Symbolic OBDD-Based Reachability Analysis Needs Exponential Space

Beate Bollig

LS2 Informatik, TU Dortmund
44221 Dortmund, Germany

Abstract. Ordered binary decision diagrams (OBDDs) are one of the most common dynamic data structures for Boolean functions. Nevertheless, many basic graph problems are known to be PSPACE-hard if their input graphs are represented by OBDDs. Despite the hardness results there are not many concrete nontrivial lower bounds known for the complexity of problems on OBDD-represented graph instances. Computing the set of vertices that are reachable from some predefined vertex $s \in V$ in a directed graph $G = (V, E)$ is an important problem in computer-aided design, hardware verification, and model checking. Until now only exponential lower bounds on the space complexity of a restricted class of OBDD-based algorithms for the reachability problem have been known. Here, the result is extended by presenting an exponential lower bound on the space complexity of an arbitrary OBDD-based algorithm for the reachability problem. As a by-product a general exponential lower bound is obtained for the computation of OBDDs representing all connected node pairs in a graph, the transitive closure.

Keywords: Computational complexity, lower bounds, ordered binary decision diagrams, reachability analysis, transitive closure.

1 Introduction

1.1 Motivation

When working with Boolean functions as in circuit verification, synthesis, and model checking, ordered binary decision diagrams, denoted OBDDs, introduced by Bryant [5], are one of the most often used data structures supporting all fundamental operations on Boolean functions.

Some modern applications require huge graphs so that explicit representations by adjacency matrices or adjacency lists are not any longer possible, e.g., in hardware verification and in the process of synthesis of sequential circuits state transition graphs that consist of 10^{27} vertices and 10^{36} edges are not uncommon. Since time and space do not suffice to consider individual vertices, one way out seems to be to deal with sets of vertices and edges represented by their characteristic functions. Since OBDDs are well suited for the representation and manipulation of Boolean functions, in the last years a research branch has emerged which is concerned with the theoretical design and analysis of so-called symbolic algorithms for classical graph problems on OBDD-represented graph instances (see, e.g., [11,12], [17,18], and [23]). Symbolic algorithms have to solve problems on a given graph instance by efficient functional operations offered by the OBDD data structure. At the beginning the OBDD-based algorithms have been justified by analyzing the number of executed OBDD operations (see, e.g., [11,12]). Since the number of OBDD operations is not directly proportional to the running time of an algorithm, as the running time for one OBDD operation depends on the sizes of the OBDDs on which the operations are performed, newer research tries to analyze the overall running time of symbolic methods including the analysis of all OBDD sizes occurring during such an algorithm (see, e.g., [23]).

In reachability analysis the task is to compute the set of states of a state-transition system that are reachable from a set of initial states. Besides explicit methods for traversing states one by one and SAT-based techniques for deciding distance-bounded reachability between pairs of state sets, symbolic methods are one of the most commonly used approaches to this problem (see, e.g., [7,8]). In the OBDD-based setting our aim is to compute a representation for the characteristic function $\mathcal{X}_R$ of the solution set $R \subseteq V$. I.e., the input consists of an OBDD representing the characteristic function of the edge set of a graph $G = (V, E)$ and a predefined vertex $s \in V$ and the output is an OBDD representing the characteristic function of the vertex set R which contains all vertices reachable from the vertex s via a directed path in G. BFS-like approaches using $O(|V|)$ OBDD operations [13] and iterative squaring methods using $O(\log^2 |V|)$ operations [17] are known. In [20] Sawitzki has proved that algorithms that solve the reachability problem by computing intermediate sets of vertices reachable from the vertex s via directed paths of length at most 2^p, $p \in \{1, \ldots, \lfloor \log |V| \rfloor\}$, need exponential space if the variable ordering is not changed during the algorithms. For this result he has proved the first exponential lower bound on the size of OBDDs representing the most significant bit of integer multiplication for a predefined variable ordering. Afterwards, he has defined inputs for the reachability problem such that during the computation of the investigated restricted class of algorithms representations for the most significant bit of integer multiplication are necessary. In [4] his result has been improved by presenting a larger lower bound on the OBDD size of the most significant bit for the variable ordering considered in [20]. Lower bounds on the size of OBDDs for a predefined variable ordering do not rule out the possibility that there are other variable orderings leading to OBDDs of small size. Since Sawitzki's assumption that the variable ordering is not changed during the computation is not realistic because in application reordering heuristics are used in order to minimize the OBDD size for intermediate OBDD results, in [2,3] the result has been improved by presenting general exponential lower bounds on the OBDD size of the most significant bit of integer multiplication. Here, we generalize Sawitzki's result and show that the reachability problem for graphs represented by OBDDs needs exponential space for all possible OBDD-based algorithms.

