

# Lower Bounds on the OBDD Size of Graphs of Some Popular Functions\*

Daniel Sawitzki\*\*

University of Dortmund, Computer Science 2  
D-44221 Dortmund, Germany  
`daniel.sawitzki@cs.uni-dortmund.de`

**Abstract.** Ordered binary decision diagrams (OBDDs) are a data structure for Boolean functions which supports many useful operations. It finds many applications in logic design, CAD, model checking, and symbolic graph algorithms. Nevertheless, many simple functions are known to have exponential OBDD size w. r. t. their number of variables. In order to investigate the limits of symbolic graph algorithms which work on OBDD-represented graph instances, it is useful to have simply-structured graphs whose OBDD representation has exponential size. Therefore, we consider fundamental arithmetic and storage access functions with exponential OBDD size and transfer these results to the graphs of these functions. Concretely, lower bounds for the graphs of integer multiplication, indirect storage access, and the hidden weighted bit function are presented. Finally, an exemplary application of the result for multiplication to the analysis of a symbolic all-pairs shortest-paths algorithm is discussed.

## 1 Introduction

The representation of Boolean functions by branching programs has been extensively studied both in complexity theory and logic design and verification. Lower bounds on the branching program size imply lower bounds on the space complexity of computations. Moreover, tradeoff results for the depth and size of branching programs imply time–space tradeoffs on sequential machines (see, e. g., [4]). Therefore, there are many lower bound results on the size of restricted types of branching programs for at best simple and important functions like arithmetic functions and storage access functions.

On the other hand, in the practical area of logic design and verification, there is the need of succinct representations for Boolean functions which allow efficient algorithms for functional manipulation. In this context, *oblivious read-once branching programs* [10, 17, 27] (also called *ordered binary decision diagrams* (OBDDs)) have proved to be very useful for the implicit representation of state

---

\* Extended version of a paper presented at the SOFSEM 2005 conference [25].

\*\* Supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Research Cluster “Algorithms on Large and Complex Networks” (1126).

transition graphs and their symbolic manipulation. The research in this practical area is limited to some application-related problems and experimental analyses.

Recently, a new research branch has emerged which is concerned with the theoretical design and analysis of *symbolic algorithms* for classical graph problems on OBDD-represented graph instances. The input of these specialized heuristic algorithms consists of one or more OBDDs which represent the input graph instance in an *implicit* way avoiding an explicit enumeration of nodes and edges. For example, a directed graph  $G = (V, E)$  with  $V = \{v_0, \dots, v_{2^n-1}\}$  can be represented by its *characteristic* Boolean function  $\chi_G: \{0, 1\}^{2^n} \rightarrow \{0, 1\}$  with  $\chi_G(x, y) = 1 \Leftrightarrow (v_{|x|}, v_{|y|}) \in E$  for the binary values  $|x|, |y|$  of binary node number encodings  $x, y \in \{0, 1\}^n$ .

Symbolic algorithms have to solve problems on  $G$  by efficient functional operations offered by the OBDD data structure. Until now, symbolic methods for flow maximization [11, 22], topological sorting [30], shortest paths computation [21, 24], and component analysis [7, 8] have been presented. Most papers justify the new OBDD-based approaches by an analysis of the number of executed OBDD operations [3, 7, 8, 20] or by experimental results [11, 12, 14, 18, 31]. Newer research also tries to analyze the over-all runtime of symbolic methods, which includes the analysis of all OBDD sizes occurring during the algorithm. In general, even basic problems like reachability analysis are PSPACE-hard on OBDD-represented graphs [6]. So analyses must investigate input instances with special properties that enable sublinear runtimes w. r. t. the explicit graph size.

OBDDs during the run of a symbolic graph algorithm represent intermediate results and, therefore, typically not well-structured functions. In order to prove that an OBDD-based algorithm needs exponential time w. r. t. input and output size we have to estimate the size of these intermediate OBDDs. Although lower bound techniques for OBDDs are well-known [5, 27, 28], it is not easy to apply them in such situations. Intermediate OBDDs often check whether some condition is fulfilled. We formalize this type of function by defining the *symbolic graph* function of a vector of Boolean functions. In the following, the class of Boolean functions  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ ,  $n \in \mathbb{N}$ , is denoted by  $B_n$ .

**Definition 1.** Let  $f = (f_0, \dots, f_{m-1})$  be a vector of  $m$  Boolean functions  $f_i \in B_n$  for  $n, m \in \mathbb{N}$ ,  $0 \leq i \leq m-1$ . The function  $f$ -GRAPH  $\in B_{n+m}$  defined by

$$f\text{-GRAPH}(x_0, \dots, x_{n-1}, y_0, \dots, y_{m-1}) = \bigwedge_{i=0}^{m-1} [f_i(x) = y_i]$$

is called the symbolic graph of  $f$ .

The contribution of this paper is to transfer existing lower bounds for some fundamental functions to their corresponding symbolic graphs. These can then be applied in the construction of worst-case inputs for symbolic algorithms yielding exponential running times.

Section 2 introduces general branching programs for Boolean functions as well as two restricted types that are of particular interest in this paper. In order to disprove the reasonable assumption that lower bounds for single output

bits directly carry over to the symbolic graph of a vector of functions, Sect. 3 investigates the OBDD size of a specially constructed storage access function.

After these preliminaries, exponential lower bounds on the OBDD size of the graphs of integer multiplication (see Sect. 4), indirect storage access (see Sect. 5), and the hidden weighted bit function (see Sect. 6) are presented. Then, Sect. 7 gives an exemplary application of the result for multiplication to the analysis of a symbolic all-pairs shortest-paths algorithm. Finally, Sect. 8 gives conclusions on the work and mentions open problems.

## 2 Branching Programs for Boolean Functions

We denote the value of a binary string  $x = x_{n-1} \dots x_0 \in \{0, 1\}^n$  by  $|x| := \sum_{i=0}^{n-1} x_i 2^i$ . On the other hand, we denote by  $(a)$  the binary string with value  $a = |(a)| \in \mathbb{N}$ . Let  $X := \{x_0, \dots, x_{n-1}\}$  be a set of  $n$  Boolean variables.

**Definition 2.** Branching programs (BPs).

- (a) A branching program  $P$  defined on the variable set  $X$  is a directed acyclic graph with two kinds of nodes: Inner nodes and sink nodes. Each inner node  $v$  is labeled with a variable  $x_i =: \text{label}(v) \in X$  and left by two edges called 0- and 1-edge. Each sink  $w$  is labeled with a Boolean constant  $c =: \text{label}(w) \in \{0, 1\}$  and has no outgoing edge. A special node  $s$  is marked as source node.
- (b) Each assignment  $a = (a_0, \dots, a_{n-1}) \in \{0, 1\}^n$  to the variables in  $X$  defines a unique computation path  $p_a$  in  $P$  from  $s$  to a sink  $t_a$  by leaving inner nodes  $v$  labeled with  $x_i = \text{label}(v)$  via their  $a_i$ -edge. The function  $f \in B_n$  represented by  $P$  is defined by  $f(a) := \text{label}(t_a)$ .
- (c) The size  $\text{size}(P)$  of  $P$  is defined as its number of nodes.

We consider two restrictions of branching programs. Let  $\pi \in X^*$  be a sequence of variables from  $X$ .

**Definition 3.** A BP  $P$  is called  $\pi$ -oblivious if the sequence  $\pi(p)$  of variables visited on any path  $p$  from the source node to a sink is a subsequence of  $\pi$ .

**Definition 4.** Oblivious read-once branching programs or ordered binary decision diagrams (OBDDs).

- (a) A  $\pi$ -oblivious BP  $P$  is called  $\pi$ -oblivious read-once branching program or ordered binary decision diagram ( $\pi$ -OBDD) if  $\pi$  contains every variable from  $X$  at most once. Then,  $\pi$  is called the variable order of  $P$ .
- (b) The size of a minimal OBDD for a Boolean function  $f \in B_n$  is denoted by  $\text{OBDD}(f)$ . The size of a minimal  $\pi$ -OBDD for  $f$  is denoted by  $\pi\text{-OBDD}(f)$ .

Every Boolean function  $f \in B_n$  has a unique minimal-size  $\pi$ -OBDD  $P_{\min}$  for any variable order  $\pi$ . It is  $\pi\text{-OBDD}(f) \leq (2 + o(1))2^n/n$ .

The book of Wegener [27] gives a comprehensive survey on the topic of branching programs.

### 3 Lower Bounds Do Not Necessarily Carry over

Consider a vector  $f = (f_0, \dots, f_{m-1})$  of  $m$  Boolean functions  $f_i \in B_n$ ,  $0 \leq i \leq m-1$ . One could conjecture that any lower bound on the OBDD size for some  $f_i$  is also a lower bound on the OBDD size for the symbolic graph  $f$ -GRAPH of  $f$ . This pleasant property would make it obsolete to transfer lower bounds for  $f$  to  $f$ -GRAPH explicitly. Unfortunately, this is not the case as the following example shows.

