COMPLEXITY THEORETICAL RESULTS ON PARTITIONED (NONDETERMINISTIC) BINARY DECISION DIAGRAMS

Beate Bollig⋆ and Ingo Wegener⋆
bollig/wegener@ls2.informatik.uni-dortmund.de

FB Informatik, LS2, Univ. Dortmund, 44221 Dortmund, Germany

Abstract. Ordered binary decision diagrams (OBDDs) have found a lot of applications in the verification of combinational and sequential circuits, protocols, and the synthesis and analysis of systems. The applications are limited, since the expressive power of polynomial-size OBDDs is too restricted. Therefore, several more general BDD models are used. Partitioned OBDDs are an OBDD model allowing a restricted use of nondeterminism and different variable orderings. They are restricted enough such that the essential operations can be performed efficiently and they allow polynomial-size representations for many more functions than OBDDs. Here the expressive power of polynomial-size partitioned OBDDs is investigated. A tight hierarchy with respect to the number of parts in the partition is proved and partitioned OBDDs are compared with other BDD models.

1 INTRODUCTION

If one would like to compute Boolean functions, one is interested in representations of small size allowing a fast evaluation. In this scenario circuits and their variants like depth-restricted circuits are the best model. In other situations, e.g., the formal verification where we like to prove that a design meets the specification, we are interested in representations fulfilling more requirements. Obviously small size for many Boolean functions is still an essential issue. In order to transform a formal specification (perhaps a circuit) into the representation it is necessary to have efficient algorithms to combine Boolean functions in the given representation by Boolean operators (like conjunction etc.). Moreover, one needs efficient algorithms to check whether two representations describe the same function and to check whether the described function is satisfiable.

Since the last problems are coNP-hard resp. NP-hard for circuits and for general branching programs, a lot of restricted types of branching programs (or binary decision diagrams which only is another notion for branching programs) have been investigated. In order to describe our results and the organization of our paper we first define the different models of binary decision diagrams which we discuss in this paper.

⋆ Supported in part by DFG grant We 1066/7.

Definition 1. A binary decision diagram (BDD) or branching program is a directed acyclic graph with one source. Each sink is labeled by a Boolean constant and each other node by a Boolean variable from $\{x_1, \ldots, x_n\}$. These nodes have two outgoing edges one labeled by 0 and the other by 1. The BDD represents the Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ defined in the following way. An input $a \in \{0,1\}^n$ activates all edges consistent with a, i.e., the edges labeled by a_i which leave nodes labeled by x_i. The value of the sink reached by the unique path which starts at the source and is activated by a is defined as $f(a)$.

Definition 2. (i) A free BDD (FBDD or read-once branching program) is a BDD where each path contains for each variable x_i at most one node labeled by x_i.

(ii) An ordered BDD (OBDD) is an FBDD where the orderings of the variables on all paths are consistent with one ordering.

(iii) A k-OBDD consists of k layers of OBDDs respecting the same ordering.

(iv) A k-IBDD consists of k layers of OBDDs respecting perhaps different orderings.

The size of a BDD G is the number of its nodes and is denoted by $|G|$.

OBDDs introduced by Bryant (1986) are the most popular representation for formal circuit verification. Gergov and Meinel (1994) and Sieling and Wegener (1995) independently have shown how FBDDs can be used for verification. In k-OBDDs and k-IBDDs the condition that variables are tested only once is relaxed. For k-OBDDs and constant k the test whether two decision diagrams represent the same function can be performed in polynomial time. This problem is coNP-complete already for 2-IBDDs. On the other hand k-IBDDs allow a more compact representation of many functions but the equality test only is based on heuristic algorithms. Bollig, Sauerhoff, Sieling, and Wegener (1997) present results on the representational power of these types of BDDs and Jain, Abadir, Bitner, Fussell, and Abraham (1992) have performed and described experiments.

Even more recently it has been shown that BDD models allowing restricted nondeterminism can be handled efficiently.

