Theoretical Informatics and Applications
Theoret. Informatics Appl. **36** (2002) 229–247
DOI: 10.1051/ita:2002011

CHARACTERIZING THE COMPLEXITY OF BOOLEAN FUNCTIONS REPRESENTED BY WELL-STRUCTURED GRAPH-DRIVEN PARITY-FBDDS *, **

Henrik Brosenne[1], Matthias Homeister[1] and Stephan Waack[1]

Abstract. We investigate well-structured graph-driven parity-FBDDs, which strictly generalize the two well-known models parity OBDDs and well-structured graph-driven FBDDs. The first main result is a characterization of the complexity of Boolean functions represented by well-structured graph-driven parity-FBDDs in terms of invariants of the function represented and the graph-ordering used. As a consequence, we derive a lower bound criterion and prove an exponential lower bound for certain linear code functions. The second main result of this paper is a polynomial time algorithm that minimizes the number of nodes in a graph-driven parity-FBDD.

Mathematics Subject Classification. 68Q10, 68Q60, 68P05.

Introduction

Branching Programs or Binary Decision Diagrams are a well established model for Boolean functions with applications both in complexity theory and in the theory of data structures for hardware design and verification.

In complexity theory branching programs are a model of sequential space bounded computations. Upper and lower bounds on the branching program size

Keywords and phrases: Well-structured graph-driven parity-FBDDs, lower bounds, minimization algorithm, complexity theory, data structures for Boolean functions.

* *Supported by DFG grant Wa766/4-1.*

** *This paper is the full version of [3].*

[1] Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, Lotzestr. 16-18, 37083 Göttingen, Germany;
e-mail: {homeiste,waack}@math.uni-goettingen.de

for explicitly defined functions are upper and lower bounds on the sequential space complexity of these functions.

Data structures for Boolean functions have to allow succinct representations of as many Boolean functions as possible. They have to admit algorithms for the most important operations. Among others, these are *minimization*, *synthesis* and *equivalence test* (for a survey see [16]). Even if a data structure is of more theoretical rather than of practical interest, minimization and equivalence test are of structural significance.

Let $\mathbb{B}_n$ denote the set of all Boolean functions of n variables. We regard $\mathbb{B}_n$ as an $\mathbb{F}_2$-algebra, where $\mathbb{F}_2$ is the prime field of characteristic 2. The product $f \wedge g$ or fg of two functions $f, g \in \mathbb{B}_n$ is defined by componentwise conjunction. Their sum $f \oplus g$ corresponds to the componentwise exclusive-or. (In line with this notation, "$\oplus$" is also used for the symmetric difference of sets.)

A *(syntactically nondeterministic) binary decision diagram* (BDD for short) B on the Boolean variables $\{x_1, \dots, x_n\}$ is a directed acyclic graph with the following properties. Let $\mathcal{N}(B)$ be the set of nodes of B. There are two distinct nodes s and t called the *source* and the *target* node. The outdegree of the target and the indegree of the source are both equal to zero. The source s is joined to each node of its successor set $\mathrm{Succ}\,(s)$ in B by an unlabeled directed edge. The nodes different from the source and the target are called *branching nodes*. Each branching node w is labeled with a Boolean variable $\mathrm{var}\,(w) \in \{x_1, \dots, x_n\}$, its successor set falls into two subsets $\mathrm{Succ}_0\,(w)$ and $\mathrm{Succ}_1\,(w)$, where, for $b \in \{0, 1\}$, the node w is joined to each $v \in \mathrm{Succ}_b\,(w)$ by a directed edge labeled with b. For $b \in \{0, 1\}$, an element of $\mathrm{Succ}_b\,(w)$ is called a *b-successor* of the node w. Moreover, we assume B to be weakly connected in the following sense. For each branching node w, there is a directed path from the source via this node to the target. The *size* of a BDD B, denoted by $\mathrm{SIZE}\,(B)$, is the number of its nodes. A branching node w of a BDD is called *deterministic*, if $\mathrm{Succ}_b\,(w) \leq 1$, for all $b \in \{0, 1\}$. The source s is called *deterministic*, if $\#\mathrm{Succ}\,(s) \leq 1$. The BDD B as a whole is defined to be *deterministic*, if the source and all branching nodes are deterministic.

An input $a \in \{0, 1\}^n$ *activates* all edges labeled with a_i outgoing from nodes labeled with x_i, for $i = 1, 2, \dots, n$. Moreover, the edges leaving the source are activated by all elements of $a \in \{0, 1\}^n$.

A *computation path* for an input $a \in \{0, 1\}^n$ in a BDD B on $\{x_1, \dots, x_n\}$ is a path in B from the source whose edges are activated by a. Such a path is called an *accepting* one, if it leads to the target.

The variants of decision diagrams tractable in the theory of data structures of Boolean functions as well as in complexity theory restrict the number and the kind of read accesses to the input variables. A very popular model is the following one. A BDD is defined to be a *free BDD* (FBDD for short), if each variable is tested on each path from the source at most once.

There are several ways to let a BDD B on $\{x_1, \dots, x_n\}$ represent a Boolean function $f \in \mathbb{B}_n$. A *parity binary decision diagram* ($\oplus$-BDD for short) is a binary decision diagram equipped with the *parity representation mode*. A $\oplus$-BDD B represents a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ defined as follows. $f(a) = 1$ if and only if the number of accepting computation paths for a is *odd.* A $\oplus$-BDD is defined to be a parity-FBDD ($\oplus$-FBDD for short), if it is free.

From now on we speak about a BDD or a FBDD, only if the diagram is *deterministic.*

Ordered binary decision diagrams (OBDDs), introduced by Bryant (see [5, 6]), are the state-of-the-art data structure for Boolean functions in the logical synthesis process, for verification and test pattern generation, and as a part of CAD tools. They are FBDDs with the following additional property. There is a permutation σ, a so-called *variable ordering*, of the set $\{1, 2, \ldots, n\}$ such that if node v labeled with $x_{\sigma(j)}$ is a successor of node u labeled with $x_{\sigma(i)}$, then $i > j$. Perhaps the most important fact about OBDDs for theory is that a size-minimal OBDD for a function f and a fixed variable ordering σ is uniquely determined. It can be efficiently computed by Bryant's minimization algorithm. But many even simple functions have exponential OBDD-size (see [2, 4]). For this reason models less restrictive than OBDDs are studied.

First we mention $\oplus$-*OBDDs.* They are defined to be $\oplus$-FBDDs subject to variable orderings in the above sense. Introduced by Gergov and Meinel in [9], they have been intensively studied in [15].

Second FBDDs without any further restriction are considered as a data structure. The problem was to find a counterpart to the variable ordering of OBDDs. It was independently solved by Gergov and Meinel in [8], and by Sieling and Wegener in [14]. Roughly speaking, the characteristic feature of a FBDD in contrast to OBDDs is that we may use different variable orderings for different inputs.