We define generalized versions of the well-known functions DSA and ISA which will be combined to construct the counterexample function vector FSA.

**Definition 5.** Let  $n = 2^m$ ,  $m \in \mathbb{N}$ , and  $i \in \mathbb{N}_0$ . The shifted direct storage access (SDSA) function  $SDSA_{n,i} \in B_{n+m}$  is defined by

$$SDSA_{n,i}(x, y) := x_{\alpha(y)} \quad , \quad \text{where } \alpha(y) := (|y| + i) \bmod n$$

and  $x \in \{0, 1\}^n$ ,  $y \in \{0, 1\}^m$ .

That is,  $y$  controls which bit of  $x$  to output. Note that  $SDSA_{n,0} = DSA_n$  (see [27, Def. 4.3.1]).

**Definition 6.** Let  $n = 2^m$ ,  $m \in \mathbb{N}$ , and  $i \in \mathbb{N}_0$ . The shifted indirect storage access (SISA) function  $SISA_{n,i} \in B_{n+m}$  is defined by

$$SISA_{n,i}(x, y) := SDSA_{n,i}(x, SDSA_{n,0}(x, y), \dots, SDSA_{n,m-1}(x, y))$$

for  $x \in \{0, 1\}^n$ ,  $y \in \{0, 1\}^m$ .

That is, we output the bit  $x_{(\alpha(x,y)+i) \bmod n}$  for the indirect address  $\alpha(x, y) := |x|_{|y| \bmod n} \dots x_{(|y|+m-1) \bmod n}$ . Note that  $SISA_{n,0} = ISA_n$  (see [27, Def. 2.2.5]).

We now combine SDSA and SISA to obtain a counterexample to the conjecture stated above.

**Definition 7.** Let  $n = 2^m$  and  $m \in \mathbb{N}$ . The full storage access (FSA) vector  $FSA_n = (FSA_{n,0}, \dots, FSA_{n,2m-1})$  of Boolean functions  $FSA_{n,i} \in B_{n+m}$ ,  $0 \leq i \leq 2m-1$ , is defined by

$$FSA_{n,i}(x, y) := \begin{cases} SDSA_{n,i}(x, y) & , 0 \leq i \leq m-1 \\ SISA_{n,i-m}(x, y) & , m \leq i \leq 2m-1 \end{cases}$$

for  $x \in \{0, 1\}^n$  and  $y \in \{0, 1\}^m$ .

The OBDD size of FSA is dominated by the hard function SISA.

**Proposition 1.**  $OBDD(FSA_{n,m}) \geq 2^{\lfloor n/\log n \rfloor - 1}$ .

*Proof.* The OBDD size of  $FSA_{n,m} = SISA_{n,0} = ISA_n$  is bounded below by  $2^{\lfloor n/\log n \rfloor - 1}$ . (See [5] or [27, Theorem 4.3.3].)  $\square$

Let  $FSA\text{-}GRAPH_n$  be the symbolic graph of  $FSA_n$ . While the inclusion of SDSA into  $FSA_n$  did not influence the lower bound on its OBDD size (which is dominated by  $SISA_{n,0}$ ), it *does simplify* the OBDD representation of  $FSA\text{-}GRAPH_n$ :

**Proposition 2.**  $OBDD(FSA\text{-}GRAPH_n) = \mathcal{O}(n^3 \cdot \log n)$ .

*Proof.* Let each  $FSA_{n,i}$  be defined on the variables  $x \in \{0,1\}^n$  and  $y \in \{0,1\}^m$  for  $n = 2^m$ . Moreover, let  $z \in \{0,1\}^{2m}$  be variables corresponding to the  $2m$  results of the functions  $FSA_{n,0}, \dots, FSA_{n,2m-1}$ .

We choose  $\pi = (y_0, \dots, y_{m-1}, z_0, \dots, z_{2m-1}, x_0, \dots, x_{n-1})$  as variable order. The following  $\pi$ -OBDD  $P$  for  $FSA\text{-}GRAPH_n$  has size  $\mathcal{O}(n^3 \cdot \log n)$ . On the first  $3m$  layers,  $P$  consists of the complete binary tree on the variables  $y$  and  $z$  having  $\mathcal{O}(n^3)$  nodes. At each of its  $n^3$  leaves  $v_{y,z}$ ,  $y$  and  $z$  are already fixed. It remains to test if  $z_i = SDSA_{n,i}(x, y)$  for all  $0 \leq i \leq m-1$  and  $z_j = SISAN_{n,j-m}(x, y)$  for all  $m \leq j \leq 2m-1$ . This is done by connecting a chain of at most  $2m = \mathcal{O}(\log n)$  nodes testing  $x$ -variables to each leaf  $v_{y,z}$  implying size  $\mathcal{O}(n^3 \cdot \log n)$  for  $P$ .  $\square$

Nevertheless, the next sections will reveal that the exponential lower bounds for three fundamental functions actually *do* carry over to the symbolic graph scenario.

## 4 Integer Multiplication

*Integer multiplication* is one of the most important and difficult functions in logic design. All data structures considered so far whose algorithmic properties allow efficient circuit verification have exponential size for this hard function (see, e. g., [27]). Only for the special case of Wallace-tree like multipliers, a polynomial formal verification method using multiplicative binary moment diagrams (\*BMDs) has been presented by Keim et al. [16].

**Definition 8.** The integer multiplication (MUL) vector  $MUL_n = (MUL_{n,0}, \dots, MUL_{n,2n-1})$  of Boolean functions  $MUL_{n,i} \in B_{2n}$ ,  $0 \leq i \leq 2n-1$ , is defined by

$$MUL_{n,i}(x, y) = (|x| \cdot |y|)_i$$

for  $x, y \in \{0,1\}^n$ .

Especially, the OBDD size of  $MUL_{n,n-1}$  is exponential w. r. t.  $n$ .

**Theorem 1 (Woelfel [28]).**  $OBDD(MUL_{n,n-1}) \geq 2^{n/2}/61$ .

On the other hand, multiplication is simply-structured, and its graph is easy to encode into characteristic functions; Sect. 7 gives a corresponding application example in symbolic algorithm analysis.

So let  $MUL\text{-}GRAPH_n$  be the symbolic graph of  $MUL_n$ . Jukna [15] presents an exponential lower bound on the size of nondeterministic read- $k$ -times BPs for the symbolic graph of a subvector  $(MUL_{n,i})_{i \in I}$  for indices  $I \subset \{0, \dots, n-1\}$ ,  $|I| \leq \sqrt{n}$ . Despite the title of Jukna's work, this is *neither* a lower bound for the graph of multiplication *nor* does it imply the following theorem and corollary.

**Theorem 2.** Any  $\pi$ -oblivious BP for  $MUL-GRAPH_n$  whose variable sequence  $\pi$  contains each variable at most  $k$  times has a size of at least  $2^{n/((4k-1)(2^{8k}))^{-1}}$ .

An OBDD is an oblivious branching program reading each variable at most once. Choosing  $k = 1$  in Theorem 2, we obtain the following corollary.

**Corollary 1.**  $OBDD(MUL-GRAPH_n) \geq 2^{n/768-1}$ .

Wegener [26] presents a read-once projection from integer multiplication to squaring (SQU) which is so far the only way to prove exponential OBDD sizes for the latter function. It can also be used to prove a lower bound for a restricted version of the symbolic graph of SQU which verifies only a subset of the result variables.

**Corollary 2.** Let  $n := 3m + 2$  for  $m \in \mathbb{N}$  and  $SQU-GRAPH_n^* \in B_{8m+4}$  be defined by  $SQU-GRAPH_n(x, y) := \bigwedge_{i=2m+3}^{4m+2} [(|x|^2)_i = y_i]$ . It is  $OBDD(SQU-GRAPH_n^*) \geq 2^{m/768-1} \geq 2^{n/2304-2}$ .

#### 4.1 Proof of Theorem 2

In order to prove Theorem 2, we use techniques from Gergov's proof of Theorem 2 in [9].

Let the Boolean variables  $X := \{x_0, \dots, x_{n-1}\}$  and  $Y := \{y_0, \dots, y_{n-1}\}$  denote the factor variables of  $MUL-GRAPH_n$  and  $Z := \{z_0, \dots, z_{2n-1}\}$  its result variables. That is,  $MUL-GRAPH_n(x, y, z) = 1 \Leftrightarrow |x| \cdot |y| = |z|$  for  $x = x_{n-1} \dots x_0$ ,  $y = y_{n-1} \dots y_0$ , and  $z = z_{2n-1} \dots z_0$ . Let  $P$  be a  $\pi$ -oblivious BP for  $MUL-GRAPH_n$  whose variable sequence  $\pi$  contains each variable at most  $k$  times.

First, we show a lower bound for a restricted symbolic graph of  $MUL_n$ . Therefore, let  $\rho$  be some arbitrary sequence that contains each variable from  $X$ ,  $Y$ , and  $Z$  exactly  $2k =: \ell$  times and which has  $\pi$  as subsequence.

