Electronic Colloquium on Computational Complexity, Report No. 68 (2003)

Lower Bounds for the Sum of Graph–driven Read–Once Parity Branching Programs

Matthias Homeister*

Institut für Numerische und Angewandte Mathematik
Georg–August–Universität Göttingen
Lotzestr. 16–18, 37083 Göttingen, Germany
homeiste@math.uni-goettingen.de

Abstract. We prove the first lower bound for restricted read–once parity branching programs with unlimited parity nondeterminism where for each input the variables may be tested according to several orderings.

Proving a superpolynomial lower bound for read–once parity branching programs is still a challenging open problem. The following variant of read–once parity branching programs is well–motivated. Let k be a fixed integer. For each input a there are k orderings $\sigma_1(a), \ldots, \sigma_k(a)$ of the variables such that for each computation path activated by a the bits are queried according to $\sigma_i(a)$ for some $i, 1 \leq i \leq k$. This model that we call k–$\oplus$BP1s for convenience strictly generalizes all restricted variants of read–once parity branching programs for that lower bounds are known. We consider a slightly more restricted version, i.e. the sum of k graph–driven $\oplus$BP1s with polynomial size graph–orderings. We prove lower bounds for linear codes and show that the considered variant strictly generalizes well–structured graph–driven $\oplus$BP1s as well as $(\oplus, k)$-BPs examined by Savický and Sieling in [24].

Keywords: read–once parity branching programs, lower bounds, computational complexity.

1 Introduction

Parity branching programs are a model of sequential space bounded computation, where the parity representation mode is used. Recently, exponential lower bounds for graph–driven read–once parity branching programs have been proved (see [8], [4] and [5] and [15] for a restricted variant called *well–structured* and [9] for general ones). In this paper the first lower bounds are proven for a read–once parity branching programs with restricted parity nondeterminism possibly testing the variables in several orderings for each assignment.

A *branching program* (BP for short) $\mathcal{B}$ on the set of Boolean variables $\{x_1, \ldots, x_n\}$ is a directed acyclic graph with one source and one target. The outdegree of the target and the indegree of the source are both equal to zero. The source is joined to its successors by unlabeled directed edges. The nodes different from the source and the target, the so–called *branching nodes*, are labeled with Boolean variables and the outgoing edges are labeled with 1 or with 0.

The *size* of a BP $\mathcal{B}$ denoted by SIZE $(\mathcal{B})$ or by $|\mathcal{B}|$ is the number of its nodes.

A branching program is called *deterministic* if the source has exactly one successor, and each branching node is left by not more than one 0- and one 1-edge.

An input $a \in \{0,1\}^n$ *activates* all edges labeled with a_i outgoing from nodes labeled with x_i, for $i = 1, 2, \ldots, n$. 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 BP $\mathcal{B}$ on $\{x_1, \ldots, x_n\}$ is a path in $\mathcal{B}$ from the source whose edges are activated by a. Such a path is called an *accepting* one, if it leads to the target.

A *parity branching program* ($\oplus$–BP for short) is a branching program equipped with the *parity representation mode.* It 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*.

* Supported by DFG grant Wa 766/4-2

ISSN 1433-8092

A *nondeterministic branching program* ($\vee$–BP for short) uses the common nondeterministic representation mode. An input $a \in \{0,1\}^n$ is accepted if and only if there is an accepting computation path under a.

If a branching program is deterministic, then the above mentioned representation modes coincide.

The best known lower bound on the size of unrestricted deterministic BPs is of order $\Omega\left(\frac{n^2}{(\log n)^2}\right)$. It was proved by Nechiporuk [22] in 1966. Consequently, restricted models have been studied intensively. (See [29] for an overview.) Here we can only review those results closely related to ours. Nevertheless, the breakthroughs for read–k–times BPs due to Borodin, Razborov, and Smolensky (see [6]) and Thathachar (see [27]), and for semantic super–linear length BPs due to Ajtai, Beame, Saks, Sun, and Vee (see [1], [2], and [3]) must be mentioned.

A branching program is called *read–once* (BP1 for short) if on every path from the source to the target each variable is tested at most once.

Ordered binary decision diagrams (OBDDs), introduced by Bryant ([10], [11]), are deterministic BP1s with the following additional property. There is a permutation σ of the set $\{1, 2, \ldots, n\}$ such if node v labeled with $x_{\sigma(j)}$ is a successor of node u labeled with $x_{\sigma(i)}$, then $i > j$. As for proving lower bounds, the existence of a global variable ordering ensures that one can proceed as follows. Having put a cut through a σ–OBDD representing f at distance of say k from the source, the number of distinct subfunctions $f|_\pi$, where π ranges over all paths from the source to the frontier nodes of the cut, is a lower bound on the σ–OBDD size of f.

OBDDs are highly restricted branching programs. Many even simple functions have exponential OBDD–size (see [7], [12]).

To maintain the essence of the above subfunction argument for more general models, the following observation is useful. If $\mathcal{B}$ is a deterministic BP1 on $\{x_1, x_2, \ldots, x_n\}$, then for each input $a \in \{0,1\}^n$ there is a variable ordering $\sigma(a)$ according to which the bits of a are queried. But not every combination of variable orderings can be implemented by deterministic BP1s. Only those resulting from *graph orderings*, independently introduced by Gergov and Meinel (see [13]) and Sieling and Wegener (see [26]), are possible.

Definition 1. *A* graph ordering *G is a deterministic BP1 such that each branching node has outdegree two, and each variable is tested on each path from the source to the target* exactly *once.*

A BP1 $\mathcal{B}$ is called a graph–driven *one guided by a graph ordering G over the same set of variables as $\mathcal{B}$, if the following condition is satisfied.*

For an arbitrary input $a \in \{0,1\}^n$, the list of variables inspected on every computation path for a in $\mathcal{B}$ is a subsequence of the corresponding list resulting from G.

For every deterministic BP1 $\mathcal{B}$, it is easy to construct a graph ordering G that guides $\mathcal{B}$. But it is clear that there are BP1s that are not guided by a graph ordering. Of course, OBDDs are graph–driven deterministic BP1s. A more general example is given in Figure 1.

$\oplus$-OBDDs were introduced by Gergov and Meinel in [14], they have been intensively studied in [28] from a theoretical point of view. Heuristics for a successful practical implementation are due to Meinel and Sack (see [23], [20], [21]). Examples of functions showing that $\oplus$-OBDDs are more powerful than OBDDs are given in [14].

Graph–driven $\oplus$–BP1s have a strictly larger descriptive power than both deterministic BP1s and $\oplus$-OBDDs with respect to polynomial size. This follows from results due to Sieling [25].

Up to now, proving superpolynomial lower bounds on the size of $\oplus$–BP1s is a challenging open problem in complexity theory.

The notion of well–structured graph–driven BP1s was introduced in [26].

Definition 2. *A graph–driven $\oplus$BP1 is called* well-structured *if there is a function* level *mapping from the nodes of $\mathcal{B}$ to the nodes of the ordering G in the following way. For any node v that under an input is traversed on a path in $\mathcal{B}$, in G the node* level(v) *is traversed under this input and is labeled with the same variable.*

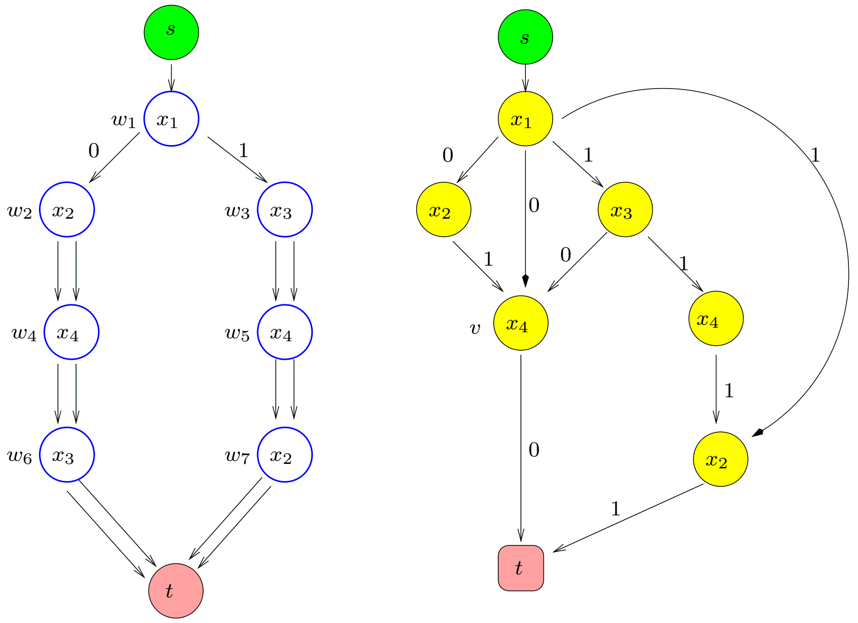

Fig. 1. A graph ordering and a graph–driven BP1 guided by this ordering.