1.2 Results and Organization of the Paper

Our main result can be summarized as follows.

Theorem 1. *The reachability problem on* OBDD*-represented graphs needs exponential space.*

In Section 2 we define some notation and present some basics concerning OBDDs. Section 3 contains the results of the paper. We prove that OBDD-based reachability analysis needs exponential space. As a by-product a general exponential lower bound is obtained for the computation of OBDDs representing all connected node pairs in a graph, the transitive closure.

2 Preliminaries

2.1 Ordered Binary Decision Diagrams

Boolean circuits, formulae, and binary decision diagrams (BDDs), sometimes called branching programs, are standard representations for Boolean functions. (For a history of results on binary decision diagrams see, e.g., the monograph of Wegener [22]). Besides the complexity theoretical viewpoint people have used restricted binary decision diagrams in applications. Bryant [5] has introduced ordered binary decision diagrams (OBDDs) which have become one of the most popular data structures for Boolean functions. Among the many areas of application are verification, model checking, computer-aided design, relational algebra, and symbolic graph algorithms.

Definition 1. *Let $X_n = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. A variable ordering π on X_n is a permutation on $\{1, \ldots, n\}$ leading to the ordered list $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the variables.*

In the following a variable ordering π is sometimes identified with the corresponding ordering $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the variables if the meaning is clear from the context.

Definition 2. *A π-*OBDD *on X_n is a directed acyclic graph $G = (V, E)$ whose sinks are labeled by Boolean constants and whose non-sink (or inner) nodes are labeled by Boolean variables from X_n. Each inner node has two outgoing edges one labeled by 0 and the other by 1. The edges between inner nodes have to respect the variable ordering π, i.e., if an edge leads from an x_i-node to an x_j-node, then $\pi^{-1}(i) \le \pi^{-1}(j)$ (x_i precedes x_j in $x_{\pi(1)}, \ldots, x_{\pi(n)}$). Each node v represents a Boolean function $f_v \in B_n$, i.e., $f_v : \{0,1\}^n \to \{0,1\}$, defined in the following way. In order to evaluate $f_v(b)$, $b \in \{0,1\}^n$, start at v. After reaching an x_i-node choose the outgoing edge with label b_i until a sink is reached. The label of this sink defines $f_v(b)$. The width of an* OBDD *is the maximum number of nodes labeled by the same variable. The size of a π-*OBDD *G is equal to the number of its nodes and the π-*OBDD *size of a function f, denoted by π-*OBDD*(f), is the size of the minimal π-*OBDD *representing f.*

The size of the reduced π-OBDD representing f is described by the following structure theorem [21].

Theorem 2. *The number of $x_{\pi(i)}$-nodes of the π-OBDD for f is the number s_i of different subfunctions $f_{|x_{\pi(1)}=a_1,\ldots,x_{\pi(i-1)}=a_{i-1}}$, $a_1,\ldots,a_{i-1} \in \{0,1\}$, that essentially depend on $x_{\pi(i)}$ (a function g depends essentially on a Boolean variable z if $g_{|z=0} \neq g_{|z=1}$).*

Theorem 2 implies the following simple observation which is helpful in order to prove lower bounds. Given an arbitrary variable ordering π the number of nodes labeled by a variable x in the reduced π-OBDD representing a given function f is not smaller than the number of x-nodes in a reduced π-OBDD representing any subfunction of f.

It is well known that the size of an OBDD representing a function f, that is defined on n Boolean variables and depends essentially on all of them, depends on the chosen variable ordering and may vary between linear and exponential size. Since in applications the variable ordering π is not given in advance we have the freedom (and the problem) to choose a good ordering for the representation of f.