**Definition 9.** For two disjoint subsets  $S$  and  $T$  of  $X$ , an interval  $(\rho_i, \dots, \rho_j)$  of  $\rho$  is called link if  $\rho_{i+1}, \dots, \rho_{j-1} \notin S \cup T$  and either  $\rho_i \in T \wedge \rho_j \in S$  or  $\rho_i \in S \wedge \rho_j \in T$ .

**Lemma 1 (Alon and Maass [2]).** Let  $M = (m_1, \dots, m_{n\ell})$  be a sequence in which each element  $m_i \in X$  appears exactly  $\ell$  times. Suppose  $X_1 \dot{\cup} X_2$  is a partition of  $X$  into two disjoint non-empty sets. Then, there are two subsets  $S \subseteq X_1$  and  $T \subseteq X_2$  with  $|S| \geq |X_1|/2^{2\ell-1}$ ,  $|T| \geq |X_2|/2^{2\ell-1}$ , and such that the number of links between  $S$  and  $T$  in  $M$  is bounded above by  $2 \cdot \ell - 1$ .

Let  $X_1 := \{x_0, \dots, x_{\lfloor n/2 \rfloor}\}$  and  $X_2 := \{x_{\lfloor n/2 \rfloor + 1}, \dots, x_{n-1}\}$ . Due to Lemma 1, there are subsets  $S \subseteq X_1$  and  $T \subseteq X_2$  such that  $|S|, |T| \geq (n/2 - 1)/2^{2\ell-1} > n/2^{2\ell} - 1$  and there are no more than  $2 \cdot \ell - 1$  links between  $S$  and  $T$  in  $\rho$ .

Since  $D := \{(x_i, x_j) \mid x_i \in S, x_j \in T\}$  contains at least  $(n/2^{2\ell} - 1)^2$  pairs, there is some index set  $I \subseteq \{0, \dots, n-1\}$  and distance parameter  $d \in$

$\{1, \dots, n-1\}$  such that  $D' := \{(x_i, x_{i+d}) \mid i \in I\} \subseteq D$  contains at least  $(n/2^{2\ell} - 1)^2/n \geq n/2^{4\ell} - 1$  pairs and  $\max I < \min I + d$ .

Let  $MUL-GRAPH_n^* := \bigwedge_{i=\min I+d}^{\max I+d} [MUL_{n,i}(x, y) = z_i]$ . Moreover, we consider the subfunction  $f_n$  of  $MUL-GRAPH_n^*$  defined by the following partial variable assignments:

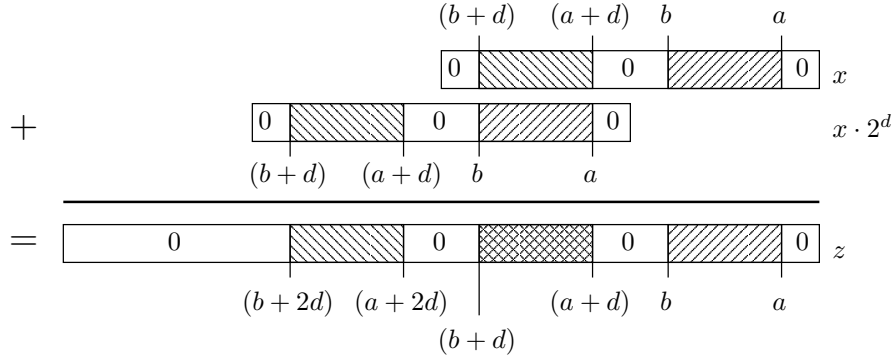
$$x_i := \begin{cases} 1 & \text{if } (i = \min I) \vee (i = \min I + d) \\ & \vee [(\min I \leq i \leq \max I) \wedge (i \notin I)] \\ 0 & \text{if } (i = \max I) \vee (i = \max I + d) \\ & \vee [((i < \min I) \vee (\max I < i)) \wedge (i - d \notin I)] \end{cases}, \quad (1)$$

$$y_j := \begin{cases} 1 & \text{if } (j = 0) \vee (j = d) \\ 0 & \text{else} \end{cases}, \quad (2)$$

$$z_r := \begin{cases} 1 & (r = \max I + d) \\ 0 & \text{else} \end{cases}. \quad (3)$$

**Lemma 2.** *Any  $\rho$ -oblivious BP  $Q$  for  $f_n$  has a size of at least  $2^{n/((2\ell-1)(2^{4\ell})-1)}$ .*

*Proof.* In (1), we replace all variables between  $\min I$  and  $\max I$  by 1 which do not take part in  $D'$  as well as  $x_{\min I}$  and  $x_{\min I+d}$ . All  $x$ -variables lying outside the interval  $[\min I, \max I]$  and not taking part in  $D'$  are replaced by 0 as well as  $x_{\max I}$  and  $x_{\max I+d}$ . In (2), the  $y$ -variables are chosen such that we sum up  $|x|$  and  $|x| \cdot 2^d$ . The result is checked against  $|z| = 2^{\max I+d}$  (see (3)). (See Fig. 1.)



**Fig. 1.** Illustration of the effect of variable replacements (1), (2), and (3) for  $a := \min I$  and  $b := \max I$ . Function  $MUL-GRAPH_n^*$  verifies only the result bits  $z_{a+d}, \dots, z_{b+d}$  which correspond to the sum of  $x_{b+d} \dots x_{a+d}$  and  $x_b \dots x_a$ .

The function  $f_n$  depends only on the  $x$ -variables being part of pairs in  $D' \setminus \{(x_{\min I}, x_{\min I+d}), (x_{\max I}, x_{\max I+d})\}$ ; it can be easily seen that it computes 1 if and only if  $x_i \neq x_{i+d}$  for all  $i \in I \setminus \{\min I, \max I\} =: I'$ . We now use methods from communication complexity (see, e. g., [13]).

*Claim.* The deterministic communication complexity of  $f_n$  w. r. t. the variable set partition  $\{x_i \mid i \in I'\} \dot{\cup} \{x_{i+d} \mid i \in I'\}$  is at least  $|I'| = |I| - 2$ .

Let  $C$  be the  $2^{|I'|} \times 2^{|I'|}$  communication matrix of  $f_n$ . It is  $C(i, j) = 1$  iff  $(i)$  is bit-inverse to  $(j)$ . Hence,  $C$  is a permutation matrix and has rank  $2^{|I'|}$ . The value  $\log(\text{rank}(C)) = |I'|$  is known to be a lower bound on the deterministic communication complexity of a Boolean function.

Due to the construction of  $I'$ , any  $\rho$ -oblivious BP  $Q$  for  $f_n$  can be partitioned into at most  $2 \cdot \ell$  parts  $S_1, T_1, \dots, S_\ell, T_\ell$  such that  $S_i$  ( $T_i$ ) contains only variables from  $S$  ( $T$ ). Therefore,  $Q$  yields a communication protocol of length  $(2 \cdot \ell - 1) \cdot \log(\text{width}(Q))$ , where the width of  $Q$  is the maximum number of nodes labeled with the same variable. (For the trivial construction see, e. g., [27, Sect. 7.5].) Due to the lower bound of  $|I'| = |I| - 2 \geq n/2^{4\ell} - 3$  on the communication complexity of  $f_n$ ,  $Q$  must have at least width (and, therefore, size)  $2^{n/((2\ell-1)(2^{4\ell})-3)/(2\ell-1)} \geq 2^{n/((2\ell-1)(2^{4\ell})-1)}$ .  $\square$

We are now able to show the lower bound on the size of  $P$ .

*Proof (Theorem 2).* The lower bound of Lemma 2 does also hold for  $P$  because we can construct an oblivious BP  $P'$  for  $f_n$  from  $P$  without enlarging it: At first, we apply variable replacements (1), (2), and (3). Then, we have to get rid of the  $z$ -variables  $z_0, \dots, z_{\min I+d-1}, z_{\max I+d+1}, \dots, z_{2n-1}$ . In satisfying inputs of  $f_n$ , it holds  $z_i = x_i$  for  $i \in I$  and  $z_j = x_{j-d}$  for  $j - 2d \in I$  (see Fig. 1). In order to force these variable pairs to be equal, we replace node labels  $z_i$ ,  $i \in I$ , by  $x_i$  as well as labels  $z_j$ ,  $j - 2d \in I$ , by  $x_{j-d}$ . The remaining  $z$ -variables are replaced by 0. The resulting BP  $P'$  represents  $f_n$ .

We added no more than  $k$  nodes for each  $z$ -variable, and  $P'$  is  $\pi'$ -oblivious for some variable sequence  $\pi'$  containing each variable  $2k = \ell$  times and having  $\pi$  as subsequence. Lemma 2 implies the lower bound of  $2^{n/((2\ell-1)(2^{4\ell})-1)} = 2^{n/((4k-1)(2^{8k})-1)}$  for  $P'$  and  $P$ .  $\square$

## 5 Indirect Storage Access

In Sect. 3 Def. 6, we defined a generalized version called SISA of the well-known ISA function. Breitbart, Hunt III, and Rosenkrantz [5] present an exponential lower bound of  $2^{\lfloor n/\log n \rfloor - 1}$  for the OBDD size of ISA. Let  $SISA\text{-GRAPH}_{n,w} \in B_{n+m+w}$  be the symbolic graph of the vector  $(SISA_{n,0}, \dots, SISA_{n,w-1})$  for  $n = 2^m$  and  $1 \leq w \leq n$ . Also  $SISA\text{-GRAPH}_{n,w}$  has superpolynomial OBDD size for  $w = o(n/\log^2 n)$ . Due to the storage access character of SISA, this restriction is not too prohibitive.