In [8] exponential lower bounds of magnitude $2^{\Omega(\sqrt{n})}$ on the size of *well–structured* graph–driven $\oplus$–BP1s for certain linear code functions have been proved. Well–structured $\oplus$–BP1s and $\vee$–BP1s have been further investigated in [4] and [5]. In [4] a strongly exponential lower bound for integer multiplication is proved. In [5] polynomial size well–structured $\oplus$–BP1s are separated from polynomial size general $\oplus$–BP1s. In [15] the method for proving lower bounds has been simplified and further lower bounds have been proved.

The first lower bounds for graph–driven $\oplus$BP1s without the restriction being well–structured have been presented in [9]. Moreover, the following characterization of all graph–driven $\oplus$BP1s that are graph–driven has been proved there.

Proposition 1. *Let $\mathcal{B}$ be a $\oplus$–BP1 on the set of variables $\{x_1, x_2, \ldots, x_n\}$. Then there exists a graph–ordering G such that $\mathcal{B}$ is guided by G if and only if the following condition is satisfied. For each input a there is an ordering $\sigma(a)$ of $\{x_1, x_2, \ldots, x_n\}$ such that on each computation path for a the bits of a are queried according to $\sigma(a)$.*

Thus interest in the following natural generalization arises.

Definition 3. *Let k be any positive integer. A k–$\oplus$*BP1 *is a $\oplus$BP1 with the following additional restriction. For each input a there are not more than k variable orderings $\sigma_1(a), \ldots, \sigma_k(a)$ such that on each computation path for a the bits of a are queried according to $\sigma_i(a)$ for some i, $1 \leq i \leq k$.*

By means of this proposition it gets plain that the computational power of graph–driven $\oplus$BP1s equals that of 1–$\oplus$BP1s. To express this, for any branching program model M, let the set $\mathcal{P}(M)$ consist of all sequences of Boolean functions that can be represented by a branching program of type M of polynomially bounded size.

Proposition 2. *$\mathcal{P}$(graph–$\oplus$BP1)$=$ $\mathcal{P}$(1–$\oplus$BP1).*

Next we observe that in terms of computational power 2–$\oplus$BP1s strictly generalize graph–driven $\oplus$BP1s. In [9] it has been proved that each graph–driven $\oplus$BP1 representing the function $\mathbb{1}_{\mathrm{C}}^{n} \vee \mathbb{1}_{\mathrm{R}^+}^{n-1,1}$ has exponential size, where

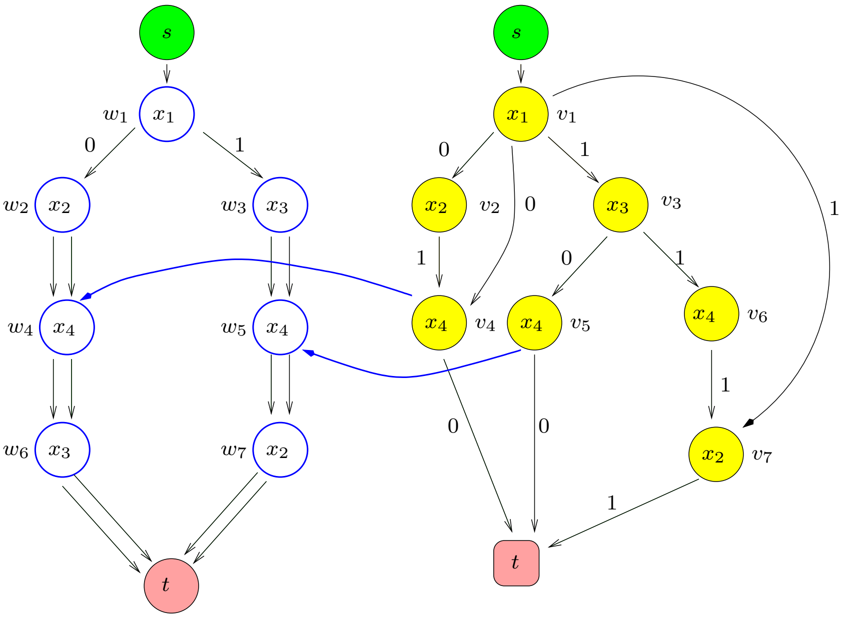

Fig. 2. A well–structured graph–driven ⊕BP1 guided by a graph ordering.

$$\mathbb{1}_{\mathrm{C}}^{n} = \begin{cases} 1 & \text{if each column of } X \text{ contains exactly one 1;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbb{1}_{\mathrm{R}^{+}}^{n-1,1} = \begin{cases} 1 & \text{if } n-1 \text{ rows of } X \text{ contain exactly one 1} \\ & \text{and one row contains two 1s;} \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, a ⊕BP1 is constructed which represents this function succinctly. That ⊕BP1 is in fact a 2–⊕BP1, since it is constructed by joining two OBDDs to a source of fanout 2. One OBDD tests the variables in a rowwise and the other one in a columnwise manner. We state this observation as

Corollary 1. *$\mathcal{P}(1$–⊕BP1$)$ is a proper subset of $\mathcal{P}(2$–⊕BP1$)$.*

Definition 4. *A ⊕BP1 $\mathcal{B}$ is a* sum of k graph–driven ⊕BP1s *driven by a sequence of graph–orderings $G = (G_1, \ldots, G_k)$, if $\mathcal{B}$ consists of k disjoint ⊕BP1s $\mathcal{B}_1, \ldots, \mathcal{B}_k$ joined to a common source such that for each i, $\mathcal{B}_i$ is guided by G_i.*

We call such a $\mathcal{B}$ a G–driven sum of graph–driven ⊕BP1s*.*

In the next section we present a lower bound method for sums of graph–driven ⊕BP1s in order to prove lower bounds for this model with the additional restriction that the orderings have polynomial size, i.e. $|G| = |G_1| + \ldots + |G_k| = n^{\mathcal{O}(1)}$. To explain the connection to well–structured graph–driven ⊕BP1s we prove the following observation.

Proposition 3. *A Boolean function f is representable by a polynomial–sized well–structured graph–driven ⊕BP1 $\mathcal{B}$ if and only if f is representable by a polynomial–sized graph–driven ⊕BP1 $\mathcal{B}'$ guided by a polynomial–sized ordering G.*

Proof. First we have to show that the condition is necessary. Let $\mathcal{B}'$ be guided by G given. To transform $\mathcal{B}'$ into a well–structured graph–driven ⊕BP1 one has to rebuild $\mathcal{B}'$ such that a level function as claimed in

Definition 2 can be chosen. This is tractable by multiplying some nodes and so assigning them to different levels. Since the number of levels is less or equal to SIZE (G) there is a well–structured graph–driven $\oplus$BP1 guided by G with SIZE $(\mathcal{B}) \leq$ SIZE $(\mathcal{B}') \cdot$ SIZE (G).

Now assume that the condition is fulfilled. In [4] the following is shown. Given a polynomial–sized well–structured graph–driven $\oplus$BP1 guided by an ordering. Then there is a ordering G such that $\mathcal{B}$ is guided by G with SIZE $(G) \leq 2 \cdot n \cdot$ SIZE $(\mathcal{B})$ and the condition being well–structured is fulfilled. The claim follows.

The following propositions state the connection between sums of graph–driven $\oplus$BP1s driven by an ordering G of polynomial size and well–structured graph–driven $\oplus$BP1s. Both of them are direct consequences of Proposition 3. By $\mathcal{P}(k^*\text{–}\oplus\text{BP1})$ we denote all functions representable by polynomial size sums of graph–driven $\oplus$BP1s guided by a sequence of graph–orderings $G = (G_1, \ldots, G_k)$ with $|G| = n^{\mathcal{O}(1)}$.

Proposition 4. *$\mathcal{P}$(wsGraph–$\oplus$BP1) = $\mathcal{P}(1^*$–$\oplus$BP1).*

For the next proposition we have to observe the following. The 2–$\oplus$BP1 for $\mathbb{1}_{\mathrm{C}}^{n} \vee \mathbb{1}_{\mathrm{R}+}^{n-1,1}$ constructed in the context of Corollary 1, is guided by two graph–orderings of polynomial size, since it is constructed by joining two OBDDs.