Definition 3. A $\oplus$-OBDD is a generalized OBDD where the number of edges leaving a node is not restricted. The output value for an input a is defined as 1 iff the number of paths between the source and the 1-sink activated by a is odd.

Here nondeterminism (inputs can activate many paths) is combined with the $\oplus$-acceptance mode. Nevertheless, Waack (1997) has shown that all important operations can be performed in polynomial time on $\oplus$-OBDDs. We have to admit that the number of edges of a $\oplus$-OBDD can be quadratic with respect to the number of nodes and it is still open whether the algorithms on $\oplus$-OBDDs are practically efficient.

The main drawback of OBDDs, k-OBDDs, and $\oplus$-OBDDs is that they only can use one variable ordering. Different variable orderings can be used explicitly in k-IBDDs and implicitly in FBDDs. These models have other drawbacks. For k-IBDDs the equality test is difficult and for FBDDs it is difficult to find a good oracle graph (Sieling and Wegener (1995)) or a good type (Gergov and Meinel (1994)). Partitioned OBDDs introduced by Jain (1993) and Jain, Bitner, Fussell, and Abraham (1992) and more intensively investigated by Narayan, Jain, Fujita, and Sangiovanni-Vincentelli (1996) try to overcome these difficulties.

Definition 4. A k-PBDD (partitioned BDD with k parts where k may depend on the number of variables) consists of k OBDDs whose variable orderings may be different. The output value for an input a is defined as 1 iff at least one of the OBDDs computes 1 on a. A PBDD is a k-PBDD for some k. The size of a k-PBDD is the sum of the sizes of the k OBDDs.

The model of k-PBDDs allows a restricted form of nondeterminism. In the case of OBDDs it is a restriction that nondeterminism only can be used at the source. Such a form of nondeterminism can be useful even if all OBDDs use the same variable ordering, see, e.g., Hu, York, and Dill (1994). The model of k-PBDDs in its general form has several practical drawbacks. Even if the k variable orderings are fixed, it is difficult to check whether two k-PBDDs G' and G'' represent the same function. An input a may lead in the i-th part of G' to 1 but only in the j-th part of G'' where $j \neq i$. For practical purposes it is necessary to work with so-called window functions $w_1, \ldots, w_k$ whose disjunction is the constant 1. The i-th part G_i of a k-PBDD G for f is responsible for the part $w_i^{-1}(1)$, i.e., G_i has to represent the function $f_i := f \wedge w_i$. This allows an efficient equality check.

In certain applications like combinatorial problems it is necessary to count the number of inputs satisfying a function f. This is efficiently possible for OBDDs but not for k-PBDDs, since many parts may compute 1 for an input a. The window functions are called disjoint, if $w_i \wedge w_j = 0$ for all $i \neq j$. Then the number of satisfying inputs is the sum of the number of satisfying inputs for the parts. Only in this case the notion "partitioned" is really justified. This kind of nondeterminism is known in complexity theory as unique nondeterminism. We use the notions k-PBDD^w and k-PBDD^d, if we work with fixed window functions resp. fixed disjoint window functions. If nothing else is explicitly mentioned, we prove our upper bounds for the weakest model of k-PBDD^{d}s and our lower bounds for the strongest model of k-PBDDs. Moreover, the upper bounds hold for window functions with small OBDD representations.

In the following we denote for a BDD model M by $P(M)$ the class of all (sequences of) Boolean functions representable by polynomial-size BDDs of type M.

In Section 2 we shortly compare the different types of PBDDs. In Section 3 we prove our main result that $P((k-1)$-PBDD) is a proper subclass of $P(k$-PBDD), i.e., the classes $P(k$-PBDD) build a tight hierarchy. The motivation of this result is twofold. First, the hierachy theorem itself as a typical complexity theoretical result. Second, we present lower bound techniques which are subtle enough to obtain large lower bounds for $k-1$ parts even if the function can be easily and efficiently represented with k parts. In Section 4 we compare the expressive power of partitioned OBDDs with the expressive power of the other mentioned BDD-models. In Section 5 we state some structural results.