Definition 3. *The* OBDD *size or* OBDD *complexity of f is the minimum of all π-OBDD(f).*

2.2 Graph Representations by OBDDs

In the following for $z = (z_{n-1},\ldots,z_0) \in \{0,1\}^n$ let $|z| := \sum_{i=0}^{n-1} z_i 2^i$. Let $G = (V,E)$ be a graph with N vertices $v_0,\ldots v_{N-1}$. The edge set E can be represented by an OBDD for its characteristic function, where $\mathcal{X}_E(x,y) = 1 \Leftrightarrow (|x|,|y| < N) \wedge (v_{|x|}, v_{|y|}) \in E$, $x,y \in \{0,1\}^n$ and $n = \lceil \log N \rceil$. Undirected edges are represented by symmetric directed ones. In the rest of the paper we assume that N is a power of 2 since it has no bearing on the essence of our results.

Figure 1 shows an example of a directed graph $G = (V,E)$, where $|V| = N = 2^3$, and the OBDD representation for the characteristic function of its edge set E (with respect to the variable ordering $x_0, y_0, x_1, y_1, x_2, y_2$). (Note, that the represented graph G is only a toy example so that the difference in the representation size is only small.)

3 OBDD-Based Reachability Analysis Needs Exponential Space

In this section we prove Theorem 1 and show that OBDD-based reachability analysis needs exponential space. The proof structure is the following one. First, we define a sequence G_n of pathological graph instances. G_n is an input for the reachability problem. We show that the size of the corresponding OBDD representation for the characteristic function of its edge set is polynomial with respect to the number of Boolean variables. Furthermore, we choose the vertex s which

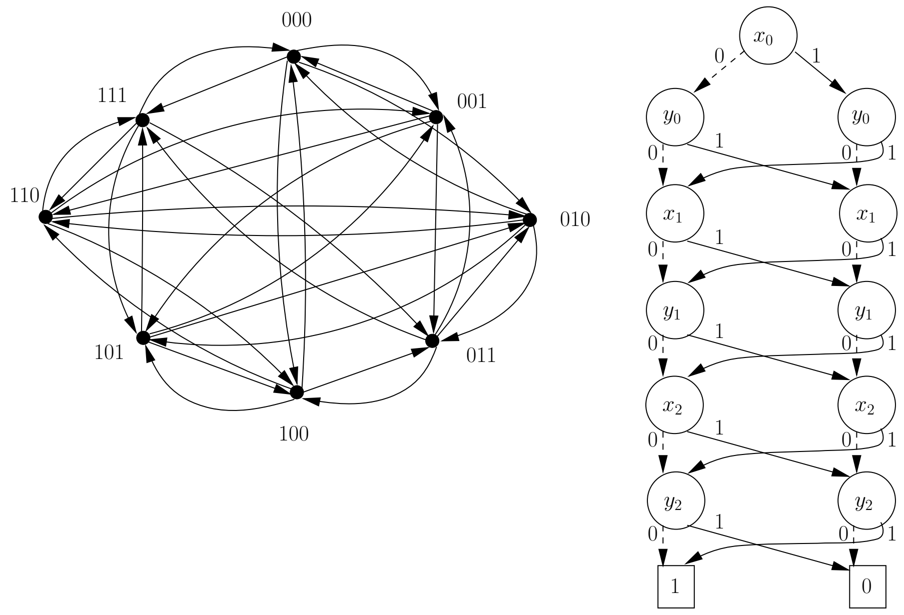

Fig. 1. A directed graph $G = (V, E)$ together with an encoding of its vertices and the corresponding OBDD representation of its edge set E

is also a part of the input in a clever way so that the characteristic function of the vertex set R, that consists of all vertices reachable from the vertex s via directed paths in G_n, is equal to a Boolean function called permutation test function PERM_n. The OBDD complexity of PERM_n is known to be exponential [14,15]. Therefore, every OBDD-based algorithm that solves the reachability problem needs exponential space with respect to its input size. Note, that this result is independent of the chosen variable ordering for the output OBDD. Now, we make our ideas more precise.

1. The definition of the input graph G_n:
   The graph G_n consists of 2^{n^2} vertices $v_{i_1,\ldots,i_n}$, $i_j \in \{0, \ldots, 2^n - 1\}$ and $j \in \{1, \ldots, n\}$. All vertices $v_{i_1,\ldots,i_n}$ for which there exists an index i_j where i_j is not a power of 2, $j \in \{1, \ldots, n\}$, are isolated vertices with no incoming or outgoing edges. A vertex $v_{i_1,\ldots,i_n}$, where i_j is a power of 2 for all $j \in \{1, \ldots, n\}$, has $n - 1$ directed outgoing edges. There exists an edge $(v_{i_1,\ldots,i_n}, v_{l_1,\ldots,l_n})$ in E iff there exists $k \in \{1, \ldots n - 1\}$ such that $i_1 = l_1, \ldots, i_{k-1} = l_{k-1}, i_k = l_{k+1}, i_{k+1} = l_k, i_{k+2} = l_{k+2}, \ldots, i_n = l_n$. The vertex s is equal to $v_{i_1,\ldots,i_n}$, where $i_1, \ldots, i_n$ are different powers of 2, i.e., $i_j := 2^{n-j}$.