Proposition 5. *$\mathcal{P}$(wsGraph–$\oplus$BP1) is a proper subset of $\mathcal{P}(2^*$–$\oplus$BP1).*

We conclude that the notion of a $(G_1, \ldots, G_k)$–driven sum of graph–driven $\oplus$BP1s with polynomial size graph-orderings is a natural restriction.

In [24] exponential lower bounds for pointer functions on the size of $(\oplus, k)$–BP1s are proved. A $(\oplus, k)$–BP1 is a read–once BP with the source being the only nondeterministic node, where k denotes the fan–out of the source. We prove that our model strictly generalizes $(\oplus, k)$–BPs. By $\mathcal{P}((\oplus, k)\text{–BP1})$ we denote the set of functions representable by polynomial size $(\oplus, k)$–BPs. First we observe that each $(\oplus, k)$–BP $\mathcal{B}$ can be considered as a sum of k graph–driven $\oplus$BP1s guided by itself. So we can construct a sequence of k graph–orderings driving $\mathcal{B}$ of the same size as $\mathcal{B}$, and conclude $\mathcal{P}((\oplus, k)\text{–BP1}) \subseteq \mathcal{P}(k^*\text{–}\oplus\text{BP1})$. To see that this containment is proper consider the functions f_n^k defined on the variables $X = \{x_0, \ldots, x_{n-1}\}$ that are examined in [24]. The set X is partitioned in $k(k+1)$ blocks $B_{i,j}, 1 \leq i \leq k+1, 1 \leq j \leq k$, and if necessary, some remaining variables. Each block $B_{i,j}$ consists of $\log n$ subblocks of size

$$s = \lfloor \frac{n}{k(k+1)\log m} \rfloor.$$

For our purpose we consider only the blocks $B_{1,1}, \ldots, B_{1,k}$. We have not changed the functions from [24], in order to facilitate checking the references. Each $B_{1,j}$ computes a binary representation of a pointer $p(j)$. Each of the $\log n$ bits of $p(j)$ is determined by the majority of the s bits in one of the $\log n$ subblocks of the block $B_{i,j}$. $f_n^k(x)$ outputs 1 if and only if all bits addressed by the pointers equal 1, i.e.

$$x_{p(1)} = x_{p(2)} = \ldots = x_{p(k)} = 1.$$

In [24] it is proved that f_n^k has no representation by polynomial size $(\oplus, k)$–BPs for $k \leq (1/2 - \gamma)\log n$ for some $\gamma > 0$. In the following we like to show that f_n^k can be represented by $\oplus$OBDDs of size $\mathcal{O}\left(n^{k+2}\right)$. The following algorithm computes f_n^k.

1. Guess the binary representation of the pointers $p_1, \ldots, p_k$.
2. Verify this choice and check, whether $x_{p(1)} = x_{p(2)} = \ldots = x_{p(k)} = 1$.

We illustrate step 2 for a certain guess. We test the variables according to an ordering, such that for each subblock of some $B_{1,j}$ all s variables are tested successively. If we read a bit x_i that is adressed by one of the guessed pointers the computation stops, or 0 is the output. Since each majority vote can be accomplished by $\mathcal{O}\left(s^2\right)$ nodes, step 2 describes an OBDD of size $\mathcal{O}\left(n^2\right)$. All n^k OBDDs of this kind can be constructed with a common ordering and in fact the algorithm stated as steps 1 and 2 describes a $\oplus$OBDD for f_n^k, since for each input step 2 accepts it if and only if the pointers are correct and all adressed bits equal 1.

Proposition 6. *For $k \leq (2/3)\log^{1/2} n$ it holds that $\mathcal{P}((\oplus, k)$–BP1) is a proper subset of $\mathcal{P}(k^*$–$\oplus$BP1).*

Proof. For k constant the claim follows immediately by the above presented construction of the $\oplus$OBDD of size $\mathcal{O}\left(n^{k+2}\right)$. For nonconstant k we are able to apply the same padding arguments that are used in [24] to prove Theorem 15 of that paper.

2 A lower bound criterion for sums of graph–driven $\oplus$BP1s

In more restricted models like deterministic BP1s, $\oplus$OBDDs or graph–driven $\oplus$BP1s (1–$\oplus$BP1s, resp.) the nodes or sets of nodes reached by certain partial assignments represent subfunctions of the function represented by the whole diagram. This is not the case for sums of graph–driven $\oplus$BP1s, but certainly there is *some* connection between the functions represented by the nodes and the function represented by the whole diagram. . First we collect some notation. For a partial assignment α to some variables, the *subfunction* f_α, or $f|_\alpha$, results by setting all variables in $\mathrm{V}(\alpha)$ to the constants according to α. Sometimes it is more convenient to express f_α, or $f|_\alpha$, as $f(\alpha)$. A function f is called *essentially dependent* on the variable x_i, if different settings to this variable result in different subfunctions, i.e. $f_{x_i=0} \neq f_{x_i=1}$. For a partial assignment α the subfunction $f(\alpha)$ formally depends on all variables in X, but indeed is not essentially dependent on the variables set by α. The following definition forms the basis of our examinations.

Let $\mathcal{B}$ be a $\oplus$BP1 driven by a graph–ordering G. By $\mathbb{B}_G(f)$ we denote the span of all subfunctions $f\,|_\pi$, where π is a path from the source to a node w in G and $f\,|_\pi$ results from f by setting the variable according to the labels of the nodes and edges on π.

Let $\mathcal{B}$ be a sum of k graph–driven $\oplus$BP1s $\mathcal{B}_1, \ldots, \mathcal{B}_k$. Since $\mathrm{Res}(\mathcal{B}) = \mathrm{Res}(\mathcal{B}_1) + \ldots + \mathrm{Res}(\mathcal{B}_k)$ we are motivated to consider the direct sum of spaces $\mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k)$ for functions $g^1, \ldots, g^k$ with $g^1 + \ldots + g^k = f$.

Lemma 1. *Let $\mathcal{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_k)$ be a $(G_1, \ldots, G_k)$–driven sum of $\oplus$BP1s representing f. Then there are functions $g^1, \ldots, g^k$ with $f = g^1 + \ldots + g^k$ such that*

$$\mathrm{SIZE}\,(\mathcal{B}) \geq \dim_{\mathbb{F}_2}\Big(\mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k)\Big).$$

Proof. We define $\mathbb{B}(\mathcal{B}) = \mathrm{span}_{\mathbb{F}_2}\{\mathrm{Res}_v \mid v \in \mathcal{B}\}$. Observe that $\mathrm{SIZE}\,(\mathcal{B}) \geq \dim_{\mathbb{F}_2}\mathbb{B}(\mathcal{B})$. For $\mathcal{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_k)$ we set $g^1 = \mathrm{Res}(\mathcal{B}_1), \ldots, g^k = \mathrm{Res}(\mathcal{B}_k)$ and prove that $\mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k) \subseteq \mathbb{B}(\mathcal{B})$. Then the claim follows, since $g^1 + \ldots + g^k = \mathrm{Res}(\mathcal{B}) = f$.

Let $g^i\,|_\pi$ be any generating element of the vector space $\mathbb{B}_{G_i}(g^i)$ for some $i = 1, \ldots, k$, and let α be the partial assignment to the set of variables $\{x_1, x_2, \ldots, x_n\}$ associated with the path π in G_i. Since the branching program $\mathcal{B}_i$ is guided by the graph ordering G_i, we are led to nodes $v_1, v_2, \ldots, v_\nu$ when traversing $\mathcal{B}_i$ starting at the source according to the partial assignment α. Consequently, $g^i\,|_\pi = \sum_{j=1}^{\nu} \mathrm{Res}_{v_j}$, and so every generating element of $\mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k)$ is contained in $\mathbb{B}(\mathcal{B})$. The claim follows.

In order to apply this lemma as a lower bound criterion, we have to examine the spaces $\mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k)$ for all decompositions $f = g^1 + \ldots + g^k$ of f. For a special case this is done in Lemma 2. To describe the setting of that lemma, we need further notation.

We examine how to combine several partial assignments. For partial assignments $\alpha_1, \ldots, \alpha_\nu$ with pairwise disjoint domains $\mathrm{V}(\alpha_i), i = 1, \ldots, n$, we denote by $(\alpha_1, \ldots, \alpha_\nu)$ the assignment α defined on $\mathrm{V}(\alpha_1) \cup \ldots \cup \mathrm{V}(\alpha_\nu)$ as

$$\alpha(x_j) := \begin{cases} \alpha_1(x_j) & \text{if } \alpha_1(x_j) \text{ is defined;} \\ \vdots & \vdots \\ \alpha_\nu(x_j) & \text{if } \alpha_\nu(x_j) \text{ is defined.} \end{cases}$$

If the domains $\mathrm{V}(\alpha_i), i = 1, \ldots, n$ are not pairwise disjoint, it is required that for all $1 \leq i, j \leq \nu$ and for all $x_k \in \mathrm{V}(\alpha_i) \cap \mathrm{V}(\alpha_j)$, the assignments to x_k are equal for α_i and for α_j, i.e. $\alpha_i(x_k) = \alpha_j(x_k)$. Then the notion $\alpha = (\alpha_1, \ldots, \alpha_\nu)$ as defined above is well–defined. Now it is clear, that $\mathrm{V}(\alpha_1, \ldots, \alpha_\nu) = \bigcup_{i=1}^{\nu} \mathrm{V}(\alpha_i)$. By $\overline{\mathrm{V}(\alpha)}$ we denote the complement $\{x_1, \ldots, x_n\} - \mathrm{V}(\alpha)$.