Definition 0.1. A *graph ordering* G on the set of Boolean variables $\{x_1, \ldots, x_n\}$ is a FBDD which is *complete* in the following sense. The source s has exactly one successor succ (s), each branching node u has exactly one 1-successor and exactly one 0-successor, and on each path from the source to the target there is for each variable x_i exactly one node labeled with x_i.

The relation between a graph ordering and a FBDD guided by it is given in Definition 0.2. Informally spoken, depending on the Boolean variables tested so far and the corresponding input bits retrieved, the graph ordering predicts the next variable to be tested. For later use in this paper, we let not only FBDDs but also $\oplus$-FBDDs be guided in that way.

Definition 0.2. Let G be a graph ordering on $\{x_1, \ldots, x_n\}$. A $\oplus$-FBDD B is defined to be a *graph-driven $\oplus$-FBDD guided by G* if the following condition is satisfied. Let, for any input $b \in \{0, 1\}^n$, $\pi_b^{(G)}$ be the path in G and π_b be an arbitrarily chosen path in B for the input b. If x_i is tested before x_j when traversing π_b, then this is true when traversing $\pi_b^{(G)}$, too.

Note, that OBDDs and $\oplus$-OBDDs can be regarded as guided by graph orderings, the so-called *line orderings.* A more general example of a graph-driven $\oplus$-FBDD is given in Figure 1.

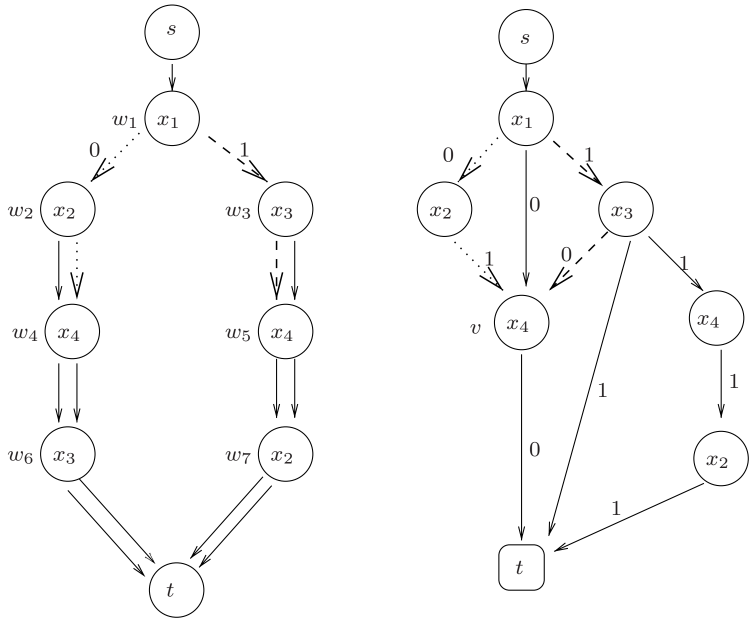

FIGURE 1. A graph ordering and a graph-driven $\oplus$-FBDD guided by this ordering.

Gergov and Meinel, and Sieling and Wegener were able to show, that graph-driven FBDDs efficiently support most of the OBDD-operations.

A drawback of graph-driven FBDDs is, that they do not have "levels" defined by the nodes of the guiding graph ordering such as OBDDs and $\oplus$-OBDDs have. To enforce the existence of levels in the case of FBDDs, Sieling and Wegener [14] introduced what they called well-structured graph-driven FBDDs. Extending this notion in Definition 0.3 to the case of $\oplus$-FBDDs, we get the structure on which this paper is mainly focused.

Definition 0.3. A graph-driven $\oplus$-FBDD B guided by G is defined to be *well-structured*, if there is an additional *level function* $\text{level} : \mathcal{N}(B) \to \mathcal{N}(G)$ with the following properties:

- $\text{level}\,(s) = s$, $\text{level}\,(t) = t$;
- for each branching node $w \in \mathcal{N}(B)$, $\text{level}\,(w)$ is a branching node of G that is labeled with the same Boolean variable as w: $\text{var}\,(w) = \text{var}\,(\text{level}(w))$;
- for $b \in \{0,1\}^n$, let $\pi_b^{(G)}$ be the path in G and let π_b be an arbitrarily chosen path in B for the input b. For each node w, if w is contained in π_b, then $\text{level}\,(w)$ is contained in $\pi_b^{(G)}$.

Let B be a well-structured graph-driven $\oplus$-FBDD guided by G, then the set of nodes of B is partitioned into *levels* as follows. For each node u of G, we define the level $\mathcal{N}_u(B) = \mathcal{N}_u$ of B associated with the node u to be the set $\{w \in \mathcal{N}(B) \mid \text{level}\,(w) = u\}$.

Figure 2 gives an example of a well-structured graph-driven $\oplus$-FBDD, whose level structure is shown in Figure 3.

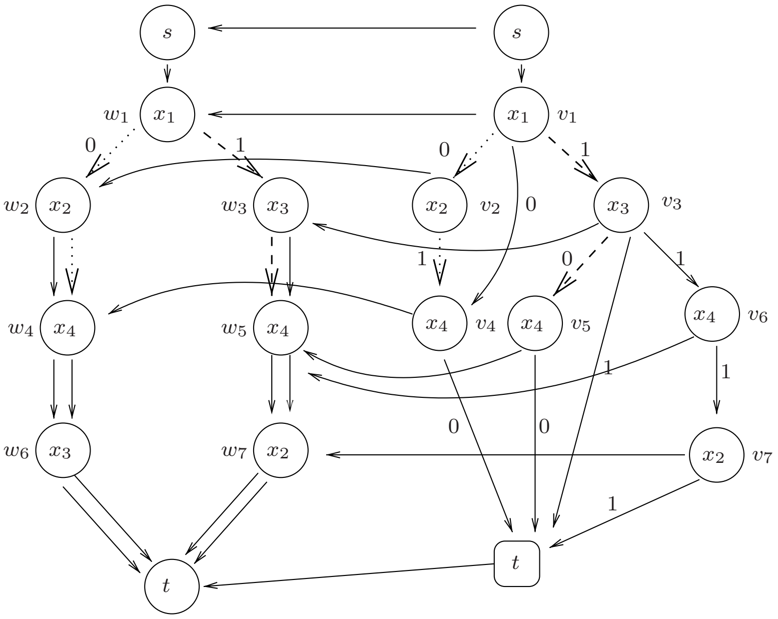

Figure 2. A well-structured graph-driven $\oplus$-FBDD guided by a graph ordering representing the same function as the diagram in Figure 1.

Note, that well-structured graph-driven $\oplus$-FBDDs have a strictly larger descriptive power than both graph-driven FBDDs and $\oplus$-OBDDs. This follows from results due to Sieling and the fact, that the size of graph-driven and well-structured graph-driven FBDDs are polynomially related. Sieling has proved in [13], that there is an explicitely defined function that has polynomial size $\oplus$-OBDDs but exponential size FBDDs, whereas another function has polynomial size FBDDs but exponential size $\oplus$-OBDDs.