**Theorem 3.**  $\text{OBDD}(SISA\text{-GRAPH}_{n,w}) \geq 2^{n/(w \cdot \log n) - 4}$ .

*Proof.* So let  $SISA\text{-GRAPH}_{n,w}(x, y, z) = 1 \Leftrightarrow \bigwedge_{i=0}^{w-1} [SISA_{n,i}(x, y) = z_i]$  for Boolean variable sets  $X := \{x_0, \dots, x_{n-1}\}$ ,  $Y := \{y_0, \dots, y_{m-1}\}$ , and  $Z := \{z_0, \dots, z_{w-1}\}$ .

We define a subfunction  $SISA-GRAPH_{n,w}^*$  of  $SISA-GRAPH_{n,w}$  which has at least  $\pi$ -OBDD size  $2^{n/(w \cdot \log n)-4}$  for any variable order  $\pi$ . We divide  $x$  into  $\lfloor n/\log n \rfloor$  blocks of length  $m = \log n$  plus a final block of length  $\ell < m$ . Let  $X' \subset X$  denote the first  $\lfloor n/\log n \rfloor - 1$   $X$ -variables according to  $\pi$ . There is at least one block  $B = (x_b, \dots, x_{b+m-1})$  whose variables are lying completely outside of  $X'$ .

We replace  $y$  by  $(b)$  and  $z$  by  $0^w$ . Moreover, the variables  $x_{(b-w+1) \bmod n}, \dots, x_{(b-1) \bmod n}$  and  $x_{(b+m) \bmod n}, \dots, x_{(b+m+w-2) \bmod n}$  are replaced by 0. Finally, we make a cyclic traversal modulo  $n$  over those  $x$ -variables which have not been fixed yet; when passing a variable from  $X'$ , we replace the following  $w - 1$   $x$ -variables by 0. Let  $X'' \subseteq X'$  denote the subset of  $X'$ -variables not yet fixed. We repeat the traversal until any two subsequent variables from  $X''$  are separated by at least  $w - 1$  variables from  $X \setminus X''$ . Obviously,  $X''$  has at least size

$$\left\lfloor \frac{\left\lfloor \frac{n}{\log n} \right\rfloor - 1 - 2(w-1)}{w} \right\rfloor - 1 \geq \frac{n}{w \cdot \log n} - 4 .$$

The remaining variables from  $X \setminus (X'' \cup B)$  are also replaced by 0. We obtained the subfunction  $SISA-GRAPH_{n,w}^*$  depending only on  $X''$  and  $B$ .

Consider two different assignments  $a$  and  $b$  to the variables of  $X''$ . We show that the corresponding subfunctions  $f_a := SISA-GRAPH_{n,w}^* | a$  resp.  $f_b := SISA-GRAPH_{n,w}^* | b$  of  $SISA-GRAPH_{n,w}^*$  are different. Assume that  $a$  assigns 0 to variable  $x_i \in X''$ , while  $b$  assigns 1 to  $x_i$ .  $f_a$  and  $f_b$  both depend only on  $x_b, \dots, x_{b+m-1}$ .

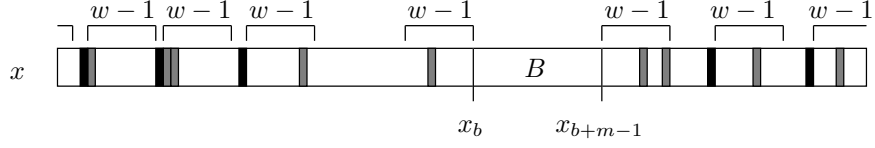
*Claim.*  $f_a((i)) = 1 \neq 0 = f_b((i))$ .

Due to the input  $B = (i)$ , the variables  $x_i, \dots, x_{i+m-1}$  are compared with  $0^w$ . All  $w - 1$  variables following a variable of  $X''$  are replaced by 0. Moreover, we have left enough space around the block  $B$ . So  $x_i = 0$  decides if the subfunction computes 1. Hence, the assignment  $B = (i)$  is a witness for  $f_a \neq f_b$ . (See Fig. 2.)

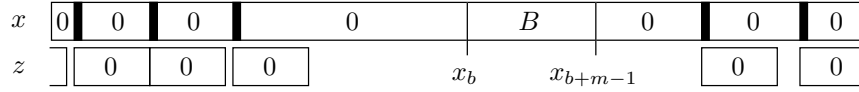
All variables of  $X''$  are tested before variables of block  $B$ . Therefore, the number  $|X''| \geq 2^{n/(w \cdot \log n)-4}$  of different subfunctions  $\{f_c \mid c \in \{0, 1\}^{|X''|}\}$  is a lower bound on the OBDD size of  $SISA-GRAPH_{n,w}^*$  (see, e. g., [27, Theorem 3.1.4]).  $\square$

## 6 The Hidden Weighted Bit Function

We now consider a generalization of one further important and fundamental storage access function.



(a) The input variables  $x$  of  $SISA-GRAPH_{n,w}$ . Black and grey cells indicate variables of  $X''$ . The visit of black cells during the traversal causes grey cells to be replaced by 0. For every black cell, we loose at most  $w-1$  grey ones. Moreover, we clean  $w-1$  cells before and after block  $B$ .



(b) The input variables of  $SISA-GRAPH_{n,w}^*$ . The function depends only on black cells and on block  $B$ . By choosing an appropriate assignment to  $B$ , the result  $z = 0^w$  can be aligned with any black cell serving as a witness for the mutual inequality of all subfunctions  $\{f_c \mid c \in \{0, 1\}^{|X''|}\}$ .

**Fig. 2.** Illustration of  $SISA-GRAPH_{n,w}^*$  and the witnesses for the inequality of all subfunctions  $\{f_c \mid c \in \{0, 1\}^{|X''|}\}$ .

**Definition 10.** Let  $n \in \mathbb{N}$  and  $i \in \mathbb{N}_0$ . The shifted hidden weighted bit (SHWB) function  $SHWB_{n,i} \in B_n$  is defined by

$$SHWB_{n,i}(x) := x_{\alpha(x)} \text{ , where } \alpha(x) := \left( \sum_{j=1}^n x_j + i \right) \bmod n \text{ ,}$$

$x = (x_1, \dots, x_n) \in \{0, 1\}^n$ , and  $x_0 := 0$ .

That is, the number of ones in  $x$  plus the shifting parameter  $i$  determines the address  $\alpha(x)$  of the output bit  $x_{\alpha(x)}$ . Note that  $SHWB_{n,0} = HWB_n$  (see [27, Def. 1.1.3]).

Wegener [27, Theorem 4.10.2] presents an exponential lower bound of  $\Omega(2^{n/5})$  on the OBDD size of the HWB function. Let  $SHWB-GRAPH_{n,w} \in B_{n+w}$  be the symbolic graph of the vector  $(SHWB_{n,0}, \dots, SHWB_{n,w-1})$  for  $n \in \mathbb{N}$  and  $1 \leq w \leq n$ . As for SISA, a certain restriction on the number  $w$  of verified result variables suffices to retain a superpolynomial lower bound.

**Theorem 4.**  $OBDD(SHWB-GRAPH_{n,w}) = \Omega(2^{n/(11w)})$ .

*Proof.* So let  $SHWB-GRAPH_{n,w}(x, y) = \bigwedge_{i=0}^{w-1} [SHWB_i(x) = y_i]$  for variables  $x \in \{0, 1\}^n$  and  $y \in \{0, 1\}^w$ . Let  $\pi$  be an arbitrary variable order of the  $x$ - and  $y$ -variables. We subsequently define a subfunction  $SHWB-GRAPH_{n,w}^*$  of  $SHWB-GRAPH_{n,w}$  yielding the lower bound on the  $\pi$ -OBDD size. First, we replace all  $y$ -variables by 0.

If the  $k$  first  $x$ -variables of  $\pi$  are assigned  $s$  ones, the number  $\alpha(x)$  of ones in  $x$  lies in the window  $\{s, \dots, s + n - k\} =: W(k, s)$ . Let  $X'(k, s)$  be the set of variables  $x_i$  whose index  $i$  lies in  $W(k, s)$  and which belong to the  $k$  first  $x$ -variables of  $\pi$ .

We now assume w.l.o.g.  $n = 11m$  for  $m \in \mathbb{N}$  and choose  $k = 7m$ . If  $s = m$ , then  $W(k, s) = \{m, \dots, 5m\}$  and if  $s = 5m$ , then  $W(k, s) = \{5m, \dots, 9m\}$ . In one of these two cases,  $X'(k, s)$  has at least  $2m$  elements.