Let $v = (v_1, \ldots, v_k)$ be in $G_1 \times \ldots \times G_k$. We denote by $V(v_i)$ the variables that are tested in G_i on a path from the source to v_i, excluding the variable tested in v_i. Let $\alpha_1, \ldots, \alpha_k$ be partial assignments such that α_i corresponds to a path from the source of G_i to v_i.

Definition 5. *Given a sequence of graph–orderings $G_1, \ldots, G_k$ and $v = (v_1, \ldots, v_k) \in G_1 \times \ldots \times G_k$, we call a tuple $(\alpha_1, \ldots, \alpha_k)$ of partial assignments a* v-assignment*, if*

- *for $1 \le i \le k$, α_i corresponds to the path from the source of G_i to v_i, and*
- *for $1 \le i, j \le k$, α_i equals α_j on $V(v_i) \cap V(v_j)$, i.e., $\alpha_i(x) = \alpha_j(x)$ for all x in $V(v_i) \cap V(v_j)$.*

We consider a v-assignment $\alpha = (\alpha_1, \ldots, \alpha_k)$ as an assignment defined on $\mathrm{V}(\alpha_1) \cup \ldots \cup \mathrm{V}(\alpha_k)$. An easy way getting v-assignments is truncating for some $a \in \{0,1\}^n$ the k paths in $G_1, \ldots, G_k$ simultanously.

Lemma 2. *Let $\mathcal{B}$ be a $(G_1, \ldots, G_k)$–driven sum of graph–driven ⊕BP1s representing f and let v be in $G_1 \times \ldots \times G_k$. For $i = 1, \ldots, k$ let A_i be some set of assignments to $\mathrm{V}(v_i)$, such that each α in $A_1 \times \ldots \times A_k$ is a v-assignment. Moreover, for all $\alpha \in A_1 \times \ldots \times A_k$ let there be some assignment δ, defined on the variables not set by α with*

$$f(\alpha, \delta) = 1, \ \textit{and,} \ f(\alpha', \delta) = 0,$$

for each $\alpha' \in A_1 \times \ldots \times A_k - \{\alpha\}$.

Then $\mathrm{SIZE}\,(\mathcal{B}) \ge \min\{|A_i|\,;\, i = 1, \ldots, k\}$.

Proof. Since the proof for some arbitrary k is a straightforward but rather technically involved consequence of the case $k = 2$, we begin with the latter. We wish to apply Lemma 1 and, to this end, we prove that for each pair of functions g^1, g^2 with $g^1 + g^2 = f$ the dimension of the space $\mathbb{B}_+ = \mathbb{B}_{G_1}(g^1) + \mathbb{B}_{G_2}(g^2)$ has a dimension greater or equal to $\min\{|A_1|, |A_2|\}$. To derive a contradiction we assume the opposite. Since $\{g^1|_\alpha\,;\, \alpha \in A_1\} \subseteq \mathbb{B}_+$ and $\{g^2|_\beta\,;\, \beta \in A_2\} \subseteq \mathbb{B}_+$, the assumption $\dim_{\mathbb{F}_2}\mathbb{B} < \min\{|A_1|, |A_2|\}$ implies for some $\alpha \in A_1, \beta \in A_2$ linear dependencies that we can state (after renaming the indices) as

$$\begin{aligned} g^1|_\alpha &= g^1|_{\alpha_1} + \ldots + g^1|_{\alpha_\mu}, \text{ and} \\ g^2|_\beta &= g^2|_{\beta_1} + \ldots + g^2|_{\beta_\nu}, \end{aligned} \tag{1}$$

with $\mu, \nu \ge 0, \alpha_i \in A_1 - \{\alpha\}$ for $1 \le i \le \nu$ and $\beta_j \in A_2 - \{\beta\}$ for $1 \le j \le \mu$. Since the setting of this lemma postulates some δ such that $f(\alpha, \beta, \delta) = 1$, we get that

$$g^1|_\alpha(\alpha, \beta, \delta) + g^2|_\beta(\alpha, \beta, \delta) = f(\alpha, \beta, \delta) = 1. \tag{2}$$

From (2) we derive in four steps a contradiction. First we apply the linear dependencies (1) and get

$$1 = \bigoplus_{i=1}^{\mu} g^1|_{\alpha_i}(\alpha_i, \beta, \delta) + \bigoplus_{j=1}^{\nu} g^2|_{\beta_j}(\alpha, \beta_j, \delta). \tag{3}$$

Since $g^1|_{\alpha_i}(\alpha_i, \beta, \delta) + g^2|_\beta(\alpha_i, \beta, \delta) = f(\alpha_i, \beta, \delta) = 0$ and $g^1|_\alpha(\alpha, \beta_j, \delta) + g^2|_{\beta_j}(\alpha, \beta_j, \delta) = f(\alpha, \beta_j, \delta) = 0$, we conclude that

$$1 = \bigoplus_{i=1}^{\mu} g^2|_\beta(\alpha_i, \beta, \delta) + \bigoplus_{j=1}^{\nu} g^1|_\alpha(\alpha, \beta_j, \delta). \tag{4}$$

Again, we apply the linear dependencies (1). Consequently,

$$1 = \bigoplus_{i=1}^{\mu}\bigoplus_{j=1}^{\nu} g^2|_{\beta_j}(\alpha_i, \beta_j, \delta) + \bigoplus_{i=1}^{\mu}\bigoplus_{j=1}^{\nu} g^1|_{\alpha_i}(\alpha_i, \beta_j, \delta) \tag{5}$$

$$= \bigoplus_{i=1}^{\mu}\bigoplus_{j=1}^{\nu} f(\alpha_i, \beta_j, \delta) = 0. \tag{6}$$

Contradiction.

Now we consider the case $k > 2$. For applying Lemma 1, we have to prove that for each choice of k functions $g^1, \ldots, g^k$ with $g^1 + \ldots + g^k = f$ the dimension of the space $\mathbb{B}_+ = \mathbb{B}_{G_1}(g^1) + \ldots + \mathbb{B}_{G_k}(g^k)$ has a dimension greater or equal to $\min\{|A_i|\,;\, i = 1, \ldots, k\}$. To derive a contradiction we assume the opposite. For all $i = 1, \ldots, k$, $\{g^i|_{\alpha^i}\,;\, \alpha^i \in A_i\} \subseteq \mathbb{B}_+$. So, $\dim_{\mathbb{F}_2}\mathbb{B} < \min\{|A_i|\,;\, i = 1, \ldots, k\}$ implies for some $\alpha_0^i \in A_i, i = 1, \ldots, k$ (after reindexing) the existence of the following linear equations.

$$g^i|_{\alpha_0^i} = g^i|_{\alpha_1^i} + \ldots + g^i|_{\alpha_{\mu(i)}^i}, \tag{7}$$

with $\alpha_j^i \in A_i - \{\alpha_0^i\}$ for $j > 0$ and $i = 1, \ldots, k$. Furthermore, by the setting of this lemma there is some $\delta = \delta(\alpha_0^1, \ldots, \alpha_0^k)$ such that for $\alpha \in A_1 \times \ldots \times A_k$

$$f(\alpha, \delta) = 1, \text{if and only if}, \alpha = (\alpha_0^1, \ldots, \alpha_0^k).$$

Consequently, for $\alpha = (\alpha_{j(1)}^1, \ldots, \alpha_{j(k)}^k)$

$$g^1|_{\alpha_{j(1)}^1}(\alpha, \delta) + \ldots + g^k|_{\alpha_{j(k)}^k}(\alpha, \delta) = 1, \tag{8}$$

if and only if $j(1) = \ldots = j(k) = 0$.

Since during the proof we have to deal with a huge number of summands, we express them by sets Σ of elements in $\{1, \ldots, k\} \times \{0, 1\}^k$. The significance of this definition is described by the following interpretation $\phi : \{1, \ldots, k\} \times \{0, 1\}^k \to \mathbb{B}_n$.