The results of this paper can be summarized as follows. An algebraic characterization (see Th. 1.7) of the well-structured graph-driven $\oplus$-FBDD complexity serves as basis both for lower and for upper bounds.

Having derived a lower bound criterion (see Cor. 1.9), we are able to prove exponential lower bounds on the size of well-structured graph-driven $\oplus$-FBDDs for linear code functions. This extends an analogous result for $\oplus$-OBDDs due to Jukna (see [11]).

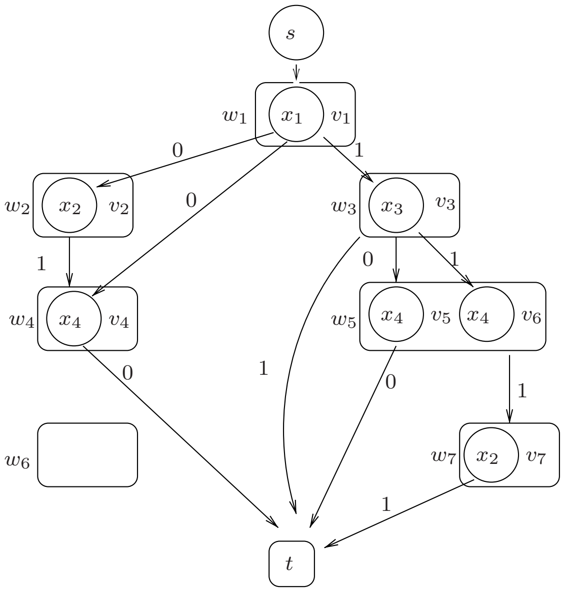

Figure 3. The level-structure of the well-structured graph-driven $\oplus$-FBDD presented in Figure 2.

Moreover, we establish deterministic polynomial time algorithms for solving the minimization problem for the number of nodes (see Th. 2.1), and the equivalence test problem (see Cor. 2.2) for well-structured graph-driven $\oplus$-FBDDs.

Recently, Bollig *et al.* have proved in [1] strong exponential lower bound on the size of well-structured graph-driven $\oplus$-FBDDs for integer multiplication by another criterion, which follows from our Theorem 1.7, too.

1. Algebraic characterization and lower bounds

Throughout this section let us fix a graph-driven $\oplus$-FBDD B on the set of Boolean variables $\{x_1, \ldots, x_n\}$ guided by a graph ordering G that represents a Boolean function $f \in \mathbb{B}_n$.

Recall, that we can identify the nodes of G with the levels of B. A level of the diagram B is called a *branching level* if the corresponding node of the graph ordering G is a branching node.

Our first problem is to determine which nodes of B can be joined together by a directed edge without violating Definition 0.2. Lemma 1.4 gives the answer. But we need a little more notation.

Definition 1.1. A level u_1 is defined to be *greater than* another level u_2 ($u_1 > u_2$ for short) if and only if any path from u_1 to the target node passes through u_2.

Recall, that an element b of a partially ordered set P (a poset for short) *covers* another element a (abbreviation: $b \succ a$), if b is greater than a, but the open interval (a, b) in P is empty.

The following lemma is obvious:

Lemma 1.2. *The relation of Definition 1.1 is a partial ordering on G.*

Moreover, each level w covers exactly one other level.

Definition 1.3. We call G equipped with the partial ordering defined in Definition 1.1 the *level poset* associated with the graph ordering G.

Lemma 1.4. *Let u be any node of B joined to a node v by a directed edge e.*

If the node u is a branching node, and the edge e is labeled with b ($b \in \{0,1\}$), then $\mathrm{level}\,(u)$*'s b-successor is greater than or equal to* $\mathrm{level}\,(v)$*.*

If the node u is equal to the source of B, then the source of G is greater than $\mathrm{level}\,(v)$*.*

Proof. Let us fix an input $a = (a_1, \dots , a_n) \in \{0,1\}^n$ such that there is a path π_a in B from the source to the target that passes through the edge $u \xrightarrow{b} v$. The path $\pi_a^{(G)}$ in G associated with a contains a subpath $\mathrm{level}\,(u) \xrightarrow{b} w \xrightarrow{*} \mathrm{level}\,(v)$. Without loss of generality, let $\mathrm{var}\,(u) = x_1$, and let $\{x_2, \dots , x_k\}$ be the set of variables tested on the subpath $w \xrightarrow{*} \mathrm{level}\,(v)$ of $\pi_a^{(G)}$ with the label of $\mathrm{level}\,(v)$ being excluded. Let $(a'_2, \dots , a'_k)$ be any assignment to the variables $(x_2, \dots , x_k)$, and let $a' := (a_1, a'_2, \dots , a'_k, a_{k+1}, \dots , a_n)$. By Definition 0.2, the two paths $\pi_a^{(G)}$ and $\pi_{a'}^{(G)}$ may diverge at node w or at one of its successors, but they have to converge at node $\mathrm{level}\,(v)$ at the latest. The first claim follows. The second claim can be similarly proved. □

Figure 4 illustrates the result of Lemma 1.4.

The *Hasse diagram* of a poset is a directed acyclic graph whose *nodes* are the *elements* of the poset. An *edge* indicates, that the upper element *covers* the lower one.

Since for each $v \neq s$ in G there is exactly one u such that $u \succ v$, the Hasse diagram of a level poset is always a tree.

The Hasse diagram of the level poset associated with the graph ordering of Figure 4 can be seen in Figure 5.

Next we associate with each node u of the graph ordering G the following $\mathbb{F}_2$-vector space.

$$\mathbb{B}_u(f) := \mathrm{span}_{\mathbb{F}_2} \bigcup_{v \leq u} \{ f|_{\alpha(\pi)} \mid \pi \text{ is a path in } G \text{ from } s \text{ to } v \} \,, \qquad (1)$$

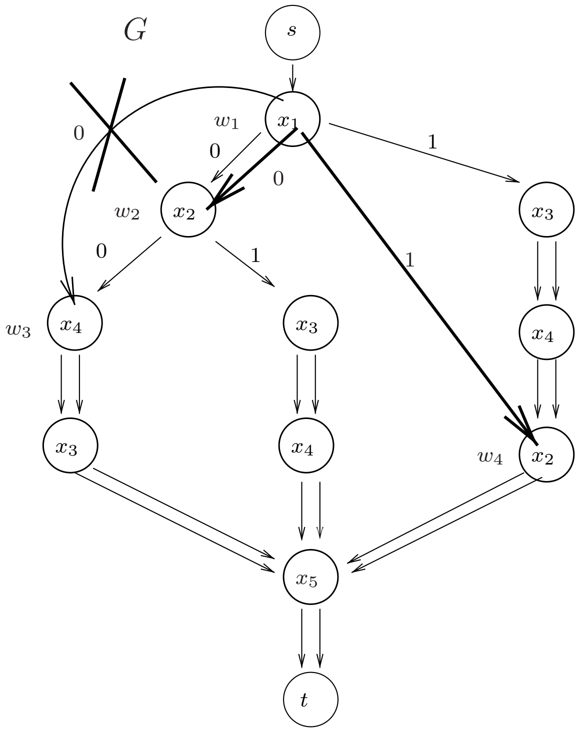

Figure 4. Allowed and forbidden edges in a well-structured graph-driven $\oplus$-FBDD guided by G.

where $\alpha(\pi)$ is defined to be the partial assignment to $\{x_1, x_2, \dots, x_n\}$ canonically associated with the path π. It is obvious that

$$\mathbb{B}_s(f) = \mathbb{B}_u(f),$$

where $s \succ u$. Thus we are no longer interested in the source s as element of the level poset.