2 THE DIFFERENT MODELS OF PARTITIONED BDDs

We present an example which shows that we loose expressive power if we ask for efficiently representable window functions. The function PERM is defined on quadratic Boolean matrices X and outputs 1 iff X is a permutation matrix. By NPERM we denote the negation of PERM.

Theorem 5. *(i) $NPERM \in P$(2-PBDD).*
(ii) $NPERM \notin P(PBDD^w)$ for window functions representable by polynomial-size OBBDs for the corresponding variable orderings.

Proof. By definition $NPERM(X) = 1$ iff there exists a row containing not exactly one 1 or there exists a column containing not exactly one 1. This leads directly to a 2-PBDD of linear size, where in the first part the first condition is tested by a rowwise variable ordering and in the second part the second condition is tested by a columnwise variable ordering.

For the lower bound we apply a result of Krause, Meinel, and Waack (1991) that PERM has no polynomial-size nondeterministic read-once branching programs. This implies that $PERM \notin P$(PBDD). Hence, it is sufficient to prove that for functions f in $P(\text{PBDD}^w)$ with efficiently representable window function the negative functions $\overline{f}$ are also contained in $P(\text{PBDD}^w)$. Let G_i represent $w_i \wedge f$. It is easy to obtain an OBDD $\overline{G_i}$ for $\overline{w_i \wedge f} = \overline{w_i} \vee \overline{f}$ of the same size as G_i. By assumption there exists a polynomial-size OBDD G_i^w for w_i with the same variable ordering as G_i and $\overline{G}_i$. Already Bryant (1986) has proved that we can obtain in time $O(|G_i||G_i^w|)$ an OBDD for $w_i \wedge (\overline{w_i} \vee \overline{f}) = w_i \wedge \overline{f}$. □

3 THE HIERARCHY THEOREM

With respect to the results in Section 4 we try to define a function in $P(k\text{-PBDD}^d) - P((k-1)\text{-PBDD})$ which is also easy for many other BDD-models. The idea is to start with a function in P(OBDD) which has small size OBDD representations only if most of a certain set of variables are tested at the beginning of the variable ordering. Then k variants of this function where for each variant another set of variables has to be tested first are combined to one function.

The function describes a graph problem on graphs with $k - 1 + (k + 1)n$ vertices $u_1, \ldots, u_{k-1}$ and $v_{i,j}, 1 \le i \le k+1, 0 \le j \le n-1$. Let U denote the set of the u-vertices and V_i the set of the vertices $v_{i,j}, 0 \le j \le n-1$. The variables $z_1, \ldots, z_{k-1}$ and $x_{i,j}, 1 \le i \le k, 0 \le j \le n-1$, may take values in $\{0, \ldots, n-1\}$ and describe the pointers starting at the vertices $u_1, \ldots, u_{k-1}$ and $v_{i,j}$ resp. If $z_i = j$, the pointer from u_i leads to $x_{1,j}$. If $x_{i,j} = m$, the pointer from $v_{i,j}$ leads to $v_{i+1,m}$. Moreover, there are n Boolean so-called color variables $c_0, \ldots, c_{n-1}$ describing the colors of the nodes in V_{k+1}, i.e., the color of $v_{k+1,j}$ is black if $c_j = 1$ and white otherwise. Each input describes a graph with $k-1$ paths starting in U and ending in V_{k+1}. The output of the function $f^1_{k,n}$ is 1 if the number of paths reaching a black vertex is odd. In order to obtain a Boolean function we have to replace each z- and x-variable by $\log n$ Boolean variables (w.l.o.g. we assume that n is a power of 2).