For convenience we identify $(i, b_1, \ldots, b_k)$ and $(i, (b_1, \ldots, b_k))$. We consider a $\sigma = (i, b)$ with $i \in \{1, \ldots, k\}$ and $b = (b_1, \ldots, b_k) \in \{0, 1\}^k$. From b we derive k sets of indices $I_1(b), \ldots, I_k(b), I_j(b) \subseteq \{0, \ldots, \mu(j)\}$ according to (7), by defining

$$I_j(b) := \begin{cases} \{0\} & \text{if } b_j = 0; \\ \{1, \ldots, \mu(j)\} & \text{if } b_j = 1, \end{cases}$$

for $j = 1 \ldots k$. Informally, $b_j = 0$ corresponds to the left side of equation (7) and $b_j = 1$ to the right side. Now we set

$$\phi(i, b) = \bigoplus_{(j(1), \ldots, j(k)) \in I_1(b) \times \ldots \times I_k(b)} g^i|_{\alpha_{j(i)}^i}(\alpha_{j(1)}^1, \ldots, \alpha_{j(k)}^k, \delta). \tag{9}$$

So, informally, the i in $\sigma = (i, b)$ determines the index of the function g^i. Making use of this notation, for some set Σ of such elements, we define

$$\phi(\Sigma) = \bigoplus_{\sigma \in \Sigma} \phi(\sigma).$$

In the end of this proof, we have restated the case $k = 2$ in terms of this notation. The reader may now already refer to that. Now we consider two rules (R1) and (R2), associated with the identities (7) and (8).

(R1) While Σ contains an element (i, b) with $b_i = 0$,
- remove (i, b) from Σ,
- add (i, b') to Σ, where b' results from b by skipping bit i.

(R2) For each $b \in \{0, 1\}^k$ consider $S(b) = \Sigma \cap \{(1, b), \ldots, (k, b)\}$. For $b \neq (0, \ldots, 0)$ and $S(b) \neq \emptyset$, remove all elements in $S(b)$ from Σ and add all elements in $\overline{S(b)} = \{(i, b)\,;\, i = 1, \ldots, k\} - S(b)$.

Informally, (R1) expresses an application of the linear dependencies (7). (R2) expresses an application of (8) with $j(\nu) \neq 0$ for some $\nu \in \{1, \ldots, k\}$.

Correctness of (R1). We show that, if Σ' is derived from Σ by applying rule (R1), then $\phi(\Sigma') = \phi(\Sigma)$. We just observe that in the notation of (R1)'s description, some $\phi(i, b)$ with $b_i = 0$ consists of a sum of terms of the form $g^i|_{\alpha_0^i}(a, \delta)$, with $a \in A_1 \times \ldots \times A_k$ and $a(x) = \alpha_0^i(x)$ for α_0^i is defined on x. This is the case, since in the setting of (9) we have $I_i(b) = \{0\}$. Applying (7) on each of these summands we get $\phi(i, b) = \phi(i, b')$.

Correctness of (R2). We observe that

$$\begin{aligned} & \phi(S(b) \cup \overline{S(b)}) \\ = & \sum_{i=1,\ldots,k} \phi(i, b) \\ = & \bigoplus_{(j(1),\ldots,j(k)) \in I_1(b) \times \ldots \times I_k(b)} f(\alpha^1_{j(1)}, \ldots, \alpha^k_{j(k)}, \delta) \\ = & \, 0, \end{aligned}$$

for $b \neq 0$ by (8). So $\phi(S(b)) = \phi(\overline{S(b)})$, for $b \neq (0, \ldots, 0)$, and the correctness of (R2) follows.

The contradiction. Next we show that one obtains by alternating applications of (R1) and (R2) for

$$\Sigma_0 = \{(1, 0, \ldots, 0), \ldots, (k, 0, \ldots, 0)\}$$

via

$$\Sigma_0 \overset{R1}{\to} \Sigma_1 \overset{R2}{\to} \Sigma_2 \overset{R1}{\to} \Sigma_3 \overset{R2}{\to} \Sigma_4 \overset{R1}{\to} \ldots \overset{R1}{\to} \Sigma_{2k-1} \overset{R2}{\to} \Sigma_{2k},$$

the set $\Sigma_{2k} = \emptyset$. Then we get the desired contradiction

$$1 = f(\alpha_0^1, \ldots, \alpha_0^k, \delta) = \phi(\Sigma_0) = \phi(\Sigma_{2k}) = \phi(\emptyset) = 0.$$

Let for any Boolean vector b, $|b|$ denote the number of bits b_i being 1. We show that Σ_{2i} consists of all elements (j, b) such that

– $b \in \{0, 1\}^k$ with $|b| = i$ and $b_j = 0$.

Note that then Σ_{2k} is indeed empty. For Σ_0 the claim holds by definition. Let us assume that for Σ_{2i} the claim holds. Then we get by rule (R1) that Σ_{2i+1} consists of all (j, b) such that

– $|b| = i + 1$ and $b_j = 1$.

By applying rule (R2) the stated situation is achieved immediately. Note that in neither of the two cases an element is produced twice, since otherwise the conclusion $\phi(\Sigma_i) = \phi(\Sigma_{i+1})$ would not be true.

Now putting all parts of this proof together the claim of this lemma follows. To illustrate this proof we finally restate the case $k = 2$ in its terminology. For $\Sigma_0 = \{(1, 0, 0), (2, 0, 0)\}$ we get $\phi(\Sigma_0) = 1$ in line with (2). We get $\Sigma_1 = \{(1, 1, 0), (2, 0, 1)\}$ corresponding to (3) and $\Sigma_2 = \{(2, 1, 0), (1, 0, 1)\}$ corresponding to (4). Applying rule (R1) we get $\Sigma_3 = \{(2, 1, 1), (1, 1, 1)\}$ in line with (5) and by rule (R2) we get $\Sigma_4 = \emptyset$, corresponding to (6).

The next lemma deals with the situation that in the setting of Lemma 2 for two nodes v_i and v_j with $i \neq j$ the same sets of variables are tested, i.e. $\mathrm{V}(v_i) = \mathrm{V}(v_j)$. Then the condition that each α in $A_1 \times \ldots \times A_k$ is a v-assignment implies that $|A_i| = |A_j| = 1$. But in that situation for some $\tilde{\alpha}$ defined on $\mathrm{V}(v_i) = \mathrm{V}(v_j)$ we can combine those nodes reached according to G_1 and those reached according to G_2 to one set. This is possible, since the assignments in A_i and A_j are not independently combinable. In each v-assignment$(\alpha_1, \ldots, \alpha_k)$ we have $\alpha_i = \alpha_j$.

Lemma 3. *Let $\mathcal{B}$ be a $(G_1, \ldots, G_k)$–driven sum of ⊕BP1s representing f and let v be in $G_1 \times \ldots \times G_k$. For $i = 1, \ldots, k$ let A_i be some set of assignments to $\mathrm{V}(v_i)$, such that $A_i = A_j$ for $\mathrm{V}(v_i) = \mathrm{V}(v_j), 1 \leq i \leq j \leq k$. Let $\mathcal{A}$ contain each α in $A_1 \times \ldots \times A_k$ with $\alpha_i = \alpha_j$ for $A_i = A_j$.*

We assume that each $\alpha \in \mathcal{A}$ is a v-assignment and that for all $\alpha \in \mathcal{A}$ there is some assignment δ, defined on the variables not set by α with

$$f(\alpha, \delta) = 1, \textit{and, } f(\alpha', \delta) = 0,$$

for each $\alpha' \in \mathcal{A} - \{\alpha\}$.

Then SIZE $(\mathcal{B}) \geq \min\{|A_i| \,;\, i = 1, \ldots, k\}$.

Proof. First we recapitulate the preceding arguments in a slightly modified way. Let $\mathcal{B}$ be a $(G_1, \ldots, G_k)$–driven sum of $\oplus$BP1s representing f and let v be in $G_1 \times \ldots \times G_k$. For $i = 1, \ldots, k$ let A_i be some set of assignments to $\mathrm{V}(v_i)$, such that each $\alpha \in (\alpha_1, \ldots, \alpha_k)$ is a v-assignment. Then we get with the proof of Lemma 2 that $\dim_{\mathbb{F}_2} \left(\mathrm{span}_{\mathbb{F}_2}\{g^1|_{\alpha^1} \,;\, \alpha^1 \in A_1\} + \ldots + \mathrm{span}_{\mathbb{F}_2}\{g^k|_{\alpha^k} \,;\, \alpha^k \in A_k\}\right) \geq \min\{|A_i| \,;\, i = 1, \ldots, k\}$. With the proof of Lemma 1 we get the following. There are functions $g^1, \ldots, g^k$ with $g^1 + \ldots + g^k = f$ such that SIZE $(\mathcal{B}) \geq \dim_{\mathbb{F}_2} \left(\mathrm{span}_{\mathbb{F}_2}\{g^1|_{\alpha^1} \,;\, \alpha^1 \in A_1\} + \ldots + \mathrm{span}_{\mathbb{F}_2}\{g^k|_{\alpha^k} \,;\, \alpha^k \in A_k\}\right)$.