Assume the level u_b to be the b-successor of a branching level u in the graph ordering G ($b \in \{0, 1\}$). Let x be the Boolean variable with which u is labeled. Then the mapping

$$\begin{aligned} \mathbb{B}_u(f) &\longrightarrow \mathbb{B}_{u_b}(f) \\ h &\mapsto h\,|_{x=b} \end{aligned} \tag{2}$$

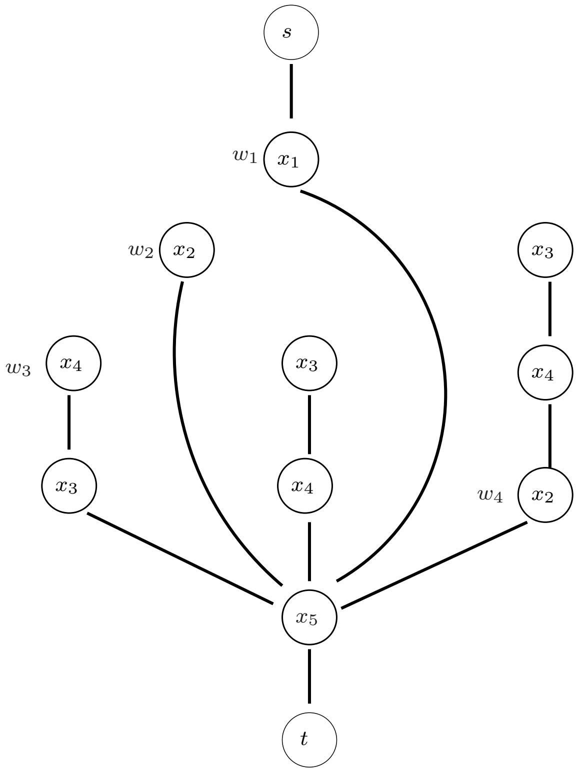

FIGURE 5. The Hasse diagram of the level poset associated with the graph ordering of Figure 4.

is an $\mathbb{F}_2$-vector space homomorphism onto $\mathbb{B}_{u_b}(f)$. By Shannon's decomposition there is a canonical one-to-one homomorphism

$$\begin{aligned} \mathbb{B}_u(f) &\longrightarrow \mathbb{B}_{u_0}(f) \oplus \mathbb{B}_{u_1}(f) \\ h &\mapsto (h\,|_{x=0}\,, h\,|_{x=1}\,)\,, \end{aligned} \tag{3}$$

where here $\mathbb{B}_{u_0}(f) \oplus \mathbb{B}_{u_1}(f)$ denotes the direct sum of the spaces $\mathbb{B}_{u_b}(f)$ $(b \in \{0,1\})$.

There is an equivalent way of defining the function $f \in \mathbb{B}_n$ represented by a $\oplus$-BDD B on $\{x_1, x_2, \ldots, x_n\}$ as the one given in the introduction. For each node u of the diagram B, we inductively define its *resulting function* Res_u. The resulting function of the target equals the all-one function. For a branching node u labeled with the variable x,

$$\mathrm{Res}_u := (x \oplus 1) \wedge \bigoplus_{v \in \mathrm{Succ}_0(u)} \mathrm{Res}_v \;\oplus\; x \wedge \bigoplus_{v \in \mathrm{Succ}_1(u)} \mathrm{Res}_v\,. \tag{4}$$

If s is the source, then

$$\mathrm{Res}_s := \bigoplus_{v \in \mathrm{Succ}(s)} \mathrm{Res}_v \,. \tag{5}$$

The function $\mathrm{Res}(B) : \mathbb{F}_2^n \to \mathbb{F}_2$ represented by the whole diagram is defined to be Res_s.

We have introduced the vector spaces of equation (1) with the intension of characterizing the complexity of the function f in terms of their dimensions. The following spaces are the connecting link to the diagram B representing f.

$$\mathbb{B}_u(B) := \mathrm{span}_{\mathbb{F}_2} \left\{ \mathrm{Res}_w \;\middle|\; w \in \bigcup_{v \le u} \mathcal{N}_u(B) \right\} \cdot \tag{6}$$

Clearly, if both u and v cover w in the level poset, then

$$\mathcal{N}_w(B) = \mathcal{N}_u(B) \cap \mathcal{N}_v(B). \tag{7}$$

Let u be any branching node of G, and let v be the unique node that is covered by u in the level poset associated with G. According to the definition given by equation (6), $\mathbb{B}_u(B) \supseteq \mathbb{B}_v(B)$, and the factor space $\mathbb{B}_u(B)/\mathbb{B}_v(B)$ is generated by the elements $\mathrm{Res}_w \oplus \mathbb{B}_v(B)$, where w ranges over all nodes of the level u. Consequently,

$$\#\mathcal{N}_u(B) \ge \dim_{\mathbb{F}_2} \mathbb{B}_u(B) \;-\; \dim_{\mathbb{F}_2} \mathbb{B}_v(B). \tag{8}$$

If the level u_b is a b-successor of a branching level u in the graph ordering G ($b \in \{0,1\}$), then we have for the vector spaces defined in equation (6) the same situation as in the case of equation (1). The mapping

$$\begin{aligned} \mathbb{B}_u(B) &\longrightarrow \mathbb{B}_{u_b}(B) \\ h &\mapsto h\,|_{x=b} \end{aligned} \tag{9}$$

is an $\mathbb{F}_2$-vector space homomorphism onto $\mathbb{B}_{u_b}(B)$, where x be the Boolean variable with which u is labeled. Moreover, it is plain that there is an embedding analogous to equation (3):

$$\begin{aligned} \mathbb{B}_u(B) &\longrightarrow \mathbb{B}_{u_0}(B) \oplus \mathbb{B}_{u_1}(f) \\ h &\mapsto (h\,|_{x=0}\,, h\,|_{x=1}\,)\,, \end{aligned} \tag{10}$$

where here again $\mathbb{B}_{u_0}(f) \oplus \mathbb{B}_{u_1}(f)$ denotes the direct sum of the spaces $\mathbb{B}_{u_b}(f)$ ($b \in \{0,1\}$).