Lemma 6. *The function $f^1_{k,n}$ can be represented by OBDDs of size $O(2^k k^2 n^k)$.*

Proof. We use the natural variable ordering where the pointers are tested with respect to the vertex sets $U, V_1, \dots, V_k$ and where finally the color variables are tested. The z-variables are tested in a complete binary tree with n^{k-1} leaves. During the tests of the $x_{i,\cdot}$-variables we store about each path the information which vertex in V_i or, if this is already known, in V_{i+1} is reached. These are $(2n)^{k-1}$ possible informations. It is only necessary to test the pointers leaving vertices in V_i which are reached by at least one path. This leads to an additional factor of $O(kn)$. The situation for the color variables is even simper. We get the last factor $O(k)$ for the different vertex sets $V_1, \dots, V_{k+1}$. □

The function $f^i_{k,n}$ is defined similar to $f^1_{k,n}$ with another meaning of the x-variables. The pointers leaving V_1 are now described by the $x_{i,\cdot}$-variables, more generally the pointers leaving V_j are described by the $x_{i+j-1,\cdot}$-variables, if $i+j-1 \le k$, and by the $x_{i+j-1-k,\cdot}$-variables else. The x-variables are rotated blockwise during the definition of $f^1_{k,n}, \dots, f^k_{k,n}$. Obviously, Lemma 6 holds also for $f^i_{k,n}$.

For the definition of the path function $P_{k,n}$ we add $\lceil \log k \rceil$ additional selection variables $s_0, \dots, s_{\lceil \log k \rceil - 1}$. If the value $||s||$ of the s-variables interpreted as binary number is equal to $i \le k-1$, we have to evaluate $f^{i+1}_{k,n}$. In the other case we have to output 0.

Lemma 7. *The function $P_{k,n}$ can be represented by k-PBDD^{d}s of size $O(2^k k^3 n^k)$.*

Proof. The window function w_i for $1 \le i \le k-1$ tests whether $||s|| = i-1$ and w_k tests whether $||s|| \ge k-1$. These disjoint window functions have for all variable orderings an OBDD size of $O(\log k)$. The i-th part of the PBDD can realize $w^*_i \wedge f^i_{k,n}$, where w^*_i tests whether $||s|| = i-1$, with an appropriate variable ordering by Lemma 6 in size $O(2^k k^2 n^k)$. □

The proof of the lower bound for $(k-1)$PBDDs representing $P_{k,n}$ is technically involved. We present the proof for the function $P^*_{k,n}$ where the pointer variables may take values in $\{0, \dots, n-1\}$ and are not divided into $\log n$ Boolean variables. This implies that a pointer is tested as a whole. In the Boolean case the $\log n$ variables of a pointer may be scattered over the variable ordering. Later we discuss how our proof works in the Boolean case. In the proof decision diagrams are used where the nodes for the pointer variables have outdegree n. This only is done for the ease of description. Each of these decision nodes with its edges easily can be replaced by a complete binary tree of depth $\log n$.

Theorem 8. *Each* $(k-1)$*-PBDD for* $P^*_{k,n}$ *has size* $2^{\Omega(n/k^4)}$.

Proof. Let G be a $(k-1)$-PBDD for $P^*_{k,n}$ whose parts $G_1, \ldots, G_{k-1}$ respect the variable orderings $\pi_1, \ldots, \pi_{k-1}$. W.l.o.g. $k^4 \le n$. Otherwise the bound is trivial.

The number of x-variables equals nk. Let A_j be the set of those n x-variables which are tested at first with respect to π_j. Hence, we can choose at least n x-variables not contained in any A_j. Moreover, we choose an index i such that at least n/k $x_{i,\cdot}$-variables are not tested among the first n x-variables with respect to any of the $k-1$ variable orderings. We fix the s-variables such that $||s|| = i-1$ and the z-variables such that the pointers from $u_1, \ldots, u_{k-1}$ point for $f^i_{k,n}$ to $k-1$ different vertices in V_i whose pointers belong to the chosen $x_{i,\cdot}$-variables. W.l.o.g. $i = 1$ and the u-pointers point to $v_{1,0}, \ldots, v_{1,k-2}$. In the following we consider the resulting $(k-1)$-PBDD for the obtained subfunction $f^*_{k,n}$ of $f^1_{k,n}$. We still have $k-1$ parts to represent $f^*_{k,n}$ but we know that all variable orderings are intuitively bad.