   The graph G_n can be described in the following way. If we write the binary representation of the indices i_j, $1 \le j \le n$, in a rowwise manner one below the other such that the indices are represented by a Boolean matrix of dimension $n \times n$, the vertices that correspond to matrices where there is not exactly one 1-entry in each row are isolated vertices. For the other vertices there exists a directed edge from one vertex to another iff the binary matrix representing the indices of one vertex can be obtained by exchanging two neighbored rows in the matrix corresponding to the other vertex. The vertex s corresponds to a matrix where there exists exactly one 1-entry in each row and in each column. Such a matrix can be seen as an encoding of a permutation π in S_n. For our construction this property is sufficient for the definition of the distinguished vertex s. For a unique definition we define s is such a way that the binary representation of s corresponds to the permutation $1\ 2\ \dots n$.

2. The polynomial upper bound on the size of the OBDD representation for the characteristic function of the edge set of G_n:

   The characteristic function $\mathcal{X}_E$ of the edge set depends on $2n^2$ Boolean variables. Our aim is to prove that $\mathcal{X}_E$ can be represented by OBDDs of size $O(n^4)$ according to the variable ordering

   $$x_{11}, y_{11}, x_{12}, y_{12}, \dots, x_{nn}, y_{nn},$$

   where $x_{j1}, \dots, x_{jn}$ is the Boolean representation of the index i_j and x_{jn} the least significant bit.

   Theorem 2 tells us that it is sufficient to prove that there are only $O(n^2)$ different subfunctions obtained by replacements of the first i variables with respect to the considered variable ordering, $i \in \{0, \dots, 2n^2 - 1\}$. Then we can conclude that the OBDD size is at most $O(n^4)$ since there are only $2n^2$ variables altogether. Different subfunctions represent in a certian sence different information about partial assignments to some of the variables. Let $x^\ell = (x_{\ell 1}, \dots, x_{\ell n})$, $\ell \in \{1, \dots, n\}$, and y^ℓ analogously defined. The OBDD for $\mathcal{X}_E$ checks whether $x^\ell = y^\ell$ and $x_{\ell 1} + \dots + x_{\ell n} = 1$. This can be done by a part of the OBDD of width 3. (Figure 2 shows an OBDD with respect to the variable ordering $x_{\ell 1}, y_{\ell 1}, \dots, x_{\ell n}, y_{\ell n}$ which represents the function $x^\ell = y^\ell$.)

   If $x^j = y^j$ and $x_{j1} + \dots + x_{jn} = 1$ for $j < \ell$, $x^\ell \neq y^\ell$ but $x_{\ell 1} + \dots + x_{\ell n} = 1$ and $y_{\ell 1} + \dots + y_{\ell n} = 1$, the values for x^ℓ and y^ℓ are stored, afterwards it is checked whether $x^{\ell+1} = y^\ell$, $x^\ell = y^{\ell+1}$, $x^i = y^i$, and $x_{i1} + \dots + x_{in} = 1$ for $i > \ell + 1$. The values are stored in the sense that partial assignments corresponding to different values for x^ℓ and y^ℓ lead to different nodes in the OBDD for $\mathcal{X}_E$. (The reason is that in this case different values for x^ℓ and y^ℓ correspond to different subfunctions.) If all requirements are fulfilled, the function value of $\mathcal{X}_E$ is 1, otherwise it is 0. Since x^ℓ and y^ℓ are binary representations of powers of 2, there are only $\binom{n}{2}$ possibilities for different values for x^ℓ and y^ℓ. (If one of them is not a power of 2 the function value is 0 and we do not have to store anything.) Therefore, the different values for x^ℓ and y^ℓ can be stored by a part of the OBDD with width $\binom{n}{2}$. Altogether the width of the OBDD for $\mathcal{X}_E$ is bounded above by $O(n^2)$. (Figure 3 shows the first part of an OBDD for $\mathcal{X}_E$ with respect to the considered variable ordering and $n = 3$.)