Lemma 1.5. *Let u be any node of G. Then*

$$\mathbb{B}_u(f) \subseteq \mathbb{B}_u(B). \tag{11}$$

Proof. Let π be any path leading to the node u. Let α be the partial assignment to the variables x_1, x_2, …, x_n canonically associated with π. It suffices to show that $f|_\alpha$ belongs to $\mathbb{B}_u(B)$.

Since the $\oplus$-FBDD B is driven by G, it is possible to follow the partial assignment α in B. Let $v_1, v_2, \dots, v_\nu$ be the nodes of B that can be reached that way. Clearly, the subfunction f_α of f is equal to $\bigoplus_{i=1}^{\nu} \mathrm{Res}_{u_\nu}$. The nodes $v_1, v_2, \dots, v_\nu$ belong to the level u because of the fact that the graph-driven $\oplus$-FBDD B is a well-structured one. □

Lemma 1.6. *Let u be any branching node of G, and let v be the unique node that is covered by u in the level poset associated with G. Then*

$$\#\mathcal{N}_u(B) \geq \dim_{\mathbb{F}_2} \mathbb{B}_u(f) \, - \, \dim_{\mathbb{F}_2} \mathbb{B}_v(f). \tag{12}$$

Proof. By Lemma 1.5 we have a canonical linear mapping

$$\begin{aligned} \phi : \mathbb{B}_u(f) &\to \mathbb{B}_u(B)/\mathbb{B}_v(B) \\ g &\mapsto g \oplus \mathbb{B}_v(B). \end{aligned}$$

Applying the dimension formula for linear mappings and equation (8), we get

$$\dim \mathbb{B}_u(f)/\ker\phi \leq \dim \mathbb{B}_u(B)/\mathbb{B}_v(B) \leq \#\mathcal{N}_u(B).$$

Thus it suffices to show that $\ker\phi = \mathbb{B}_v(f)$.

The inclusion $\ker\phi \supseteq \mathbb{B}_v(f)$ is an immediate consequence of Lemma 1.5. Let $g \in \ker\phi$. Then $g \in \mathbb{B}_v(B)$, because $\ker\phi = \mathbb{B}_u(f) \cap \mathbb{B}_v(B)$ by definitions. Consequently, g does not essentially depend on the set of variable $\mathcal{V}_{u,v}$ tested on any path from u to v in G. Let α be an arbitrarily chosen partial assignment to $\mathcal{V}_{u,v}$. Concatenating the mappings defined by equation (2) for all pairs $(x, \alpha(x))$ $(x \in \mathcal{V}_{u,v})$, we obtain a $\mathbb{F}_2$-linear mapping

$$\begin{aligned} \mathbb{B}_u(f) &\longrightarrow \mathbb{B}_v(f) \\ h &\mapsto h\,|_\alpha \,, \end{aligned}$$

that is invariant on all functions not depending on $\mathcal{V}_{u,v}$. Consequently, g is mapped onto itself. It follows that $g \in \mathbb{B}_v(f)$. □

We are able to formulate and prove Theorem 1.7 that will prove useful both in designing algorithms and in proving lower bounds.

Theorem 1.7. *Let B be a size-minimal well-structered graph-driven $\oplus$-FBDD on the set of variables $\{x_1, \ldots, x_n\}$ guided by a graph ordering G representing $f \in \mathbb{B}_n$. Let u be any branching node of G, and let v be the node that is covered by u in the level poset associated with G. Then*

$$\#\mathcal{N}_u(B) = \dim_{\mathbb{F}_2} \mathbb{B}_u(f) \, - \, \dim_{\mathbb{F}_2} \mathbb{B}_v(f). \tag{13}$$

Proof. By Lemma 1.6, it suffices to construct a graph-driven $\oplus$-BDD B guided by G representing f such that the asserted equations hold.

First, let us turn to the set of nodes of B. We define $\mathcal{N}_s(B)$ to be the set containing only f, $\mathcal{N}_t(B)$ to be the set containing only the all-one function, and $\mathcal{N}_u(B)$ to be a set of representatives of a basis of the space $\mathbb{B}_u(f)/\mathbb{B}_v(f)$, where u is a branching node of G, and v is covered by u in the level poset of G.

Second, we have to inductively create edges in such a way that B represents the Boolean function f. We do that in a bottom-up manner, such that for any node u of the graph ordering G, and any $h \in \mathcal{N}_u(B)$, we have $\mathrm{Res}_h = h$. To this end, we fix a topological ordering of the nodes of G with the least node being the target.

The claim is true for $u = t$. For the induction step, we assume that $u \neq t$, $\mathrm{var}\,(u) = x_l$, for some l, and $h \in \mathcal{N}_u(B)$. For $b \in \{0, 1\}$, let u_b be the b-successor of the node u in the graph ordering G. For $b = 0, 1$, we conclude by Lemma 1.4 $h|_{x_l=b} \in \mathbb{B}_{u_b}(f)$. By induction hypothesis, the functions $h|_{x_l=0}$ and $h|_{x_l=1}$ have unique representations as sums of resulting functions of nodes defined so far. We have to "hardwire" these representations in the decision diagram. For $b = 0, 1$, the node h is joined to any other node h' by a directed edge labeled b if and only if h' occurs in the sum that represents $h|_{x_l=b}$. Thus we have implemented equation (3). □

Theorem 1.7 implicitely contains a lower bound technique. We see that it does not suffice to estimate from below the sum of dimensions of the spaces $\mathbb{B}_u(f)$, where u traverses a fixed depth level of G. This is because of the fact, that if u_1 and u_2 are two of these levels both covering v in the level poset, then the space $\mathbb{B}_v(f)$ is contained in $\mathbb{B}_{u_1}(f)$ as well as $\mathbb{B}_{u_2}(f)$. But if we are able to detect for each such u a large subspace of $\mathbb{B}_u(f)$ that has trivial intersection with $\mathbb{B}_v(f)$, where v is covered by u, we are done.

Taking pattern from [10], we define the following notion:

Definition 1.8. A function $f \in \mathbb{B}_n$ is called *strongly k-mixed*, for $k < n$, if for an arbitrary k-subset $V \subset \{x_1, x_2, \ldots, x_n\}$, all nontrivial linear combinations

$$\sum_{\substack{\alpha \text{ is an} \\ \text{assignment to } V}} \beta_\alpha \cdot f\,|_\alpha \qquad (\beta_\alpha \in \{0, 1\})$$

essentially depend on any variable taken from the set $\{x_1, x_2, \ldots, x_n\} \setminus V$.

Corollary 1.9 (Lower Bound Criterion)**.** *Let $f(x_1, x_2, \ldots, x_n)$ be a strongly k-mixed function, and let B be a well-structured graph driven $\oplus$-FBDD representing f. Then the number of nodes of B is greater than or equal to 2^k.*

Proof. Let B be any well-structured graph driven $\oplus$-FBDD guided by G representing f, let u be any level at distance $k+1$ from the source, and let v be the level covered by u.