Now we turn to the proof of this lemma and assume that $A_1 = A_2$ and that for each $\tilde{\alpha} \in A_1$ and each $(\alpha_3, \ldots, \alpha_k) \in A_3 \times \ldots \times A_k$ the sequence $(\tilde{\alpha}, \tilde{\alpha}, \alpha_3, \ldots, \alpha_k)$ is a v-assignment. We show that in this situation the claim of the lemma holds and the claim on the general setting follows by repeating this argument.

Considering the proof of Lemma 1 it is easy to see that there are functions $g^2, \ldots, g^k$ with $g^2 + \ldots + g^k = f$ such that

$$\mathrm{SIZE}\,(\mathcal{B}) \geq \dim_{\mathbb{F}_2} \left(\mathrm{span}_{\mathbb{F}_2}\{g^2|_{\alpha^2} \,;\, \alpha^2 \in A_2\} + \ldots + \mathrm{span}_{\mathbb{F}_2}\{g^k|_{\alpha^k} \,;\, \alpha^k \in A_k\}\right).$$

Now we can apply Lemma 2 and get that $\dim_{\mathbb{F}_2} \left(\mathrm{span}_{\mathbb{F}_2}\{g^2|_{\alpha^2} \,;\, \alpha^2 \in A_2\} + \ldots + \mathrm{span}_{\mathbb{F}_2}\{g^k|_{\alpha^k} \,;\, \alpha^k \in A_k\}\right) \geq \min\{|A_i| \,;\, i = 2, \ldots, k\}$. The claim follows.

In the next proposition we state the observation that we are able to set some of the variables on that some sum of graph–driven $\oplus$BP1s is defined to constants without a blow–up of the size. This may be considered to be plain, but it follows with results in [9] that it can be necessary to change the ordering.

Proposition 7. *Let $\mathcal{B}$ be a $(G_1, \ldots, G_k)$–driven sum of $\oplus$BP1s in the variables $\{x_1, \ldots, x_n\}$ representing f. Then for a variable x_i and a Boolean constant e there is a sum of graph–driven $\oplus$BP1s $\mathcal{B}'$ in the variables $\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$ that is guided by some orderings $(G'_1, \ldots, G'_k)$ representing $f_{x_i=e}$ with* SIZE $(\mathcal{B}') \leq$ SIZE $(\mathcal{B})$.

Furthermore, for some v-assignment α with $\alpha(x_i) = e$ provided α is defined on x_i, there is some $v' \in G'_1 \times \ldots \times G'_k$ such that α is a v'-assignment.

Proof. The standard method to set x_i to e is the following. For all x_i-nodes v redirect all edges reaching v to the e-successor of v. Observe that applying this method results in a sum of graph–driven $\oplus$BP1s representing $f_{x_i=e}$. In the same way we get from $G = (G_1, \ldots, G_k)$ a sequence of read–once BPs $G' = (G'_1, \ldots, G'_k)$ on the variables $\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$. To see that $\mathcal{B}'$ is driven by G' consider some assignment a to $\{x_1, \ldots, x_n\}$ and observe that if for some λ in G_i the variable x_ν is tested before x_μ, $\nu, \mu \neq i$, then the same holds in G'_i.

The latter claim follows immediately by the construction of G'.

3 Lower bounds for linear codes

A *linear code C* is a linear subspace of $\mathbb{F}_2^n$. Our first explicit lower bound is for the *characteristic function* of such a linear code C, that is $f_C : \mathbb{F}_2^n \to \{0, 1\}$ defined by $f_C(a) = 1 \iff a \in C$. To this end we will give some basic definitions and facts on linear codes.

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 $a_1 b_1 \oplus \ldots \oplus a_n b_n = 0$, for all elements $a \in C$. 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 [18] for a proof.

Lemma 4. *If C is a code of minimal distance $k+1$, then its dual $C^{\perp}$ is k–universal.*

The following theorem shows how to apply our lower bound criterion to linear codes.

Theorem 1. *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 each sum of k $\oplus$BP1s guided by a sequence of graph–orderings $G = (G_1, \ldots, G_k)$ representing its characteristic function f_C has size bounded below by $2^{\Omega\left(\min\{d,d^{\perp}\}/2^k\right)}/(|G_1| \cdot \ldots \cdot |G_k|)$.

Proof. Let $\mathcal{B}$ be a sum of graph–driven $\oplus$BP1s guided by $G = (G_1, \ldots, G_k)$ representing $f = f_C$. We set $l := \min\{d, d^{\perp}\} - 1$. Observe, that the code C is both of distance $l+1$ and l–universal. We wish to find a tuple v and sets of partial assignments $A_1, \ldots, A_k$ such that we can apply Lemma 3. We use an inductive approach and in order to make the proof readable we define the following predicate P.

We define $P(i)$ to hold if and only if

– there is a tuple $v = (v_1, \ldots, v_i) \in G_1 \times \ldots \times G_i$,
– there are sets of variables $V_1^{(i)}, \ldots, V_i^{(i)}$ with $V_j^{(i)} \subseteq \mathrm{V}(v_j)$ and $|V_j^{(i)}| \geq l/2^i$ such that for $j, k \leq i$, either $V_j^{(i)} = V_k^{(i)}$ or $V_j^{(i)} \cap V_k^{(i)} = \emptyset$,
– there is a set $\mathcal{A}_i$ of assignments with $|\,\mathcal{A}_i\,| \geq 2^n/(|G_1| \cdot \ldots \cdot |G_i|)$ such that for $j = 1, \ldots, k$ each $a \in \mathcal{A}_i$ passes in G_j the node v_j, and
– $|\bigcup_{j \leq i} \mathrm{V}(v_j)| \leq l/2 + l/4 + \ldots + l/2^i$.

Before we inductively show that $P(k)$ holds, *we argue how $P(k)$ implies the claim.* Since our aim is to find convenient assignments defined on $V_1^{(k)}, \ldots, V_k^{(k)}$, first according to $\mathcal{A}_k$ we set all variables in

$$V' = \bigcup_{j<i} (\mathrm{V}(v_j) - V_j^{(k)}),$$

to constants. Since there are at most $2^{|V'|}$ assignments defined on $|V'|$, we can fix some γ with $\mathrm{V}(\gamma) = V'$ such that for

$$\mathcal{A}_\gamma = \{\alpha \in \mathcal{A}_k \mid \alpha(x) = \gamma(x) \text{ for } x \in V'\},$$

we have

$$|\,\mathcal{A}_\gamma\,| \geq 2^{n-|\,\mathrm{V}(\gamma)|}/(|G_1| \cdot \ldots \cdot |G_k|).$$

Now by a similar argument we choose sets $A_1, \ldots, A_k$ by decomposing $\mathcal{A}_\gamma$ according to $V_1^{(k)}, \ldots, V_k^{(k)}$.

Let $\mathcal{V}$ be a set of variables. For each subset $M \subseteq \mathcal{V}$, there are at most $2^{|\mathcal{V}|-|M|}$ assignments defined on $\mathcal{V} - M$. For $j = 1, \ldots, k$ we apply this to $M = V_j^{(k)}$ and $\mathcal{V} = \{x_1, \ldots, x_n\} - \mathrm{V}(\gamma)$ and define A_j as the projection of $\mathcal{A}_\gamma$ onto $V_j^{(k)}$. Since the elements of A_γ differ only on variables contained in $\mathcal{V}$, projecting $\mathcal{A}_\gamma$ to $M = V_i^{(k)}$ results in at least

$$|\,\mathcal{A}_\gamma\,|/2^{|\mathcal{V}-M|} = \left(2^{n-|\,\mathrm{V}(\gamma)|}/(|G_1| \cdot \ldots \cdot |G_k|)\right)/2^{n-|\,\mathrm{V}(\gamma)|-|V_i^{(k)}|}$$

different partial assignments. Thus we can choose sets $A_1, \ldots, A_k$ such that A_i consists of partial assignments defined on $V_i^{(k)}$ with size

$$|A_i| \geq 2^{|V_i^{(k)}|}/(|G_1| \cdot \ldots \cdot |G_k|) \geq 2^{l/2^k}/(|G_1| \cdot \ldots \cdot |G_k|).$$