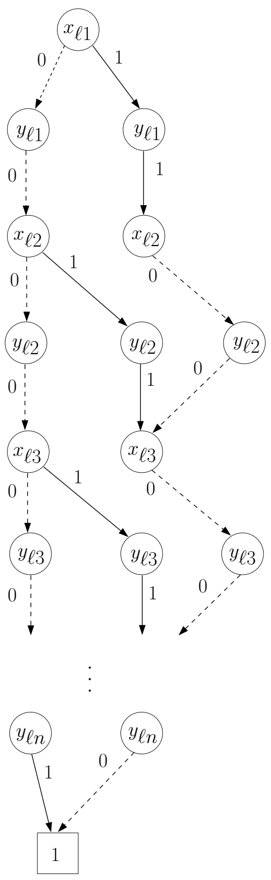

Fig. 2. An OBDD that checks whether $x^\ell = y^\ell$ and $x_{\ell 1} + \cdots + x_{\ell n} = 1$. Missing edges are leading to the 0-sink.

Since there are altogether $2n^2$ Boolean variables, the OBDD for the function $\mathcal{X}_E$ has size at most $O(n^4)$.

3. The exponential lower bound on the size of the OBDD representation for the characteristic function of the vertex set R:

   It remains to show that the output $\mathcal{X}_R$ has exponential OBDD complexity. In [14,15] exponential lower bounds of $\Omega(n^{-1/2}2^n)$ on the size of so-called nondeterministic read-once branching programs representing the function PERM$_n$, the test whether a Boolean matrix contains exactly one 1-entry in each row and in each column, have been presented. It is not difficult to see that the characteristic function of the set R of reachable vertices from s in G_n is equal to the function PERM$_n$. The reasoning is the following. Since the number of 1-entries in a column of a Boolean matrix does not change if rows

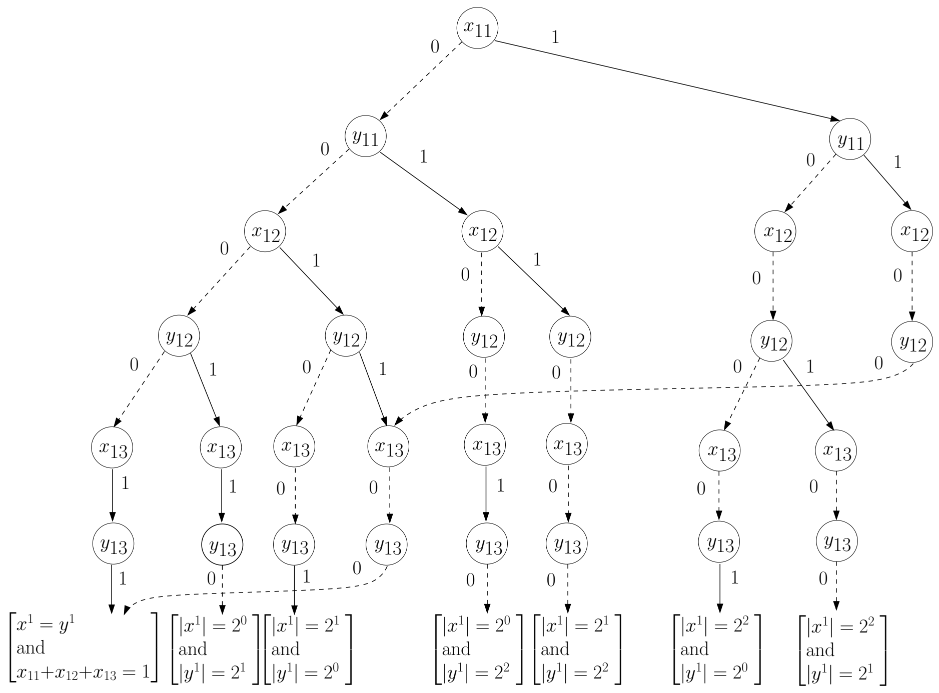
x_{11}
y_{11}
x_{12}
y_{12}
x_{13}
y_{13}
0
1
$x^1 = y^1$ and $x_{11}+x_{12}+x_{13} = 1$
$|x^1| = 2^0$ and $|y^1| = 2^1$
$|x^1| = 2^1$ and $|y^1| = 2^0$
$|x^1| = 2^0$ and $|y^1| = 2^2$
$|x^1| = 2^1$ and $|y^1| = 2^2$
$|x^1| = 2^2$ and $|y^1| = 2^0$
$|x^1| = 2^2$ and $|y^1| = 2^1$