Our goal is to “clean” enough variables with index in  $W(k, s)$  in terms of replacing them by 0 such that the comparison result  $(x_i, \dots, x_{i+w-1}) = 0^w$  depends only on the first variable  $x_i \in X'(k, s)$  for a sufficient number of different positions  $i$ . For each replaced variable, the window size  $|W(k, s)|$  shrinks at most by 1. Hence, after replacing at most  $m$  variables by 0, the address  $\alpha(x)$  may be still in the window  $W'(k, s) = \{m, \dots, 4m\}$  for  $s = m$  resp.  $W'(k, s) = \{5m, \dots, 8m\}$  for  $s = 5m$ . Due to  $|X'(k, s)| \geq 2m$ , there are still at least  $m$  variables of  $X'(k, s)$  with indices in  $W'(k, s)$ . We present an appropriate replacement scheme similar to the proof of Theorem 3:

At first, we replace  $x_{s+n-k-m-w+2}, \dots, x_{s+n-k-m}$  by 0. Then, we traverse  $x_s, \dots, x_{s+n-k-m-w+1}$ ; when visiting a variable of  $X'(k, s)$ , we replace the following  $w - 1$  variables by 0. We repeat this until exactly  $\lfloor (m - (w - 1))/w \rfloor =: c$  variables of  $X'(k, s)$  have been visited this way without replacing them; we denote their set by  $X''(k, s)$ . No more than  $(w - 1) + c \cdot (w - 1) \leq m$  variables have been replaced by 0. We finished the definition of subfunction  $SHWB-GRAPH_{n,w}^*$ .

Denote by  $X'''(k, s)$  the set of variables  $SHWB-GRAPH_{n,w}^*$  is still depending on and which belong to the first  $k$  positions of  $\pi$ .

*Claim.* Consider two different assignments  $a$  and  $b$  to the variables of  $X'''(k, s)$  such that  $s$  ones are assigned to the first  $k$   $x$ -variables of  $\pi$ . If  $a$  and  $b$  differ in some variable  $x_i \in X''(k, s) \subseteq X'''(k, s)$ , the corresponding subfunctions  $f_a := SHWB-GRAPH_{n,w}^* | a$  and  $f_b := SHWB-GRAPH_{n,w}^* | b$  are different.

Due to the definition of  $SHWB-GRAPH_{n,w}^*$ , it is  $i \in W'(k, s)$ . That is, we can complete  $a$  and  $b$  by an assignment  $\gamma$  such that  $x$  contains  $i$  ones. Therefore, the graph of shifted hidden weighted bit compares the word  $(x_i, \dots, x_{i+w-1})$  with  $0^w$ . Because  $a$  and  $b$  differ in  $x_i$  and  $x_{i+1}, \dots, x_{i+w-1}$  have been replaced by 0, it is  $f_a(\gamma) \neq f_b(\gamma)$ .

We now define  $\binom{n}{k} = 0$  for  $k < 0$  or  $k > n$ . Consider the case  $s = m$ . Because of  $|X'''(k, s)| - |X''(k, s)| \geq k - c - m$  there are at least

$$\begin{aligned} & \binom{c}{s - (k - c - m)} + \cdots + \binom{c}{s} \\ & \geq \binom{\lfloor m/w - 1 \rfloor}{m - (7m - \lceil m/w \rceil - m)} + \cdots + \binom{\lfloor m/w - 1 \rfloor}{m} \\ & = \binom{\lfloor m/w - 1 \rfloor}{0} + \cdots + \binom{\lfloor m/w - 1 \rfloor}{\lfloor m/w - 1 \rfloor} = 2^{\lfloor m/w - 1 \rfloor} \end{aligned}$$

different assignments to the variables of  $X''(k, s)$  such that the  $k$  first  $x$ -variables due to  $\pi$  contain exactly  $s$  ones. For  $s = 5m$ , this number is also at least

$$\begin{aligned} & \binom{\lfloor m/w - 1 \rfloor}{5m - (7m - \lceil m/w \rceil - m)} + \cdots + \binom{\lfloor m/w - 1 \rfloor}{5m} \\ & = \binom{\lfloor m/w - 1 \rfloor}{0} + \cdots + \binom{\lfloor m/w - 1 \rfloor}{\lfloor m/w - 1 \rfloor} = 2^{\lfloor m/w - 1 \rfloor} . \end{aligned}$$

Hence, fixing the first  $k$   $x$ -variables of  $\pi$  yields at least  $2^{\lfloor m/w - 1 \rfloor} = \Omega(2^{n/(11w)})$  different subfunctions of  $SHWB-GRAPH_{n,w}$  implying the lower bound on its OBDD size (see, e. g., [27, Theorem 3.1.4]).  $\square$

That is, the OBDD size of  $SHWB-GRAPH_{n,w}$  is superpolynomial w. r. t.  $n$  if  $w = o(n/\log n)$ . Due to the storage access character of SHWB, this restriction is not too prohibitive. Moreover, for  $n = 11m$  and  $m \in \mathbb{N}$  the lower bound is  $2^{\lfloor n/(11w) \rfloor - 1}$ .

## 7 An Application to Symbolic Algorithm Analysis

Finally, we present an application of the lower bound for the graph of integer multiplication to the analysis of a symbolic algorithm for the all-pairs shortest-paths problem in OBDD-represented weighted graphs. Input is the OBDD of a graph  $G = (V, E, c)$ ,  $c: E \rightarrow \mathbb{N}$ , while the output OBDD represents the shortest path distances  $\text{dist}: V^2 \rightarrow \mathbb{N}_0$ . The algorithm presented in [24] (called  $\mathcal{A}$  in the following) has polylogarithmic runtime  $\mathcal{O}(\log^3(|V| \cdot c^{\max}))$ ,  $c^{\max} := \{c(e) \mid e \in E\}$ , if both input and output OBDD have constant width while not skipping any variable tests. Such functions are called *bounded-width functions* (see Def. 12). However, the method works only for strictly positive edge weights  $c(e) \in \mathbb{N}_{>0}$ .

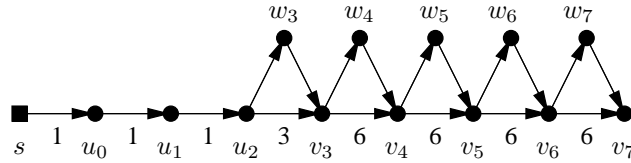
Another approach mentioned in [24] works also for weights  $c(e) = 0$ . It iteratively generates OBDDs for functions  $S^{(k)}$  with  $S^{(k)}(x, y, d) = 1$  iff there is a shortest path from node  $|x|$  to node  $|y|$  of length  $|d|$  visiting no more than  $2^k$  edges. We call this algorithm  $\mathcal{B}$ . In order to justify algorithm  $\mathcal{A}$ , it can be shown that  $\mathcal{B}$  has *not* necessarily polylog. runtime on constant width OBDDs. Similar to [23, Sect. 7], we construct an input graph  $G_n = (V_n, E_n, c_n)$  such that both  $G_n$  and the algorithm output  $\text{dist}_n$  have constant width OBDDs, while the

intermediate function  $S_n^{(n)}$  generated by  $\mathcal{B}$  on  $G_n$  has only exponential OBDDs w. r. t.  $n$ .

First, we define  $G_n$ . Then, we give foundations on bounded-width functions and introduce multivariate threshold functions before the symbolic representation of  $G_n$  is discussed. We show that  $G_n$  and its corresponding shortest-paths function  $\text{dist}_n$  have bounded width—our counterexample fulfills the conditions demanded for polylog. runtime of  $\mathcal{A}$ . Finally, we show that  $S_n^{(n)}$  has exponential OBDD size for every variable order.

### 7.1 Definition of $G_n$

$G_n$  is the union of subgraphs  $G_n^{(i,j)}$  with  $2^{n-1} + 1 \leq i, j \leq 2^n - 1$  sharing only one special node  $s$ . Each subgraph  $G_n^{(i,j)}$  is a path  $(s, u_0, \dots, u_{2^n-j-1}, v_{2^n-j}, \dots, v_{2^n-1}) =: p_{i,j}$  with edge weights  $c_n(\cdot, u.) := 1$ ,  $c_n(v., \cdot) := i$ , and  $c_n(u_{2^n-j-1}, v_{2^n-j}) := i + j - 2^n$ . Moreover, there are nodes  $w_{2^n-j}, \dots, w_{2^n-1}$  connected by *shortcut edges*  $(u_{2^n-j-1}, w_{2^n-j})$ ,  $(w_{2^n-j}, v_{2^n-j})$ ,  $(v_\ell, w_{\ell+1})$ , and  $(w_{\ell+1}, v_{\ell+1})$  for  $2^n - j \leq \ell \leq 2^n - 2$  with weight 1. Note that shortcut edges bridge all edges whose weight is larger than 1. Figure 3 shows subgraph  $G_3^{(6,5)}$ .