With respect to π_j there are n/k $x_{1,\cdot}$-variables not tested among the first n x-variables. Hence, at least n/k of the first n x-variables with respect to π_j do not belong to pointers leaving V_1. We choose a set B_j of at least n/k^2 pointers which leave the same set $V_{j'}$, $j' \ne 1$, and which are tested before the pointers leaving $V_1^* = \{v_{1,0}, \ldots, v_{1,k-2}\}$. Let C_j be a subset of B_j of at least n/k^3 pointers such that the sets $C_1, \ldots, C_{k-1}$ are pairwise disjoint. We associate with C_j the index set I_j of all indices m such that the pointer leaving $v_{j',m}$ belongs to C_j. We may construct the sets C_j even in such a way that the sets $I_1, \ldots, I_{k-1}$ are pairwise disjoint. Let $K := \lceil n/k^3 \rceil$.

Now we further restrict our function. The pointer from $v_{1,j}$ may point only to a vertex $v_{2,m}$ where $m \in I_j$. The pointer from $v_{i,m}$, where $2 \le i \le k-1$, $m \in I_j$, and $i \ne j'$, leads to $v_{i+1,m}$. The pointer from $v_{j',m}$, where $m \in I_j$, may point only to a vertex $v_{j'+1,l}$ where $l \in I_j$. Now the paths have been made independent. Finally, we fix the color variables in such a way that half of the vertices $v_{k+1,m}$, $m \in I_j$, are white and the other half is black. All paths have length k, since the start vertices belong to V_1. For the j-th path the first pointer is not fixed and may lead to one of K vertices. Then the path does not change the index of the vertex until $V_{j'}$ is reached. There we again have K possibilities and afterwards the path does not change the index of the vertex. Hence, we may interpret the path as a path of length 2.

For the j-th part G_j perhaps the color of several paths (more exactly their last vertices) can easily be determined but for the j-th path all variables describing the second edge of the path are tested before the first edge of the path. By our restrictions we have made the paths completely independent. In order to know whether the number of black paths is odd we have to know the color of all paths.

Our restricted OBDDs $G_1, \ldots, G_{k-1}$ all have to output 0 if the function $f^*_{k,n}$ outputs 0 and at least one of them has to output 1 if the function $f^*_{k,n}$ does so. We simplify the task for the OBDD G_j by giving the information about the color of all paths but the j-th one for free. Then G_j is left with one path starting at a definite vertex whose pointer may lead to K vertices which may lead to K vertices half of which are white and black resp. But G_j has to test all possible second edges before the first edge. The OBDD G_j has to output one color, w.l.o.g. white, correctly and for the other color it may make mistakes. Since for each input there has to be one OBDD G_j giving the correct answer, there has to exist one G_j which answers correctly for at least a fraction of $1/(k-1)$ of the inputs belonging to a black path. W.l.o.g. $j = 1$.

In G_1 we investigate the cut where all possible second edges have been tested. Here we may distinguish 2^K input sets describing which of the K vertices lead to a black vertex. If the PBDD size is bounded by S these 2^K information patterns are collected in at most S disjoint nodes where the first edge is tested. By assumption G_1 computes 1 for at least $K2^K/(2(k-1))$ of the $K2^K/2$ "inputs" consisting of the first edge and the set of vertices leading to a black vertex. Let w be a node of the cut of G_1. If less than $K/(4(k-1))$ of the edges leaving w lead to the 1-sink, the node w is called "poor". By definition the other nodes called "rich" nodes have to ensure that G_1 computes for at least $K2^K/(4(k-1))$ of the inputs the output 1. Let w be a rich node. Then we can assume w.l.o.g. that the first $N' \geq N := \lceil K/(4(k-1)) \rceil$ edges leaving w lead to the 1-sink. Hence, we have to be sure that the N' vertices which are reached if the first edge takes one of the values $0, \ldots, N'-1$ lead to a black vertex. This implies that such a node is passed by at most $N'2^{K-N'} \leq N2^{K-N}$ inputs. Hence,

$$S \geq \frac{K2^K}{4(k-1)N2^{K-N}} = \frac{K2^N}{4(k-1)N} \geq 2^{N-1} \geq 2^{\Omega(n/k^4)}. \qquad \square$$