Fig. 3. The first part of an OBDD for the characteristic function of the edge set $\mathcal{X}_E$, where $n = 3$ and the variable ordering $x_{11}, y_{11}, x_{12}, y_{12}, x_{13}, y_{13} \ldots$. Missing edges are leading to the 0-sink. [. . .] contains the necessary and sufficient information about the partial assignments to the variables $x_{11}, \ldots, y_{13}$. Different information lead to different subfunctions.

are exchanged, in G_n there are only directed paths from s to vertices whose binary representations correspond to Boolean matrices with exactly one 1-entry in each row and in each column, i.e., the binary encodings correspond to permutations in S_n. Moreover, each permutation π in S_n can be obtained from 1 2 . . . n by using only swaps between two neighbored integers. As a result we can conclude that there exists a directed path from s to a vertex $v_{i_1,\ldots,i_n}$ iff the binary representation of $i_1, \ldots, i_n$ corresponds to a Boolean matrix with exactly one 1-entry in each row and in each column. Summarizing $\mathcal{X}_R$ is equal to PERM$_n$. Since read-once branching programs are even a more general model than OBDDs, we obtain the desired exponential lower bound on the size of our output OBDD.

By simple variable replacements the reachability problem can be reduced to the computation of an OBDD for all connected node pairs, the so-called transitive closure. The problem is an important submodule in many OBDD-based graph algorithms (see, e.g., [13,17,23]).

Corollary 1. *Let transitive closure be the problem to compute an* OBDD *representing all connected node pairs for a graph symbolically represented by an* OBDD*. The problem transitive closure is not computable in polynomial space.*

If we replace the x-variables by the binary representation of the vertex s in an OBDD for the characteristic function of the transitive closure of G_n, we obtain an OBDD for PERM_n. As we have mentioned before, Theorem 2 implies that the OBDD size for a subfunction of a Boolean function f cannot be larger than the OBDD size of f. Therefore, we are done.

Sawitzki [19] has commented that it is easy to show that the computation of (strong) connected components in a (directed) graph is a problem which is not computable in polynomial space for OBDD-based algorithms by looking at graphs without any edges since then the number of (strong) connected components is exponential with respect to the number of Boolean variables. Nevertheless, it would be nice to have an example for a graph where the OBDD-representation is small but for which at least one connected component has exponential OBDD-size. Obviously, our graphs fulfills the required properties. Let a connected component be non-trivial if it contains more than a single vertex. Our graph G_n has 2^{n-1} non-trivial connected components.

4 Concluding Remarks

Representing graphs with regularities by means of data structures smaller than adjacency matrices or adjacency lists seems to be a natural idea. But problems typically get harder when their input is represented implicitly. For circuit representations this has been shown in [1,10,16]. These results do not directly carry over to problems on OBDD-represented inputs since there are Boolean functions like some output bits of integer multiplication whose OBDD complexity is exponentially larger than its circuit size [6]. In [9] it has been shown that even the very basic problem of deciding whether two vertices s and t are connected in a directed graph G, the so-called graph accessibility problem GAP, is PSPACE-complete on OBDD-represented graphs. Despite the hardness results there are not many nontrivial lower bounds known for the complexity of problems on OBDD-represented graph instances. The challenge is to prove small upper bounds on the OBDD size of input graphs and simultaneously large lower bounds on the size of OBDDs occuring during the computation. In [20] exponential lower bounds on OBDD-based algorithms for the single-source shortest paths problem, the maximum flow problem, and a restricted class of algorithms for the reachability problem have been presented. We have extended these results by presenting a concrete exponential lower bound on the space complexity of general OBDD-based algorithms for the reachability problem and the transitive closure, where the input and the output OBDD can be ordered according to different variable orderings.

Moreover, since the exponential lower bound on the representation size for the permutation test function PERM_n has been shown for so-called nondeterministic read-once branching programs [14,15], our results do also carry over to a more general model than OBDDs.

Acknowledgment. The author would like to thank the anonymous referees for their helpful comments.