**Fig. 3.**  $G_3^{(6,5)}$ . Edges incident to  $w$ -nodes have weight 1.

### 7.2 Bounded-Width Functions

We now introduce the class of Boolean bounded-width functions, whose convenient properties have been successfully used in the analysis of symbolic algorithms.

**Definition 11.** A  $\pi$ -OBDD for a function  $f \in B_n$  is called complete if every path from its source node to a sink has length  $n$ .

That is, complete OBDDs are not allowed to skip variable tests. The minimal-size complete  $\pi$ -OBDD for  $f \in B_n$  is also known to be canonical.

**Definition 12.** Let  $F := (f_n)_{n \in \mathbb{N}}$  be a sequence of functions  $f_n \in B_{\mathcal{N}(n)}$ ,  $\mathcal{N}: \mathbb{N} \rightarrow \mathbb{N}$ , defined on variables  $X_n := \{x_0, \dots, x_{\mathcal{N}(n)-1}\}$ . Moreover, let  $\Pi := (\pi_n)_{n \in \mathbb{N}}$  be a sequence of variable orders  $\pi_n$  on  $X_n$ .  $F$  has bounded

width  $b$  w. r. t.  $\Pi$  ( $F$  is  $b$ -bounded by  $\Pi$ ) iff for all  $n \in \mathbb{N}$  the minimal  $\pi_n$ -OBDD for  $f_n$  contains no more than  $b$  nodes labeled with the same variable  $x_i$  for  $i \in \{0, \dots, \mathcal{N}(n) - 1\}$ .

Later on, we will use the convenient property that bounded-width functions are closed under binary operations in  $B_2$ , e. g., conjunction and disjunction [23, Sect. 3]. In order to compose the characteristic function of  $G_n$  of simple expressions, we will use comparisons like  $h(x, y, z) := (|x| + |y| = |z|)$  as building blocks. These can be composed of *multivariate threshold functions*.

**Definition 13 (Woelfel [30]).** Let  $f \in B_{kn}$  be defined on variables  $x^{(1)}, \dots, x^{(k)} \in \{0, 1\}^n$ . Then,  $f$  is called  $k$ -variate threshold function iff there are  $W \in \mathbb{N}$ ,  $T \in \mathbb{Z}$ , and  $w_1, \dots, w_k \in \{-W, \dots, W\}$  such that

$$f(x^{(1)}, \dots, x^{(k)}) = \left( \sum_{i=1}^k w_i \cdot |x^{(i)}| \geq T \right) .$$

Assume w. l. o. g.  $W = \max\{|w_1|, \dots, |w_k|\}$ . Then,  $W$  is called the maximum absolute weight of  $f$ . The class of  $k$ -variate threshold functions  $f \in B_{kn}$  with maximum absolute weight  $W$  is denoted by  $\mathbb{T}_{k,n}^W$ .

Obviously, our example  $h$  can be expressed as  $(|x| + |y| - |z| \geq 0) \wedge (|z| - |x| - |y| \geq 0)$ . Analogue, the relations  $>$ ,  $\leq$ , and  $<$  can be composed of multivariate threshold functions, too.

Assume that each of the  $k$  function arguments  $x^{(1)}, \dots, x^{(k)} \in \{0, 1\}^n$  has its own variable order  $\tau_\ell$ . The global order  $\pi$  is called *interleaved* if it respects each  $\tau_\ell$  while reading variables  $x_\beta^{(\ell)}$  with same bit index  $\beta$  en bloc, that is,  $\pi := (\tau_1(0), \tau_2(0), \dots, \tau_k(0), \tau_1(1), \dots, \tau_k(n-1))$ . We say that argument  $i$  is read with *increasing bit significance* if  $\tau_\ell = (x_0^{(\ell)}, \dots, x_{n-1}^{(\ell)})$ .

**Theorem 5 (Woelfel [30]).** Let  $F := (f_n)_{n \in \mathbb{N}}$  be a sequence of functions  $f_n \in B_{k\mathcal{N}(n)}$ ,  $k \in \mathbb{N}$ ,  $\mathcal{N}: \mathbb{N} \rightarrow \mathbb{N}$ , and  $\Pi := (\pi_n)_{n \in \mathbb{N}}$  interleaved variable orders  $\pi_n$  with increasing bit significance. If for all  $n \in \mathbb{N}$  it is  $f_n \in \mathbb{T}_{k,\mathcal{N}(n)}^W$  then  $F$  is  $\mathcal{O}(k^2W)$ -bounded by  $\Pi$ .

From the closedness under binary operations, we conclude that weighted comparisons with relations  $>$ ,  $\geq$ ,  $=$ ,  $<$ , and  $\leq$  have also bounded width w. r. t. interleaved variable orders  $\Pi$  with increasing bit significance. Because  $W$  and  $k$  are independent of  $n$ , the comparisons have  $\pi_n$ -OBDD size  $\mathcal{O}(\mathcal{N}(n))$ .

### 7.3 Symbolic Representation of $G_n$

**Node encoding.** At first, we consider the node encoding of  $G_n$ . We overestimate  $|V_n|$  by  $2^{3n+1}$  and denote its nodes by  $q_0, \dots, q_{2^{3n+1}-1}$ . Every node  $q \in V_n$  that belongs to subgraph  $G_n^{(i,j)}$  gets a binary node number  $x$  consisting of 4 components:

1. The binary value  $(i)$  of length  $n$ ,
2. the binary value  $(j)$  of length  $n$ ,
3. a flag bit  $a$ , where  $a = 0$  indicates that  $q$  is a  $u$ - or  $v$ -node in  $G^{(i,j)}$  and  $a = 1$  indicated that  $q$  is a  $w$ -node,
4. the final index  $b$  of  $q$  on  $p_{i,j}$  respectively its corresponding  $w$ -part. It is a binary string of length  $n$ .

For example,  $(i)(j)00^n$  identifies node  $u_0$  of  $G_n^{(i,j)}$ . The unused singletons inherently included in this encoding do not disturb our further considerations. Finally, we define  $s$  to have node number 0.

The maximum edge weight is at most  $2^n - 1$ . Due to our node encoding,  $G_n$  has  $2^{3n+1}$  nodes and the value  $2^{4n+1} - 1$  is an upper bound for the length of any path. That is, we use  $4n + 1$  bits to encode a binary distance value  $d$ . We now represent  $G_n$  symbolically by its characteristic function  $\chi_n \in B_{10n+3}$  defined by

$$\chi_n(x, y, d) = 1 \Leftrightarrow [(q_{|x|}, q_{|y|}) \in E_n] \wedge [c_n(|x|, |y|) = |d|] \ .$$

On the other hand, the shortest-paths function  $\text{dist}_n$  is symbolically represented by  $D_n \in B_{10n+3}$  defined by

$$D_n(x, y, d) = 1 \Leftrightarrow \text{dist}_n(q_{|x|}, q_{|y|}) = |d| \ .$$

In the following, we write the arguments  $x$  and  $y$  of  $\chi_n$  and  $\text{dist}_n$  in terms of their components, e. g.,  $\chi_n(i, j, a, b, i', j', a', b', d)$  for  $x = i j a b$  and  $y = i' j' a' b'$ .

**Expressing  $\chi_n$  in terms of multivariate threshold functions.** Let  $\Pi := (\pi_n)_{n \in \mathbb{N}}$  be the sequence of variable orders  $\pi_n$  for  $G_n$  which reads  $\chi_n$ 's variable components  $(i, j, a, b, i', j', a', b')$  interleaved with increasing bit significance (see, e. g., [23]). Due to the node encoding discussed above, it is  $\mathcal{N}(n) = 10n + 3$ .

**Proposition 3.** *The sequence  $\chi := (\chi_n)_{n \in \mathbb{N}}$  has bounded width w. r. t.  $\Pi$ .*

*Proof.* We show that each  $\chi_n$  can be composed of multivariate threshold functions with constant maximum absolute weight, i. e., functions in  $\mathbb{T}_{9,10n+3}^W$  for  $W = \mathcal{O}(1)$  and the number  $k = 9$  of argument components  $\chi_n$  is defined on. Therefore, we express  $\chi_n$  as disjunction of six helping functions  $H_n^{(1)}, \dots, H_n^{(6)}$  representing different kinds of edges in  $G_n$ .

$$\chi_n(i, j, a, b, i', j', a', b', d) := \bigvee_{k=1}^6 H_n^{(k)}(i, j, a, b, i', j', a', b', d)$$

$H_n^{(1)}$  represents edges incident to node  $s$ .

$$\begin{aligned} H_n^{(1)}(i, j, a, b, i', j', a', b', d) := & (|i| = |j| = a = |b| = 0) \wedge (|d| = 1) \\ & \wedge (2^{n-1} + 1 \leq |i'|, |j'|) \wedge (a' = |b'| = 0) \end{aligned}$$

$H_n^{(2)}$  represents edges  $(u_\ell, u_{\ell+1})$  for  $0 \leq \ell \leq 2^n - j - 2$ .