From Lemma 7 and Theorem 8 we obtain a polynomial upper bound for k-PBDD^{d}s and an exponential lower bound for $(k-1)$-PBDDs only for constant k. For a separation result between $P(k\text{-PBDD}^d)$ and $P((k-1)\text{-PBDD})$ a non polynomial lower bound is sufficient. We apply the well-known padding technique. E.g., we can work with $r = \log^6 n$ "real" vertices in each V_j and $n - r$ dummy vertices. Then the upper bound of Lemma 7 is polynomial for $k = O((\log n)/\log\log n)$. The lower bound of Theorem 8 is larger than each polynomial.

It is more difficult to obtain a lower bound on the size of $(k-1)$-PBDDs for $P_{k,n}$. Then we only can ensure that a few, namely $(\log n)/k$, bit variables of the pointers leaving w.l.o.g. $v_{1,0}, \ldots, v_{1,k-2}$ for $f^1_{k,n}$ are tested later than $(n \log n)/k$ x-variables which do not belong to V_1-pointers for each of the orderings $\pi_1, \ldots, \pi_{k-1}$. In order to obtain a situation of $k-1$ independent paths like in the proof of Theorem 8 we have to replace more variables by constants. Instead of $K = \lceil n/k^3 \rceil$ interior vertices on the paths of length 2 our proof for the Boolean case only can ensure the existence of $\lceil n^{1/k}(\log^{-1} n)k^{-4}/2 \rceil$ interior vertices on the independent paths.

Theorem 8*. *Each $(k-1)$-PBDD for $P_{k,n}$ has size at least 2^B where $B = \Omega(n^{1/k}k^{-5}\log^{-1} n)$.*

For a hierarchy theorem we restrict the vertex sets to $r = 2^{(\log^{1/2} n)\alpha(n)}$ "real" vertices where $\alpha(n) = \omega((\log\log n)^{1/2})$. For $k = O((\log^{1/2} n)\alpha^{-1}(n))$ the upper bound of Lemma 7 becomes polynomial while the lower bound of Theorem 8* grows faster than each polynomial. This leads to the following hierarchy result.

Theorem 9. *There are Boolean functions contained in* $P(k\text{-PBDD}^d)$ *but not in* $P((k-1)\text{-PBDD})$, *if* $k = o(((\log n)/\log\log n)^{1/2})$.

4 A COMPARISON WITH OTHER BDD MODELS

The function $P_{k,n}$ is not only easy for k-PBDDs but also for 2-OBDDs, FBDDs, and $\oplus$-OBDDs.

Theorem 10. *The function* $P_{k,n}$ *can be represented by k-PBDDs, 2-OBDDs, and FBDDs of size* $O(2^k k^3 n^k)$, *and by* $\oplus$*-OBDDs of size* $O(2^{3k} k^9 n^{3k})$.

Proof. The claim for k-PBDDs is proved in Lemma 7. The FBDD and the 2-OBDD start with a complete binary tree of the s-variables. If $||s|| = i \le k-1$, $f^{i+1}_{k,n}$ is represented by the OBDD constructed in the proof of Lemma 6. This obviously leads to an FBDD. Using the variable ordering of the OBDD for $f^1_{k,n}$ we obtain 2-OBDDs for all $f^i_{k,n}$. In the representation for $f^i_{k,n}$ we test all variables $x_{m,\cdot}$ where $m \ge i$ in the right order in the first layer and all other x-variables in the right order in the second layer. Now the result for $\oplus$-OBDDs follows from the proof of Theorem 12, part viii). □

Now we look for a function which is easy for 2-PBDD^{d}s and difficult for many other models.