References

1. Balcázar, J.L., Lozano, A.: The Complexity of Graph Problems for Succinctly Represented Graphs. In: Nagl, M. (ed.) WG 1989. LNCS, vol. 411, pp. 277–285. Springer, Heidelberg (1990)
2. Bollig, B.: On the OBDD Complexity of the Most Significant Bit of Integer Multiplication. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 306–317. Springer, Heidelberg (2008)
3. Bollig, B.: Larger Lower Bounds on the OBDD Complexity of Integer Multiplication. In: Dediu, A.H., Ionescu, A.M., Martín-Vide, C. (eds.) LATA 2009. LNCS, vol. 5457, pp. 212–223. Springer, Heidelberg (2009)
4. Bollig, B., Klump, J.: New Results on the Most Significant Bit of Integer Multiplication. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 883–894. Springer, Heidelberg (2008)
5. Bryant, R.E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. on Computers 35, 677–691 (1986)
6. Bryant, R.E.: On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication. IEEE Trans. on Computers 40, 205–213 (1991)
7. Burch, J.R., Clarke, E.M., Long, D.E., Mc Millan, K.L., Dill, D.L., Hwang, L.J.: Symbolic Model Checking: 10^{20} States and Beyond. In: Proc. of Symposium on Logic in Computer Science, pp. 428–439 (1990)
8. Coudert, O., Berthet, C., Madre, J.C.: Verification of Synchronous Sequential Machines Based on Symbolic Execution. In: Sifakis, J. (ed.) CAV 1989. LNCS, vol. 407, pp. 365–373. Springer, Heidelberg (1990)
9. Feigenbaum, J., Kannan, S., Vardi, M.V., Viswanathan, M.: Complexity of Problems on Graphs Represented as OBDDs. In: Meinel, C., Morvan, M. (eds.) STACS 1998. LNCS, vol. 1373, pp. 216–226. Springer, Heidelberg (1998)
10. Galperin, H., Wigderson, A.: Succinct Representations of Graphs. Information and Control 56, 183–198 (1983)
11. Gentilini, R., Piazza, C., Policriti, A.: Computing Strongly Connected Components in a Linear Number of Symbolic Steps. In: Proc. of SODA, pp. 573–582. ACM Press, New York (2003)
12. Gentilini, R., Piazza, C., Policriti, A.: Symbolic Graphs: Linear Solutions to Connectivity Related Problems. Algorithmica 50, 120–158 (2008)
13. Hachtel, G.D., Somenzi, F.: A Symbolic Algorithm for Maximum Flow in $0-1$ Networks. Formal Methods in System Design 10, 207–219 (1997)
14. Jukna, S.: The Effect of Null-Chains on the Complexity of Contact Schemes. In: Csirik, J.A., Demetrovics, J., Gecseg, F. (eds.) FCT 1989. LNCS, vol. 380, pp. 246–256. Springer, Heidelberg (1989)
15. Krause, M., Meinel, C., Waack, S.: Separating the Eraser Turing Machine Classes L_e, NL_e, co-NL_e and P_e. Theoretical Computer Science 86, 267–275 (1991)

16. Papadimitriou, C.H., Yannakakis, M.: A Note on Succinct Representations of Graphs. Information and Control 71, 181–185 (1986)
17. Sawitzki, D.: Implicit Flow Maximization by Iterative Squaring. In: Van Emde Boas, P., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2004. LNCS, vol. 2932, pp. 301–313. Springer, Heidelberg (2004)
18. Sawitzki, D.: Lower Bounds on the OBDD Size of Graphs of Some Popular Functions. In: Vojtáš, P., Bieliková, M., Charron-Bost, B., Sýkora, O. (eds.) SOFSEM 2005. LNCS, vol. 3381, pp. 298–309. Springer, Heidelberg (2005)
19. Sawitzki, D.: Algorithmik und Komplexität OBDD-repräsentierter Graphen. PhD Thesis, University of Dortmund, in German (2006)
20. Sawitzki, D.: Exponential Lower Bounds on the Space Complexity of OBDD-Based Graph Algorithms. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 781–792. Springer, Heidelberg (2006)
21. Sieling, D., Wegener, I.: NC-Algorithms for Operations on Binary Decision Diagrams. Parallel Processing Letters 48, 139–144 (1993)
22. Wegener, I.: Branching Programs and Binary Decision Diagrams – Theory and Applications. SIAM Monographs on Discrete Mathematics and Applications (2000)
23. Woelfel, P.: Symbolic Topological Sorting with OBDDs. Journal of Discrete Algorithms 4(1), 51–71 (2006)