$$\begin{aligned} H_n^{(2)}(i, j, a, b, i', j', a', b', d) &:= (2^{n-1} + 1 \leq |i|, |j|) \wedge (|d| = 1) \\ &\wedge (i = i') \wedge (j = j') \wedge (a = a' = 0) \wedge (|b'| = |b| + 1) \wedge (|b'| \leq 2^n - j - 1) \end{aligned}$$

$H_n^{(3)}$  represents edges  $(u_{2^n-j-1}, v_{2^n-j})$ .

$$\begin{aligned} H_n^{(3)}(i, j, a, b, i', j', a', b', d) &:= (2^{n-1} + 1 \leq |i|, |j|) \wedge (|d| = |i| + |j| - 2^n) \\ &\wedge (i = i') \wedge (j = j') \wedge (a = a' = 0) \wedge (|b| = 2^n - |j| - 1) \wedge (|b'| = 2^n - |j|) \end{aligned}$$

$H_n^{(4)}$  represents edges  $(v_\ell, v_{\ell+1})$  for  $2^n - j \leq \ell \leq 2^n - 2$ .

$$\begin{aligned} H_n^{(4)}(i, j, a, b, i', j', a', b', d) &:= (2^{n-1} + 1 \leq |i|, |j|) \wedge (|d| = |i|) \\ &\wedge (i = i') \wedge (j = j') \wedge (a = a' = 0) \wedge (|b'| = |b| + 1) \\ &\wedge (2^n - |j| \leq |b| \leq 2^n - 2) \end{aligned}$$

$H_n^{(5)}$  represents shortcut edges directed towards  $w$ -nodes.

$$\begin{aligned} H_n^{(5)}(i, j, a, b, i', j', a', b', d) &:= (2^{n-1} + 1 \leq |i|, |j|) \wedge (|d| = 1) \\ &\wedge (i = i') \wedge (j = j') \wedge (a = 0) \wedge (a' = 1) \wedge (|b'| = |b| + 1) \\ &\wedge (2^n - |j| - 1 \leq |b| \leq 2^n - 2) \end{aligned}$$

$H_n^{(6)}$  represents shortcut edges directed towards  $v$ -nodes.

$$\begin{aligned} H_n^{(6)}(i, j, a, b, i', j', a', b', d) &:= (2^{n-1} + 1 \leq |i|, |j|) \wedge (|d| = 1) \\ &\wedge (i = i') \wedge (j = j') \wedge (a = 1) \wedge (a' = 0) \wedge (|b'| = |b|) \wedge (2^n - |j| \leq |b|) \end{aligned}$$

We have defined  $\chi_n$  according to the definition of  $G_n$ . Each atomic expression of a function  $H_n^{(k)}$ ,  $k \in \{1, \dots, 6\}$ , is a comparison that can be expressed as a conjunction of a constant number of multivariate threshold functions. Each of the latter has a constant maximum absolute value for all  $n \in \mathbb{N}$ . Altogether,  $\chi_n$  is composed by binary operations preserving the bounded width property; hence,  $\chi$  has bounded width, too.  $\square$

#### 7.4 Shortest Paths in $G_n$

Shortest paths in  $G_n^{(i,j)}$  use shortcut edges to avoid visiting expensive edges of weight  $i$  or  $i + j - 2^n > 1$ . We conclude

$$\begin{aligned} \text{dist}(\phi_k, \phi_{k'}) &= \max\{2^n - j - 1 - k, 0\} - \max\{2^n - j - 1 - k', 0\} \\ &\quad + 2 \cdot \left[ \max\{k' - (2^n - j - 1), 0\} - \max\{k - (2^n - j - 1), 0\} \right] \\ &\leq 2 \cdot (k' - k) , \end{aligned}$$

$$\text{dist}(\phi_k, w_{k'}) = \text{dist}(\phi_k, \phi_{k'-1}) + 1 ,$$

$$\text{dist}(w_k, \phi_{k'}) = \text{dist}(\phi_k, \phi_{k'}) + 1 ,$$

$$\text{dist}(w_k, w_{k'}) = \text{dist}(\phi_k, \phi_{k'-1}) + 2 ,$$

and  $\text{dist}(s, q) = \text{dist}(u_0, q) + 1$  for  $\phi \in \{u, v\}$ ,  $0 \leq k \leq k' \leq 2^n - 1$ , and  $q \in V_n^{(i,j)} \setminus \{s\}$ .

**Proposition 4.** *The sequence  $D := (D_n)_{n \in \mathbb{N}}$  has bounded width w. r. t.  $\Pi$ .*

*Proof.* Analogue to the proof of Prop. 3, we express  $D_n$  in terms of simple expressions that can be realized by functions in  $\mathbb{T}_{9,10n+3}^{\mathcal{O}(1)}$ . We only construct a function  $D'_n$  for the case that start- and end-node are  $u$ - or  $v$ -nodes. The remaining cases for  $s$  and  $w$ -nodes resp. identic start- and end-nodes are straightforward. Again, we use helping functions which we call  $P_n^{(1)}, \dots, P_n^{(3)}$  this time and whose disjunction yields  $D'_n$ .

$P_n^{(1)}$  covers the case that start- and end-node are both  $u$ -nodes.

$$P_n^{(1)}(i, j, a, b, i', j', a', b', d) := (2^{n-1} + 1 \leq |i|, |j|) \wedge (i = i') \wedge (j = j') \\ \wedge (a = a' = 0) \wedge (|b| < |b'|) \wedge (|b'| \leq 2^n - |j| - 1) \wedge (|d| = |b'| - |b|)$$

$P_n^{(2)}$  covers the case that start- and end-node are both  $v$ -nodes.

$$P_n^{(2)}(i, j, a, b, i', j', a', b', d) := (2^{n-1} + 1 \leq |i|, |j|) \wedge (i = i') \wedge (j = j') \\ \wedge (a = a' = 0) \wedge (|b| < |b'|) \wedge (2^n - |j| \leq |b|) \wedge [|d| = 2 \cdot (|b'| - |b|)]$$

$P_n^{(3)}$  covers the remaining case that the start-node is a  $u$ -node and the end-node is a  $v$ -node.

$$P_n^{(3)}(i, j, a, b, i', j', a', b', d) := (2^{n-1} + 1 \leq |i|, |j|) \wedge (i = i') \wedge (j = j') \\ \wedge (a = a' = 0) \wedge (|b| \leq 2^n - |j| - 1) \wedge (2^n - |j| \leq |b'|) \\ \wedge [|d| = (2^n - |j| - 1 - |b|) + 2 \cdot (|b'| - 2^n + |j| + 1)]$$

□

## 7.5 Shortest Paths in $G_n$ Consisting of at Most $2^n$ Edges

On the other hand, the intermediate function  $S_n^{(n)}$  generated by  $\mathcal{B}$  on  $G_n$  must not cover the shortest  $s$ - $v_{2^n-1}$ -path in  $G_n^{(i,j)}$  because only  $2^n$  edges are allowed. Therefore, it has to represent the direct path  $p_{i,j}$  that uses no shortcut edges, and which has length  $(2^n - j) + (i + j - 2^n) + (j - 1)i = i \cdot j$ . The OBDD representation of the corresponding characteristic function inherently contains the symbolic graph of integer multiplication. In this way, it is possible to show an exponential lower bound on the OBDD size of  $S_n^{(n)}$  w. r. t. its number of  $\Theta(\log |V_n| + \log c_n^{\max})$  Boolean variables (see Theorem 6). This implies that  $\mathcal{B}$ , while allowing zero weights, has not the same convenient runtime properties as  $\mathcal{A}$ .

So Theorem 2 has been used to show limits of the symbolic APSP-algorithm  $\mathcal{B}$  and to justify the restriction of  $\mathcal{A}$  to strictly positive edge weights.

**Theorem 6.** *The function  $S_n^{(n)}$  has exponential  $\pi$ -OBDD size w. r. t.  $n = \Theta(\mathcal{N}(n)) = \Theta(\log |V_n| + \log c_n^{\max})$  for every variable order  $\pi$  on  $\mathcal{N}(n) = 10n + 3$  variables.*

*Proof.* Assume w. l. o. g.  $n = 3m + 1$  for  $m \in \mathbb{N}$ . Let  $P$  be a  $\pi$ -OBDD for  $S_n^{(n)}$  for some arbitrary variable order  $\pi$  on  $\mathcal{N}(n) = 10n + 3$  variables. We now use  $P$  to construct a  $\pi'$ -oblivious BP  $P'$  of size  $\mathcal{O}(m \cdot \text{size}(P))$  for  $MUL-GRAPH_m$  whose variable sequence  $\pi'$  contains each variable at most twice. Due to Theorem 2, this implies an exponential lower bound on  $\text{OBDD}(S_n^{(n)})$ .

In order to construct  $P'$ , we replace the variables for  $i, j, a,$  and  $b$  in  $P$  by zeros forcing the start-node to be  $s$ . Moreover, we replace  $a'$  by 0 and  $b'$  by  $(2^n - 1)$ . That is, the target-node is  $v_{2^n-1}$  in  $p_{i,j}$ . Finally, we replace  $i_{3m}$  and  $j_{2m}$  by 1 and all other variables  $i_k, j_k$  with  $k \geq m$  by 0. We denote the resulting function by  $f_n$ , the remaining  $m$   $i$ -variables by  $x$ , and the remaining  $m$   $j$ -variables by  $y$ .