If π_1, π_2, ..., π_ν are the paths leading from s to u in G, then it suffices to show, that

$$\#\mathcal{N}_u = \dim \mathbb{B}_u(f)/\mathbb{B}_v(f) \geq \nu.$$

For deriving a contradiction, let us assume that this is not the case. Then there is a nontrivial linear combination

$$\phi := \sum_{i=1}^{\nu} \beta_i \cdot f\,\big|_{\alpha(\pi_i)} \in \mathbb{B}_v(f) \qquad (\beta_i \in \{0,1\}),$$

where $\alpha(\pi_i)$ is the partial assignment to $\{x_1, x_2, \ldots, x_n\}$ canonically associated with the path π_i $(i = 1, 2, \ldots \nu)$. In particular, ϕ does not essentially depend on the variable with which u is labeled. Contradiction to the fact, that f is a strongly k-mixed function. □

We use Corollary 1.9 to prove exponential lower bounds on the size of graph-driven $\oplus$-FBDDs for characteristic functions of linear codes.

A *linear code* C is a linear subspace of $\mathbb{F}_2^n$. We consider the *characteristic function* $f_C : \mathbb{F}_2^n \to \{0,1\}$ defined by $f_C(a) = 1 \iff a \in C$. The *Hamming distance* of two code words $a, b \in C$ is defined to be the number of 1's of $a \oplus b$. The *minimal distance* of a code C is the minimal Hamming distance of two distinct elements of C. The *dual* $C^\perp$ is the set of all vectors b such that $b^T a = 0$, for all elements $a \in C$. (Here $b^T a = b_1 a_1 \oplus \ldots \oplus b_n a_n$ is the standard inner product with respect to $\mathbb{F}_2$.) A set $D \subseteq \mathbb{F}_2^n$ is defined to be *k-universal*, if for any subset of k indices $I \subseteq \{1, \ldots, n\}$ the projection onto these coordinates restricted to the set D gives the whole space $\mathbb{F}_2^k$.

The next lemma is well-known. See [11] for a proof.

Lemma 1.10. *If C is a code of minimal distance $k+1$, then its dual $C^\perp$ is k-universal.*

We shall prove a general lower bound on the size of graph-driven $\oplus$-FBDDs representing f_C.

Theorem 1.11. *Let $C \subseteq \mathbb{F}_2^n$ be a linear code of minimal distance d whose dual $C^\perp$ has minimal distance $d^\perp$. Then any well-structured graph-driven $\oplus$-FBDD representing f_C has size greater than or equal to $2^{\min\{d,d^\perp\}-2}$.*

Proof. Let $k = \min\{d, d^\perp\} - 2$. By Corollary 1.9 is suffices to show that the function f_C is strongly k-mixed. Without loss of generality, we assume the set V of Definition 1.8 to be $\{x_1, \ldots, x_k\}$ and the additional variable to be x_{k+1}. Let $A = \{\alpha_1, \alpha_2, \ldots \alpha_{2^k}\}$ be the set of all assignments of constants to the variables

$x_1, \dots, x_k$. We have to show that each nontrivial linear combination

$$\sum_{i \in I} f_C(\alpha_i, x_{k+1}, x_{k+2}, \dots, x_n) \quad (\emptyset \subset I \subseteq \{1, 2, \dots, 2^k\})$$

essentially depends on any variable taken from $\{x_{k+1}, x_{k+2}, \dots, x_n\}$.

Since the distance $d^\perp$ of the dual code $C^\perp$ is greater than k, by Lemma 1.10 the code C is k-universal. Consequently, there are assignments $\beta_1, \beta_2, \dots, \beta_{2^k}$ to the variables $x_{k+1}, x_{k+2}, \dots, x_n$ such that

$$f_C(\alpha_i, \beta_i) = 1, \text{ for } i = 1, 2, \dots, 2^k.$$

For $i = 1, 2, \dots, 2^k$, let γ_i be the assignment to the variables $x_{k+1}, x_{k+2}, \dots, x_n$ defined as follows:

$$\gamma_i(x_j) := \begin{cases} \beta_i(x_j) \oplus 1 & \text{if } j = k+1; \\ \beta_i(x_j) & \text{if } j = k+2, k+3, \dots, n. \end{cases}$$

Since the distance of the code C is greater than or equal to $k+2$, we have

$$\begin{aligned} f_C(\alpha_i, \gamma_j) &= 0, \text{ for } i, j = 1, 2, \dots, 2^k; \\ f_C(\alpha_i, \beta_j) &= 0, \text{ for } i, j = 1, 2, \dots, 2^k, \ i \neq j. \end{aligned}$$

For the sake of deriving a contradiction, let us assume, that there is a nonempty subset $I \subseteq \{1, 2, \dots, 2^k\}$ such that

$$\bigoplus_{i \in I} f_C(\alpha_i, x_{k+1}, x_{k+2}, \dots, x_n) = g,$$

where the Boolean function g does not essentially depend on the variable x_{k+1}. Let us fix an index $j \in I$. Then $g(\beta_j) = 1$, and consequently $g(\gamma_j) = 1$. The latter equation implies, that there is an index $i \in I$, such that $f_C(\alpha_i, \gamma_j) = 1$. Contradiction. □

In order to prove an *explicit lower bound*, recall that the r-th order binary Reed-Muller code $R(r, l)$ of length $n = 2^l$ is the set of graphs of all polynomials in l variables over $\mathbb{F}_2$ of degree at most r. The code $R(r, l)$ is linear and has minimal distance 2^{l-r}. It is known that the dual of $R(r, l)$ is $R(l - r - 1, l)$ (see [12]).

Corollary 1.12. *Let $n = 2^l$ and $r = \lfloor l/2 \rfloor$. Then every well-structured graph-driven $\oplus$-FBDD representing the characteristic function of $R(r, l)$ has size bounded below by $2^{\Omega(\sqrt{n})}$.*

Proof. Taking the notation of Theorem 1.11, we have $d = 2^{l-r} = \Omega(\sqrt{n})$ and $d^\perp = 2^{r+1} = \Omega(\sqrt{n})$. The claim follows. □

Corollary 1.13. *Let $n = 2^l$ and $r = \lfloor l/2 \rfloor$. Then any graph-driven $\oplus$-FBDD guided by G representing the characteristic function of $R(r, l)$ has size bounded below by $2^{\Omega\left(\sqrt{n}\right)}/\mathrm{SIZE}\,(G)$.*

2. Minimizing the number of nodes

Let us define *a feasible exponent ω of matrix multiplication* over a field k to be a real number such that multiplication of two square matrices of order h may be algorithmically achieved with $\mathcal{O}\,(h^{\omega})$ arithmetical operations. It is well-known that matrix multiplication plays a key role in numerical linear algebra. Thus the following problems all have "exponent" ω: matrix inversion, L-R-decomposition, evaluation of the determinant. Up to now, the best known ω is 2.376 (see [7]). For practical reasons it might be best to use Gaussian elimination. Then we work with the feasible matrix exponent 3.