Next we apply Proposition 7 for transforming $\mathcal{B}$ into a sum of graph–driven $\oplus$BP1s $\mathcal{B}'$ representing $f|_\gamma$, i.e. we set all variables in $\mathrm{V}(\gamma)$ according to γ. Moreover, there is a sequence of graph–orderings $G' = (G'_1, \ldots, G'_k)$ and some $v' \in G'_1 \times \ldots \times G'_k$ such that each v-assignment a becomes a v'-assignment a' with

$$a' = \begin{cases} a(x) & \text{if } x \notin \mathrm{V}(\gamma); \\ \text{undefined} & \text{if } x \in \mathrm{V}(\gamma). \end{cases}$$

In line with Lemma 3 we let $\mathcal{A}$ contain each α in $A_1 \times \ldots \times A_k$ with $\alpha_i = \alpha_j$ for $A_i = A_j$. It is plain that each element of $\mathcal{A}$ is a v'-assignment. Thus, to apply Lemma 3 we only have to find for each $\alpha \in \mathcal{A}$ some partial assignment δ defined on the variables not tested up to v with $f(\alpha, \gamma, \delta) = 1$ and $f(\alpha', \gamma, \delta) = 0$ for each $\alpha' \in \mathcal{A}$ with $\alpha' \neq \alpha$. We do this with the help of the following standard arguments on linear codes that are due to Jukna ([18]).

Since $|\bigcup_{j \leq i} \mathrm{V}(v_j)| \leq l/2 + l/4 + \ldots + l/2^i < l$ we get by the l-universality the existence of some δ as claimed. $f(\alpha', \gamma, \delta) = 0$ for $\alpha' \neq \alpha$ follows since the hamming distance of two accepting assignments has to be greater or equal to l. Now we get with Lemma 3, that $\mathrm{SIZE}\,(\mathcal{B}') \geq \min\{|A_j|\,;\, j = 1, \ldots, k\} \geq 2^{l/2^k}/(|G_1| \cdot \ldots \cdot |G_k|)$ and the claim follows.

In the setting of this theorem $P(1)$ holds. We consider all nodes of G_1 at depth $l/2$ from the source. Thus for each such node v and each path π leading from the source to v exactly $l/2$ variables are tested on π. One of these nodes is passed by $2^n/|G_1|$ of these paths. We denote this node by v_1 and define $\mathcal{A}_1$ to contain all the assignments associated with these paths. We set $V_1^{(1)} = \mathrm{V}(v_1)$ and see that $P(1)$ holds.

$P(i-1)$ implies $P(i)$. For each node w of G_i we denote by

$$old(w) = \mathrm{V}(w) \cap \bigcup_{j<i} \mathrm{V}(v_j),$$

all variables tested on the path from the source of G_i to w that are already tested on the path from the source to some $v_j, j < i$. By

$$new(w) = \mathrm{V}(w) \; - \; \bigcup_{j<i} \mathrm{V}(v_j),$$

we denote those variables in $\mathrm{V}(w)$ not tested on a path to some $v_j, j < i$. Let C be the set of all nodes w of G_i such that

- $|new(w)| = l/2^i$ and $|old(w) \cap V_j^{(i-1)}| < l/2^i$ for all $j = 1, \ldots, i-1$,

or,

- $|new(w)| < l/2^i$, $|old(w) \cap V_j^{(i-1)}| = l/2^i$ for some $j \in \{1, \ldots, i-1\}$, and $|old(w) \cap V_m^{(i-1)}| < l/2^i$, for all m with $V_m^{(i-1)} \neq V_j^{(i-1)}$.

Since each path in G_i passes exactly one node of C, there is some node v_i such that $|\,\mathcal{A}_{i-1}\,|/|G_i|$ paths associated with elements of $\mathcal{A}$ pass it. We determine sets $V_1^{(i)}, \ldots, V_i^{(i)}$ in line with $P(i)$. To this end we have to distinguish two cases, dependent on the choice of v_i.

(1) Case $|new(v_i)| = l/2^i$. After definition of C we additionally get $|old(v_i) \cap V_j^{(i-1)}| < l/2^i$ for all $j = 1, \ldots, i-1$.

First we define

$$V_i^{(i)} = new(v_i),$$

and

$$V_j^{(i)} = V_j^{(i-1)} - old(v_i),$$

for $j = 1, \ldots, i-1$. Then $|V_i^{(i)}| = l/2^i$ and $|V_j^{(i)}| \geq l/2^{i-1} - l/2^i = l/2^i$ for $j = 1, \ldots, i-1$.

(2) Case $|new(v_i)| < l/2^i$. In addition it holds that $|old(v_i) \cap V_j^{(i-1)}| = l/2^i$ for some $j \in \{1, \ldots, i-1\}$ and for all $V_m^{(i-1)} \neq V_j^{(i-1)}$ it holds that $|old(v_i) \cap V_m^{(i-1)}| < l/2^i$. Let $j(1), \ldots, j(\lambda)$ be all indices such that

$$|old(v_i) \cap V_{j(1)}^{(i-1)}| = \ldots = |old(v_i) \cap V_{j(\lambda)}^{(i-1)}| = l/2^i.$$

Recall that by the choice of the sets $V_j^{(i-1)}$, $V_{j(1)}^{(i-1)} = \ldots = V_{j(\lambda)}^{(i-1)}$ and for $j \in \{j(1), \ldots, j(\lambda)\}$ and $m \notin \{j(1), \ldots, j(\lambda)\}$, $V_j^{(i-1)}$ and $V_m^{(i-1)}$ are disjoint. We define

$$V_j^{(i)} := \begin{cases} old(v_i) \cap V_j^{(i-1)} & \text{for } j \in \{j(1), \ldots, j(\lambda)\}; \\ V_j^{(i-1)} - old(v_i) & \text{for } j \notin \{j(1), \ldots, j(\lambda)\}. \end{cases}$$

Note that $|V_j^{(i)}| \geq l/2^i$ for $j = 1, \ldots, i$. So $P(i)$ holds and the claim follows.

Now we are able to formulate the following corollary, that states our first lower bound for an explicitly defined function. 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.

Corollary 2. *Let $n = 2^l$ and $r = \lfloor l/2 \rfloor$.*

Then every sum of graph–driven $\oplus$BP1s guided by a sequence of graph–orderings $G = (G_1, \ldots, G_k)$ representing the characteristic function of $R(r, l)$ has size bounded below by $2^{\Omega\left(n^{1/2}/2^k\right)/(|G_1|\cdot\ldots\cdot|G_k|)}$.

Proof. We apply that 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 [19]. □

An easy calculation shows that this bound is superpolynomial for

$$k = o\left(\frac{\log n}{\log\log n \cdot \log\log|G|}\right),$$

for $|G| = |G_1|+|G_2|+\ldots+|G_k|$. So we can conclude that for $k = o(\log n/(\log\log n)^2)$, the considered linear code is not contained in $\mathcal{P}(k^*\text{–}\oplus\text{BP1})$. We get the same result even if we allow G to have quasipolynomial size, $|G| = 2^{\log^{\mathcal{O}(1)} n}$.

In [16] Jukna observed the interesting fact that linear codes give some information about the hardness of integer multiplication. For an integer X we denote the i-th bit of its binary representation by X_i. For a subset of bits $S \subseteq \{1, \ldots, n\}$, we denote by Mult_n^S the following Boolean function on $2n$ variables. For n–bit integers X and Y, $\text{Mult}_n^S(X, Y) = 1$ if and only if, $(X \cdot Y)_i = 1$ for all $i \in S$. Jukna proved the following.

Theorem 2. *For every linear code $C \subseteq \{0, 1\}^n$ there is an integer $A \in \{1, \ldots, 2^\nu\}, \nu = (n+1)\cdot n\cdot\log n + n + 1$, an injection $\phi : \{0, 1\}^n \to \{0, 1\}^\nu$ and a subset of bits S, $|S| \leq n - \dim C$, such that for every $x \in \{0, 1\}^n$, $x \in C$ if and only if $\text{Mult}_\nu^S(A, \phi(x)) = 1$. Furthermore, x can be got from $\phi(x)$ by setting some variables to constants.*

In [17], Jukna applied this Theorem to get a lower bound for $\oplus$OBDDs and Mult_n^S. Combining our lower bound on linear codes with Proposition 7 and Theorem 2 we get the following corollary.