Theorem 11. *There exists a function* f_n *on* n^2 *Boolean variables which can be represented in 2-PBDD*^{d}*s of size* $O(n^2)$ *and which needs size* $2^{\Omega(n^{1/2})}$ *for FBDDs, size* $2^{\Omega(n/k)}$ *for k-OBDDs, and size* $2^{\Omega(n)}$ *for* $\oplus$*-OBDDs.*

Sketch of proof. The function f_n is defined on $n \times n$ Boolean matrices X and outputs 1 iff the matrix X contains an odd number of ones and a row consisting of ones only or an even number of ones and a column consisting of ones only.

For the 2-PBDD^d we use the disjoint window functions testing whether the number of ones in X is odd and even resp. They have linear OBDD size for all variable orderings. The upper bound for 2-PBDD^{d}s is obvious.

The standard lower bound technique for FBDDs shows that the corresponding tree looks like a tree up to a certain depth d. This technique cannot be applied here. Instead of that we consider $\binom{n-1}{n^{1/2}}$ paths with special properties and prove that at most $(n^{1/2})!$ of them can be merged in one FBDD node. This implies the lower bound.

For the other lower bounds we apply a lower bound technique due to Krause (1991). For a given variable ordering we consider the cut where for the first time $n-2$ rows or columns have a tested variable. W.l.o.g. these are the first $n-2$ rows. We consider the following assignments of the variables tested before the cut. Exactly half of the variables tested first in the first $n-2$ rows are replaced by ones and the other by zeros. All other tested variables are replaced by ones. For each of these assignments we define a corresponding assignment of the remaining variables in the following way. The rows consisting up to now only of ones are filled with zeros and the rows with a zero are filled with ones. The last but one row is filled with zeros ensuring that no column contains only ones and the last row is filled always in a way ensuring that the total number of ones is odd. Each assignment leads to the output 0. But if we combine one assignment of the variables before the cut with a non-corresponding assignment of the other variables this leads to the output 1. Hence, we obtain a fooling set of size $\binom{n-2}{(n-2)/2} = 2^{\Omega(n)}$. The lower bound for k-OBDDs follows from the lower bound of Krause (1991). The lower bound for $\oplus$-OBDDs follows from a new result of Dietzfelbinger and Savický (1997) that such fooling sets also lead to lower bounds for $\oplus$-OBDDs. $\square$

Theorem 12. *(i)* $P(\text{2-PBDD}^d) \not\subseteq P(\text{FBDD})$.
(ii) $P(\text{FBDD}) \not\subseteq P(k\text{-PBDD})$, *if* $k = o(((\log n)/\log\log n)^{1/2})$.
(iii) $P(\text{2-PBDD}^d) \not\subseteq P(\oplus\text{-OBDD})$.
(iv) $P(\oplus\text{-OBDD}) \not\subseteq P(k\text{-OBDD})$ *if* $k = o(((\log n)/\log\log n)^{1/2})$.
(v) $P(\text{2-PBDD}^d) \not\subseteq P(k\text{-OBDD})$ *if* $k = o(n/\log n)$.
(vi) $P(\text{2-OBDD}) \not\subseteq P(k\text{-PBDD})$ *if* $k = o(((\log n)/\log\log n)^{1/2})$.
(vii) $P(k\text{-OBDD}) \subseteq P(\text{PBDD}^d)$ *if* $k = O(1)$.
(viii) $P(k\text{-OBDD}) \subseteq P(\oplus\text{-OBDD})$ *if* $k = O(1)$.
(ix) $P(k\text{-PBDD}) \subseteq P(k\text{-IBDD})$ *for all* k.

Sketch of proof. Part i), iii), and v) follow from Theorem 11. Part ii), iv), and vi) follow from Theorem 8* and Theorem 10. For part vii) consider the at most $s = |G|^{k-1}$ possibilities to switch between the layers of a k-OBDD G. The i-th window function, $1 \leq i \leq s$, equals 1 for the inputs choosing the i-th possibility. Each part of the considered function can be represented by an OBDD of size $|G|^k$. All these OBDDs use the variable ordering of G. Therefore, it is easy to obtain also a $\oplus$-OBDD of size $|G|^{2k-1}$. Part ix) is easy. $\square$

5 READ-ONCE PROJECTIONS

Definition 13. The function $f = (f_n)$ is a read-once projection of $g = (g_n)$, $f \leq_{rop} g$, if $f_n(x_1, \ldots, x_n) = g_{p(n)}(y_1, \ldots, y_{p(n)})$ for some polynomial p, $y_j \in \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}, 0, 1\}$, and for all i the restriction that $y_j \in \{x_i, \overline{x_i}\}$ for at most one j is fulfilled.