It is the aim of this section to prove the following theorem:

Theorem 2.1. *Let ω be any feasible exponent of matrix multiplication. Let G be a fixed graph ordering on the set of Boolean variables $\{x_1, \ldots, x_n\}$. Then there is an algorithm that computes taking a well-structured graph-driven $\oplus$-FBDD B guided by G as input a size-minimal one representing the same Boolean function as B in time $\mathcal{O}\,(\mathrm{SIZE}\,(G) \cdot \mathrm{SIZE}\,(B)^{\omega})$ and space $\mathcal{O}\left(\mathrm{SIZE}\,(G) + \mathrm{SIZE}\,(B)^2\right)$.*

Proof. Let $f(x_1, \ldots, x_n)$ be the Boolean function represented by the input BDD B. We assume B to be represented in an object-oriented way, where each node and each edge of the abstract diagram B is given by its own object.

For the sake of simplifying notations, we denote an abstract node, its resulting function (see Eq. (4)), and the corresponding node-object by the same letter. Moreover, we identify the nodes of G with the levels of B indexed.

The algorithm that proves the theorem falls into two phases. The first phase, which we call the *linear reduction phase*, insures that the functions represented by nodes of B are linearly independent, if they belong to a space $\mathbb{B}_w(B)$, for $w \in G$. We use a bottom-up approach here, where we refer to the direction of the graph ordering B. The second phase, called the *semantic reduction phase*, transforms the input B in a top-down manner such that afterwards the spaces $\mathbb{B}_w(B)$ are subspaces of the space $\mathbb{B}_w(f)$ ($w \in G$). Having performed these two phases, the BDD B thus modified is size-minimal by Theorem 1.7.

Each phase consists of several global steps each of which associated with a level w of the current version of B. As a rule, in such a global step the level w of B is modified. At the very beginning of this modification we internally represent B as two $\mathrm{SIZE}\,(B) \times \mathrm{SIZE}\,(B)$-adjacency matrices $A^{(0)}$ and $A^{(1)}$ over $\mathbb{F}_2$, where the columns of $A^{(b)}$ represent the b-successor sets of B's branching nodes, and a column vector R over $\mathbb{F}_2$ of length $\mathrm{SIZE}\,(B)$ that represents the successor set of the source s. These matrices can be set up in linear time. In line with that, all auxiliary subsets of non-source nodes are represented as column-vectors of length $\mathrm{SIZE}\,(B)$. Again, we use the same letter for the set and the vector.

We describe the linear reduction phase of the algorithm. Assume that we are about to linearly reduce the level w, where all levels that can be reached from w in G by a nontrivial directed path have already been linearly reduced. Let x be the Boolean variable with which w is labeled, for $b \in \{0,1\}$, let w_b be the b-successor of w in G, and let w' be the level covered by w in the level poset of G.

We want to linearly reduce the level w by means of a matrix representation M of the set $\bigcup_{v\leq w}\mathcal{N}_v(B)$, which is different from the vector representation mentioned above. The matrix M is computed on the basis of the mapping

$$\begin{array}{rcl} \mathbb{B}_w(B) & \rightarrow & \mathbb{B}_{w_0}(B)\times\mathbb{B}_{w_1}(B) \\ v & \mapsto & (v|_{x=0}, v|_{x=1}) \end{array}$$

(see Eq. (10)). There are two cases to distinguish.

Case 1. The two levels w_0 and w_1 coincide. Then $w' = w_0 = w_1$. We have to proceed exactly in the same way as in the case of $\oplus$-OBDDs (see [15]). We omit this.

Case 2. The levels w_0 and w_1 are different. Then both w_0 and w_1 cover w' in the level poset and we can apply equation (7) as follows. Let $u_1,\ldots,u_\mu$ be the nodes that span the space $\mathbb{B}_{w'}(B)$, let $u_{\mu+1},\ldots,u_{\mu+\mu'}$ be the nodes of level w, and let $v_1^0,\ldots,v_{\nu_0}^0$ and $v_1^1,\ldots,v_{\nu_1}^1$ be the nodes that span the spaces $\mathbb{B}_{w_0}(B)$ and $\mathbb{B}_{w_1}(B)$, respectively. For $i=1,2,\ldots,m$, we can assume then that $u_i = v_i^0 = v_i^1$.

To define the matrix M, let $\langle .,. \rangle$ denote the inner product in both spaces $\mathbb{B}_{w_0}(B)$ and $\mathbb{B}_{w_1}(B)$ defined by $\left\langle v_i^b, v_j^b\right\rangle = \delta_{ij}$, for $i,j=1,\ldots,\nu_b$, and $b=0,1$. Then the matrix $M=(M_{ij})$, where $i=1,\ldots,\nu_0+\nu_1$, $j=1,\ldots,\mu+\mu'$ is defined as follows. For $i=1,\ldots,\nu_0$, $j=1,\ldots,\mu+\mu'$, let $M_{ij} := \left\langle v_i^0, u_j|_{x=0}\right\rangle$, and for $i=1,\ldots,\nu_1$, let $M_{\nu_0+i,j} := \left\langle v_i^1, u_j|_{x=1}\right\rangle$. The columns $M_{\cdot 1}, \ldots, M_{\cdot\mu}$ represent the canonical basis of the space $\mathbb{B}_{w'}(B)$, the columns $M_{\cdot\mu+1}, \ldots, M_{\cdot\mu+\mu'}$ the nodes of level w.

In *the first step* this matrix M is set up. This can be done by traversing the graph B.

The second step is to find out which of the nodes of level w can be eliminated. To this end, we select columns $M_{\cdot\mu+j}$, $j\in J$, such that the columns $M_{\cdot 1},\ldots,M_{\cdot\mu},M_{\cdot\mu+j}$, $j\in J$, form a basis of the space spanned by all columns of M. We then represent the columns not selected in terms of those selected. We assume the result of the second step to be presented as follows.

$$\begin{aligned} M_{\cdot\mu+l} &= \sum_{k=1}^{\mu}\alpha_k M_{\cdot k} + \sum_{j\in J}\alpha_{\mu+j}M_{\cdot\mu+j}, \text{ for } l\notin J, \\ U_{\mu+l} &:= \{u_j \mid \alpha_j = 1\}, \text{ for } l\notin J. \end{aligned}$$

The third step is to hardwire the results of the second step in the decision diagram B. The problem is, to do so within the desired time bound. Having set up the matrix H resulting from the SIZE $(B)\times$ SIZE (B)-identity matrix by replacing the columns associated with $u_{\mu+l}$, for $l\notin J$, by the columns $U_{\mu+l}$, we execute the following three instructions.

$$\begin{aligned} A^{(b)} &\leftarrow H \cdot A^{(b)}, \text{ for } b = 0, 1, \\ R &\leftarrow H \cdot R. \end{aligned}$$

The first and the second one update the branching nodes of B, the third one the source of B. Afterwards the nodes $u_{\mu+l}$, for $l \notin J$, have indegree zero.

In the last step we remove all nodes no longer reachable from a source. This can be done by a depth-first-search traversal.