Corollary 3. *Each k–$\oplus$BP1 guided by some polynomial size k a sequence of graph-orderings representing Mult_n^S has size exponential in $n^{1/4-\epsilon}/2^k$ is not polynomial for $k = o(\log n/(\log\log n)^2)$.*

4 Summary

The following figure summarizes the relationship of the mentioned function classes. Additionally, $\mathcal{P}$(wsGraph–$\oplus$BP1)= $\mathcal{P}(1^*\text{–}\oplus\text{BP1})$ and $\mathcal{P}$(graph–$\oplus$BP1)= $\mathcal{P}(1\text{–}\oplus\text{BP1})$. k must not exceed the borders stated in Proposition 6 and subsequent to Theorem 1.

Acknowledgements

I would like to thank Henrik Brosenne, Carsten Damm and Stephan Waack for many helpful suggestions and comments.

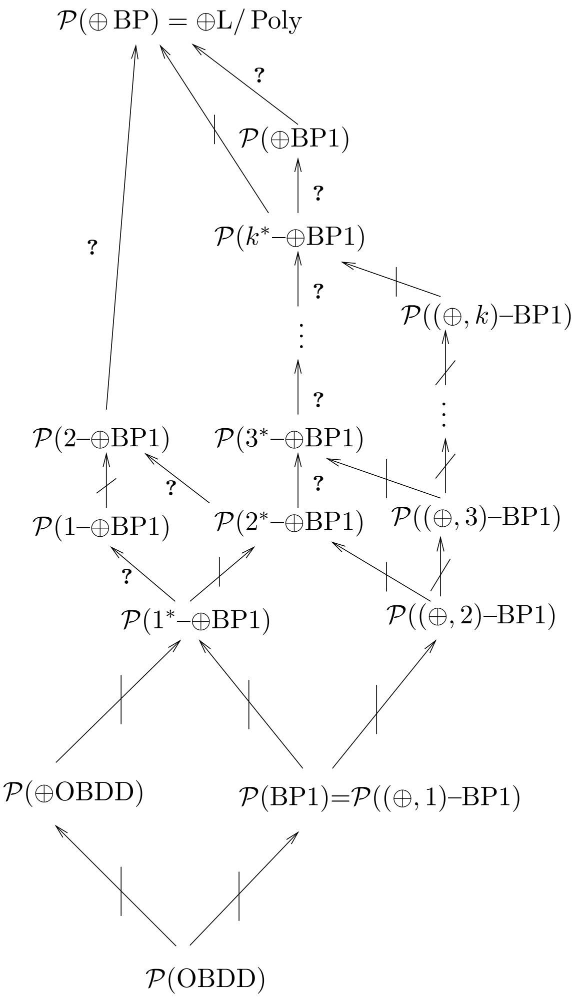
$\mathcal{P}(\oplus\,\mathrm{BP}) = \oplus\mathrm{L}/\,\mathrm{Poly}$
?
$\mathcal{P}(\oplus\mathrm{BP1})$
?
$\mathcal{P}(k^*\text{–}\oplus\mathrm{BP1})$
?
?
$\mathcal{P}((\oplus, k)\text{–BP1})$
?
$\mathcal{P}(2\text{–}\oplus\mathrm{BP1})$
$\mathcal{P}(3^*\text{–}\oplus\mathrm{BP1})$
?
?
$\mathcal{P}(1\text{–}\oplus\mathrm{BP1})$
$\mathcal{P}(2^*\text{–}\oplus\mathrm{BP1})$
$\mathcal{P}((\oplus, 3)\text{–BP1})$
?
$\mathcal{P}(1^*\text{–}\oplus\mathrm{BP1})$
$\mathcal{P}((\oplus, 2)\text{–BP1})$
$\mathcal{P}(\oplus\mathrm{OBDD})$
$\mathcal{P}(\mathrm{BP1}){=}\mathcal{P}((\oplus, 1)\text{–BP1})$
$\mathcal{P}(\mathrm{OBDD})$

Fig. 3. The landscape of the mentioned function classes.

References

1. M. Ajtai. A non-linear time lower bound for Boolean branching programs. In *Proceedings, 40th FOCS*, pages 60–70, 1999.
2. P. Beame, M. Saks, X. Sun, and E. Vee. Super–linear time-space tradeoff lower bounds for randomized computations. In *Proceedings, 41st FOCS*, pages 169–179, 2000.
3. P. Beame and E. Vee. Time-space trade-offs, multiparty communication complexity, and nearest neighbour problems. In *Proceedings, 34th STOC*, pages 688–697, 2002.
4. B. Bollig, St. Waack, and P. Woelfel. Parity graph-driven read-once branching programs and an exponential lower bound for integer multiplication. In *Proceedings 2nd IFIP International Conference on Theoretical ComputerScience*, 2002.
5. B. Bollig and P. Woelfel. A lower bound technique for nondeterministic graph-driven read-once branching programs and its applications. In *Proceedings, 27th MFCS*, Lecture Notes in Computer Science. Springer, 2002.
6. A. Borodin, A. Razborov, and R. Smolensky. On lower bounds for read-k-times branching programs. *Computational Complexity*, 3:1–18, 1993.
7. Y. Breitbart, H. B. Hunt, and D. Rosenkrantz. The size of binary decision diagrams representing Boolean functions. *Theoretical Computer Science*, 145:45–69, 1995.
8. H. Brosenne, M. Homeister, and St. Waack. Graph–driven free parity BDDs: Algorithms and lower bounds. In *Proceedings of the 26th symposion on Mathematical Foundations of Computer Science (MFCS)*, volume 2136 of *Lecture Notes in Computer Science*, pages 212–223. Springer Verlag, 2001.

9. H. Brosenne, M. Homeister, and St. Waack. Lower bounds for general graph-driven read-once parity branching programs. In *Proceedings of the 28th symposion on Mathematical Foundations of Computer Science (MFCS)*, Lecture Notes in Computer Science. Springer Verlag, 2003.
10. R. E. Bryant. Symbolic manipulation of Boolean functions using a graphical representation. In *Proceedings, 22nd DAC*, pages 688–694, Piscataway, NJ, 1985. IEEE.
11. R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, 35:677–691, 1986.
12. R. E. Bryant. On the complexity of VLSI implementations of Boolean functions with applications to integer multiplication. *IEEE Transactions on Computers*, 40:205–213, 1991.
13. J. Gergov and Ch. Meinel. Frontiers of feasible and probabilistic feasible Boolean manipulation with branching programs. In *Proceedings, 10th STACS*, volume 665 of *Lecture Notes in Computer Science*, pages 576–585. Springer Verlag, 1993.
14. 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*, 8:273–282, 1996.
15. M. Homeister. On well–structured parity–FBDDs. In *Proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technology*, 2003.
16. S. Jukna. The graph of integer multiplication is hard for read–k–times networks. *Tech. Rep. 95-10, University of Trier*, 1995.
17. S. Jukna. *Combinatorics of Finite Computations - The Lower Bounds Problem.* Habilitationsschrift, University of Trier, 1999.
18. S. Jukna. Linear codes are hard for oblivious read-once parity branching programs. *Information Processing Letters*, 69:267–269, 1999.
19. E. J. MacWilliams and N. J. A. Sloane. *The Theory of Error–Correcting Codes.* Elsevier, 1977.
20. Ch. Meinel and H. Sack. Heuristics for $\oplus$-OBDDs. In *Proceedings, IEEE/ACM International Workshop of Logic and Synthesis*, pages 304–309, 2001.
21. Ch. Meinel and H. Sack. Improving XOR-node placements for $\oplus$-OBDDs. In *Proceedings, 5th International Workshop of Reed-Muller Expansion in Circuit Design*, pages 51–55, 2001.
22. È. I. Nechiporuk. A Boolean function. *Sov. Math. Doklady*, 7:999–1000, 1966.
23. H. Sack. *Improving the Power of OBDDs by Integrating Parity Nodes.* PhD thesis, Univ. Trier, 2001.
24. P. Savický and D. Sieling. A hierarchy result for read–once branching programs with restricted parity nondeterminism. In *Proceedings, 25th MFCS*, volume 1893 of *Lecture Notes in Computer Science*, pages 650–659. Springer Verlag, 2000.
25. D. Sieling. Lower bounds for linear transformed OBDDs and FBDDs. In *Proceedings, FSTTCS*, number 1738 in Lecture Notes in Computer Science, pages 356–368. Springer Verlag, 1999.
26. D. Sieling and I. Wegener. Graph driven BDDs – a new data structure for Boolean functions. *Theoretical Computer Science*, 141:238–310, 1995.
27. J. Thathachar. On separating the read-k-times hierarchy. In *Proceedings, 30th STOC*, pages 653–662, 1998.
28. St. Waack. On the descriptive and algorithmic power of parity ordered binary decision diagrams. *Information and Computation*, 166:61–70, 2001.
29. I. Wegener. *Branching Programs and Binary Decision Diagrams – Theory and Applications.* SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, 2000.