Because shortest paths represented by  $S_n^{(n)}$  may visit at most  $2^n$  edges, no shortcut edges may be used on the  $s-v_{2^n-1}$ -path in  $G_n^{(i,j)}$ . Hence, it is  $f_n(x, y, d) = 1$  iff

$$|d| = |i| \cdot |j| = (|x| + 2^{3m}) \cdot (|y| + 2^{2m}) = |x| \cdot |y| + |x| \cdot 2^{2m} + |y| \cdot 2^{3m} + 2^{5m}$$

and  $|x|, |y| > 0$ . Then, the  $2m$  variables  $d_{2m-1}, \dots, d_0$  contain the multiplication result  $(|x| \cdot |y|) =: r$ . Moreover,  $d_{3m-1}, \dots, d_{2m}$  corresponds to  $x$  and  $d_{4m-1}, \dots, d_{3m}$  corresponds to  $y$ . Besides  $d_{5m} = 1$ , all other  $d$ -variables are 0.

We force  $d$ -bits not belonging to  $r$  to fit this bit scheme for satisfying inputs: We replace  $d_{5m}$  by 1 and relabel BP nodes labeled by  $d_{2m+k}$  by  $x_k$  resp.  $d_{3m+k}$  by  $y_k$  for  $0 \leq k \leq m - 1$ . All other  $d$ -variables not belonging to  $r$  are replaced by 0.

The remaining  $2m$   $d$ -variables are denoted by  $z$ . We obtained a restricted version  $MUL-GRAPH_m^*$  of  $MUL-GRAPH_m$  with

$$MUL-GRAPH_m^*(x, y, z) = 1 \Leftrightarrow (|x| \cdot |y| = |z|) \wedge (|x|, |y| > 0)$$

which can be easily extended to  $MUL-GRAPH_m$  by

$$MUL-GRAPH_m(x, y, z) = MUL-GRAPH_m^* \vee [((|x| = 0) \vee (|y| = 0)) \wedge (|d| = 0)] \quad (4)$$

Let  $P'$  be the resulting BP. Due to our node relabeling, each variable is read at most twice. Replacing variables by Boolean constants does not enlarge BPs. Comparisons with 0 are of bounded-width and have OBDD size  $\mathcal{O}(m)$  for any variable order. Hence, the disjunction with  $MUL-GRAPH_m^*$  in (4) causes the BP size to grow at most by a factor of  $\mathcal{O}(m)$  (see, e. g., [27, Sect. 3.3]), and  $P'$  has size  $\mathcal{O}(m \cdot \text{size}(P))$ .  $\square$

## 8 Conclusions and Open Problems

Exponential lower bounds on the OBDD size of the fundamental and hard functions MUL, SQU, ISA, and HWB carry over to the symbolic graph scenario.

This has not to be the case in general as seen for the counterexample function vector FSA. Graphs of such simply-structured functions can then be used to show limits of symbolic graph algorithms, which has been done exemplarily for a symbolic all-pairs shortest-paths algorithm.

Corollary 2 covers only a restricted version of the symbolic graph of squaring. The OBDD size of its unrestricted graph is an open question yet; experiments suggest that it is also exponential. Moreover, it is of interest whether exponential lower bounds for integer multiplication w. r. t. more general types of branching programs carry over to the symbolic graph scenario, too (see, e. g., [1, 19, 29]).

**Acknowledgments.** Thanks to André Gronemeier, Martin Sauerhoff, Detlef Sieling, and Ingo Wegener for proofreading and helpful discussions.

## References

- [1] F.M. Ablayev and M. Karpinski. A lower bound for integer multiplication on randomized ordered read-once branching programs. *Information and Computation*, 186(1):78–89, 2003.
- [2] N. Alon and W. Maass. Meanders and their applications in lower bound arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.
- [3] R. Bloem, H.N. Gabow, and F. Somenzi. An algorithm for strongly connected component analysis in  $n \log n$  symbolic steps. In *FMCAD'00*, volume 1954 of *LNCS*, pages 37–54. Springer, 2000.
- [4] A. Borodin and S. Cook. A time–space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11:287–297, 1982.
- [5] Y. Breitbart, H.B. Hunt III, and D. Rosenkrantz. On the size of binary decision diagrams representing Boolean functions. *Theoretical Computer Science*, 145:45–69, 1995.
- [6] J. Feigenbaum, S. Kannan, M.Y. Vardi, and M. Viswanathan. Complexity of problems on graphs represented as OBDDs. *Chicago Journal of Theoretical Computer Science*, 1999:1–25, 1999.
- [7] R. Gentilini, C. Piazza, and A. Policriti. Computing strongly connected components in a linear number of symbolic steps. In *SODA'03*, pages 573–582. ACM Press, 2003.
- [8] R. Gentilini and A. Policriti. Biconnectivity on symbolically represented graphs: A linear solution. In *ISAAC'03*, volume 2906 of *LNCS*, pages 554–564. Springer, 2003.
- [9] J. Gergov. Time-space tradeoffs for integer multiplication on various types of input oblivious sequential machines. *Information Processing Letters*, 51:265–269, 1994.
- [10] G.D. Hachtel and F. Somenzi. *Logic Synthesis and Verification Algorithms*. Kluwer Academic Publishers, Boston, 1996.
- [11] G.D. Hachtel and F. Somenzi. A symbolic algorithm for maximum flow in 0–1 networks. *Formal Methods in System Design*, 10:207–219, 1997.
- [12] R. Hojati, H. Touati, R.P. Kurshan, and R.K. Brayton. Efficient  $\omega$ -regular language containment. In *CAV'93*, volume 663 of *LNCS*, pages 396–409. Springer, 1993.

- [13] J. Hromkovič. *Communication Complexity and Parallel Computing*. Springer, Berlin Heidelberg New-York, 1997.
- [14] H. Jin, A. Kuehlmann, and F. Somenzi. Fine-grain conjunction scheduling for symbolic reachability analysis. In *TACAS'02*, volume 2280 of *LNCS*, pages 312–326. Springer, 2002.
- [15] S. Jukna. The graph of integer multiplication is hard for read- $k$ -times networks. Technical Report 95–10, Universität Trier, 1995.
- [16] M. Keim, R. Drechsler, B. Becker, M. Martin, and P. Molitor. Polynomial formal verification of multipliers. *Formal Methods in System Design*, 22:39–58, 2003.
- [17] K.L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, Boston, 1994.
- [18] I. Moon, J.H. Kukula, K. Ravi, and F. Somenzi. To split or to conjoin: The question in image computation. In *DAC'00*, pages 23–28. ACM Press, 2000.
- [19] S. Ponzio. A lower bound for integer multiplication with read-once branching programs. *SIAM Journal on Computing*, 28(3):798–815, 1998.
- [20] K. Ravi, R. Bloem, and F. Somenzi. A comparative study of symbolic algorithms for the computation of fair cycles. In *FMCAD'00*, volume 1954 of *LNCS*, pages 143–160. Springer, 2000.
- [21] D. Sawitzki. Experimental studies of symbolic shortest-path algorithms. In *WEA'04*, volume 3059 of *LNCS*, pages 482–497. Springer, 2004.
- [22] D. Sawitzki. Implicit flow maximization by iterative squaring. In *SOFSEM'04*, volume 2932 of *LNCS*, pages 301–313. Springer, 2004.
- [23] D. Sawitzki. On graphs with characteristic bounded-width functions. Technical report, Universität Dortmund, 2004. Available via <http://ls2-www.cs.uni-dortmund.de/~sawitzki/OGwCBWF.pdf>.
- [24] D. Sawitzki. A symbolic approach to the all-pairs shortest-paths problem. To appear in *WG'04*, 2004.
- [25] D. Sawitzki. Lower bounds on the OBDD size of graphs of some popular functions. In *SOFSEM'05*, volume 3381 of *LNCS*, pages 298–309. Springer, 2005.
- [26] I. Wegener. Optimal lower bounds on the depth of polynomial-size threshold circuits for some arithmetic functions. *Information Processing Letters*, 46:85–87, 1993.
- [27] I. Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM, Philadelphia, 2000.
- [28] P. Woelfel. New bounds on the OBDD-size of integer multiplication via universal hashing. In *STACS'01*, volume 2010 of *LNCS*, pages 563–574. Springer, 2001.
- [29] P. Woelfel. On the complexity of integer multiplication in branching programs with multiple tests and in read-once branching programs with limited nondeterminism. In *CCC'02*, pages 80–89. IEEE Press, 2002.
- [30] P. Woelfel. Symbolic topological sorting with OBDDs. In *MFCS'03*, volume 2747 of *LNCS*, pages 671–680. Springer, 2003.
- [31] A. Xie and P.A. Beerel. Implicit enumeration of strongly connected components. In *ICCAD'99*, pages 37–40. ACM Press, 1999.