Now we describe the semantic reduction phase of the algorithm. It suffices to consider the case, that B has a nonempty level different from s and t. By Theorem 1.7 the BDD B has then a uniquely determined nonempty top level w_{top} which is covered by G's source in the level poset. Clearly, the level w_{top} is joined to any other nonempty level by a directed path in G.

As in the case of the linear reduction phase, we assume that we are on the point of semantically reducing level w, where all levels that precede w in a fixed topological ordering of the nodes of G have already been reduced.

Case 1. The level w is equal to w_{top}. We have to transform w in such a way that it contains afterwards a single node only. Let $v_1, \ldots, v_\mu$ be the node of level w that can be reached from the source of B by a directed edge. We merge these nodes in such a way together that for the resulting node u holds: $u = \bigoplus_{i=1}^{\mu} v_i$. For $b = 0, 1$, the b-successor set of node u is computed by executing the matrix operation

$$V^{(b)} \leftarrow A^{(b)} \cdot M + L,$$

where here $M = \{v_1, \ldots, v_\mu\}$ and L is the set of all nodes that can be reached from the source of B by a directed edge and that do not belong to the level w. Having computed this, the new node u can be created and connected to the nodes of $V^{(b)}$ by an b-edge, for $b = 0, 1$.

Case 2. The level w is not equal to w_{top}.

In the first step of this case for each node u of B and each Boolean constant b such that there is an b-edge leading from u to a node of level w we do the following. We partition the b-successors of u into the three sets O_u^b, M_u^b, and L_u^b, where M_u^b contains all nodes belonging to level w, L_u^b contains all nodes belonging to levels that are properly less than w in the level poset, and O_u^b is the remaining set. (By induction hypothesis we have already semantically reduced the levels to which the nodes of O_u^b belong.) Having created a new node $v(u, b)$ of level $\mathcal{N}_w(B)$, we remove all b-edges from u to nodes of $M_u^{(b)} \cup L_u^{(b)}$, and join u to $v(u, b)$ by a directed b-edge. In order to compute the edges outgoing from $v(u, b)$ in such a way that $v(u, b) = \bigoplus_{v \in M_u^{(b)} \cup L_u^{(b)}} v$ holds, we set up two matrices M and L. The columns of M and L are the sets $M_u^{(b)}$ and $L_u^{(b)}$, respectively, we have just created. Then we compute for each node $v(u, b)$ its successor sets $\text{Succ}_0\,(v(u, b))$ and $\text{Succ}_1\,(v(u, b))$

by means of the matrix operations

$$V^{(b)} \leftarrow A^{(b)} \cdot M \, + \, L, \text{ for } b = 0, 1.$$

(Because of the induction hypothesis on the sets O_u^b, we are sure that $v(u, b) \in \mathbb{B}_w(f)$, for all u and b under consideration.)

In the second step we remove all nodes of B that are no longer reachable from the source.

The new nodes of level w are not necessarily linearly independent from each other and from other nodes belonging to a level less than or equal to w in the level poset of G. *In the last step* we linearly reduce the space $\mathbb{B}_w(B)$ in the same way as in the first phase.

The runtime of the algorithm is dominated by the $\mathcal{O}(|G|)$ multiplications of $\text{SIZE}(B) \times \text{SIZE}(B)$-matrices. The space demand is obvious. □

Let B' and B'' be two graph-driven free parity BDDs on $\{x_1, \ldots, x_n\}$ guided by G.

First, using standard techniques, for example the well-known "product construction", and taking pattern from [14], one can easily perform the Boolean synthesis operations in time $\mathcal{O}(\text{SIZE}(G) \cdot (\text{SIZE}(B') \cdot \text{SIZE}(B''))^\omega)$.

Second, we have the following:

Corollary 2.2. *It can be decided in time* $\mathcal{O}(\text{SIZE}(G) \cdot (\text{SIZE}(B') + \text{SIZE}(B''))^\omega)$ *whether or not* B' *and* B'' *represent the same function.*

Acknowledgements. The authors are indebted to Beate Bollig and the referees for their helpful suggestions and comments.

References

[1] B. Bollig, St. Waack and P. Woelfel, Parity graph-driven read-once branching programs and an exponential lower bound for integer multiplication, in *Proc. 2nd IFIP International Conference on Theoretical Computer Science* (2002).

[2] Y. Breitbart, H.B. Hunt and D. Rosenkrantz, The size of binary decision diagrams representing Boolean functions. *Theoret. Comput. Sci.* **145** (1995) 45-69.

[3] H. Brosenne, M. Homeister and St. Waack, Graph-driven free parity BDDs: Algorithms and lower bounds, in *Proc. 26th MFCS*. Springer Verlag, *Lecture Notes in Comput. Sci.* **2136** (2001) 212-223.

[4] R. Bryant, On the complexity of VLSI implementations of Boolean functions with applications to integer multiplication. *IEEE Trans. Comput.* **40** (1991) 205-213.

[5] R.E. Bryant, Symbolic manipulation of Boolean functions using a graphical representation, in *Proc. 22nd DAC*. Piscataway, NJ (1985) 688-694.

[6] R.E. Bryant, Graph-based algorithms for Boolean function manipulation. *IEEE Trans. Comput.* **35** (1986) 677-691.

[7] D. Coppersmith and S. Winograd, Matrix multiplication *via* arithmetic progressions. *J. Symb. Comput.* **9** (1990) 251-280.

[8] J. Gergov and Ch. Meinel, Frontiers of feasible and probabilistic feasible Boolean manipulation with branching programs, in *Proc. 10th STACS*. Springer Verlag, *Lecture Notes in Comput. Sci.* **665** (1993) 576-585.

[9] J. Gergov and Ch. Meinel, Mod-2-OBDDs – A data structure that generalizes exor-sum-of-products and ordered binary decision diagrams. *Formal Methods in System Design* **8** (1996) 273-282.

[10] S. Jukna, Entropy of contact circuits and lower bounds on their complexit. *Theoret. Comput. Sci.* **57** (1988) 113-129.

[11] S. Jukna, Linear codes are hard for oblivious read-once parity branching programs. *Inform. Process. Lett.* **69** (1999) 267-269.

[12] E.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes.* Elsevier (1977).

[13] D. Sieling, Lower bounds for linear transformed OBDDs and FBDDs, in *Proc. 19th FSTTCS*. Springer Verlag, *Lecture Notes in Comput. Sci.* **1738** (1999) 356-368.

[14] D. Sieling and I. Wegener, Graph driven BDDs – A new data structure for Boolean functions. *Theoret. Comput. Sci.* **141** (1995) 238-310.

[15] St. Waack, On the descriptive and algorithmic power of parity ordered binary decision diagrams. *Inform. Comput.* **166** (2001) 61-70.

[16] I. Wegener, *Branching Programs and Binary Decision Diagrams – Theory and Applications*. SIAM, Philadelphia, *SIAM Monogr. Discrete Math. Appl.* (2000).

Communicated by H. Hromkovic.
Received November, 2001. Accepted September, 2002.