Bollig and Wegener (1996) have shown that read-once projections are an adequate reduction concept for BDD models. If $f \leq_{rop} g$ and $g \in P(k\text{-PBDD})$, then also $f \in P(k\text{-PBDD})$. The same holds for the classes $P(k\text{-PBDD}^w)$, $P(k\text{-PBDD}^d)$, and $P(\oplus\text{-PBDD})$.

Theorem 14. *The complexity classes $P(k\text{-PBDD})$ and $P(\oplus\text{-OBDD})$ have complete problems with respect to read-once projections.*

Sketch of proof. The proof for $P(\oplus\text{-OBDD})$ follows the lines of the proof of the completeness result for $P(\text{OBDD})$ (Bollig and Wegener (1996)) and the proof for $P(k\text{-PBDD})$ is an extension of the proof of the completeness result for $P(k\text{-IBDD})$ (Bollig and Wegener (1996)). □

References

Bollig, B., Sauerhoff, M., Sieling, D., and Wegener, I.: Hierarchy theorems for kOBDDs and kIBDDs. Accepted for publication in Theoretical Computer Science (1997). (Preliminary version: On the power of different types of restricted branching programs. Electronic Colloquium on Computational Complexity. TR 94-026).

Bollig, B. and Wegener, I.: Read-once projections and formal circuit verification with binary decision diagrams. STACS'96, Lecture Notes in Computer Science **1046** (1996), 491 – 502.

Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Trans. on Computers **35** (1986), 677 – 691.

Dietzfelbinger, M. and Savický, P.: Parity OBDDs cannot represent the multiplication succinctly. Manuscript, Univ. Dortmund (1997).

Gergov, J. and Meinel, C.: Efficient Boolean manipulation with OBDDs can be extended to FBDDs. IEEE Trans. on Computers **43** (1994), 1197 – 1209.

Hu, A.J., York, G., and Dill, D.L.: New techniques for efficient verification with implicitly conjoined BDDs. Proc. of 31st ACM/IEEE Design Automation Conference (1994), 276 – 282.

Jain, J.: On Analysis of Boolean Functions. Ph.D. Thesis, Dept. of Electrical and Computer Engineering, University of Texas at Austin (1993).

Jain, J., Abadir, M., Bitner, J., Fussell, D.S., and Abraham, J.A.: IBDDs: an efficient functional representation for digital circuits. Proc. European Design Automation Conference (1992), 440 – 446.

Jain, J., Bitner, J., Fussell, D.S., and Abraham, J.A.: Functional partitioning for verification and related problems. Brown/MIT VLSI Conf. (1992), 210 – 226.

Krause, M.: Lower bounds for depth-restricted branching programs. Information and Computation **91** (1991), 1 – 14.

Krause, M., Meinel, C., and Waack, S.: Separating the eraser Turing machine classes L_e, NL_e, co-NL_e and P_e. Theoretical Computer Science **86** (1991), 267 –275.

Narayan, A., Jain, J., Fujita, M., and Sangiovanni-Vincentelli, A.: Partitioned ROBDDs - a compact, canonical and efficiently manipulable representation for Boolean functions. Proc. of IEEE/ACM Int. Conf. on Computer Aided Design ICCAD'96 (1996), 547 – 554.

Sieling, D. and Wegener, I.: Graph driven BDDs - a new data structure for Boolean functions. Theoretical Computer Science **141** (1995), 283 – 310.

Waack, S.: On the descriptive and algorithmic power of parity ordered binary decision diagrams. Proc. of STACS'97, LNCS 1200 (1997), 201 – 212.