$ . Note that  $x + f(x) = n - 1$  and  $|f(X_A) \cup Y_B| \leq n$ . Then the assumptions of this case imply

$$|f(X_A) \cap Y_B| = |f(X_A)| + |Y_B| - |f(X_A) \cup Y_B| \geq (1 + \alpha)n - n = \alpha n$$

and the claim can be satisfied for  $i = n - 1$  and constants  $c$  which satisfy  $c \leq \alpha n$ .

*Case 2.*  $|X_A| < (1 + \alpha)\frac{n}{2}$  and  $|X_B| < (1 + \alpha)\frac{n}{2}$ : In this case, the restrictions on the size of  $X_A$  and  $Y_A$  imply  $|X_A|, |Y_A|, |X_B|, |Y_B| > (1 - \alpha)\frac{n}{2}$ . For a constant  $\beta \in ]0, 1[$  which will be fixed later define sets

$$S := \{0, \dots, \lfloor \beta(1 - \alpha)\frac{n}{2} \rfloor\} \quad \text{and} \quad L := \{0, \dots, n - 1 - \lfloor \beta(1 - \alpha)\frac{n}{2} \rfloor\}.$$

For sets  $A, B \subseteq \{0, \dots, n - 1\}$  let  $S(A) := A \cap S$  and  $L(B) := B \cap L$ . The sets  $S(X_A)$  and  $S(X_B)$  are a partition of  $S(X)$ . Then the fact  $|S(X)| \geq \beta(1 - \alpha)\frac{n}{2}$  implies that  $|S(X_A)| \geq \beta(1 - \alpha)\frac{n}{4}$  or  $|S(X_B)| \geq \beta(1 - \alpha)\frac{n}{4}$ . W.l.o.g. assume that the first case is true and define the set

$$P' := \{(a, b) \mid a \in S(X_A), b \in L(Y_B)\}.$$

By definition,  $|Y \setminus L| \leq \beta(1 - \alpha)\frac{n}{2}$  and the lower bound on  $|Y_B|$  implies  $|L(Y_B)| \geq |Y_B| - |Y \setminus L| \geq (1 - \beta)(1 - \alpha)\frac{n}{2}$ . Combining the lower bounds on  $|S(X_A)|$  and  $|L(Y_B)|$  yields  $|P'| = |S(X_A)| \cdot |L(Y_B)| \geq \beta(1 - \beta)(1 - \alpha)^2 \frac{n^2}{8}$ . We will choose the set  $P$  as a subset of  $P'$  for an appropriate value of  $i$ : Note that  $a + b \leq n - 1$  for  $a \in S(X_A)$  and  $b \in L(Y_B)$ . There are  $n$  numbers in  $\{0, \dots, n - 1\}$  and there are  $|P'|$  pairs  $(a, b) \in P'$  with  $a + b \leq n - 1$ . Then, by averaging,  $|\{(a, b) \in P' \mid a + b = i\}| \geq \frac{|P'|}{n}$  for some  $i \in \{0, \dots, n - 1\}$ . Thus the claim can be satisfied for this  $i$  and sets of size  $c \cdot n \leq \frac{|P'|}{n}$  implying  $c \leq \frac{1}{8}\beta(1 - \beta)(1 - \alpha)^2$ .

Combining both cases shows that the claim can be satisfied in general for constants  $c$  with  $c \leq \max_{\alpha, \beta \in ]0, 1[} \{\min\{\alpha, \frac{1}{8}\beta(1 - \beta)(1 - \alpha)^2\}\}$ . Choosing  $\alpha = 17 - \sqrt{17^2 - 1}$  and  $\beta = \frac{1}{2}$  maximizes the upper bound on  $c$ .

There are  $i + 1$  ways of expressing  $i$  as the sum of two nonnegative integers when  $a + b$  and  $b + a$  are treated as different. This implies  $|P| \leq i + 1 \Leftrightarrow i \geq |P| - 1$ . Thus  $i \geq \lceil c \cdot n \rceil - 1$ .  $\square$