PARITY GRAPH-DRIVEN READ-ONCE BRANCHING PROGRAMS AND AN EXPONENTIAL LOWER BOUND FOR INTEGER MULTIPLICATION

(Extended Abstract)

Beate Bollig*
FB Informatik, LS2, Univ. Dortmund, Germany
bollig@ls2.cs.uni-dortmund.de

Stephan Waack
Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen, Germany
waack@math.uni-goettingen.de

Philipp Woelfel*
FB Informatik, LS2, Univ. Dortmund, Germany
woelfel@ls2.cs.uni-dortmund.de

Abstract Branching programs are a well-established computation model for boolean functions, especially read-once branching programs have been studied intensively. Exponential lower bounds for deterministic and nondeterministic read-once branching programs are known for a long time. On the other hand, the problem of proving superpolynomial lower bounds for parity read-once branching programs is still open. In this paper restricted parity read-once branching programs are considered and an exponential lower bound on the size of well-structured parity graph-driven read-once branching programs for integer multiplication is proven. This is the first strongly exponential lower bound on the size of a nonoblivious parity read-once branching program model for an explicitly defined boolean function. In addition, more insight into the structure of integer multiplication is yielded.

*Supported in part by DFG grant WE 1066.

1. Introduction

Branching programs (BPs) or Binary Decision Diagrams (BDDs) are a well-established representation type or computation model for boolean functions.

Definition 1 A *branching program* (BP) or binary decision diagram (BDD) on the variable set $X_n = \{x_1, \ldots, x_n\}$ is a directed acyclic graph with one source and two sinks labeled by the constants 0 and 1. Each non-sink node (or internal node) is labeled by a boolean variable and has two outgoing edges, one labeled by 0 and the other by 1. A *nondeterministic branching program* is a generalized branching program where the number of edges leaving an internal node is not restricted.

An input $a \in \{0,1\}^n$ activates all edges consistent with a, i.e., the edges labeled by a_i which leave nodes labeled by x_i. A computation path for an input a in a BP G is a path of edges activated by a that leads from the source to a sink. A computation path for an input a which leads to the 1-sink is called accepting path for a.

The output for an input a is 1 iff there is an accepting path for a. A *parity branching program* is a nondeterministic branching program with the parity acceptance mode, i.e., an input is accepted iff the number of its accepting paths is odd.

The size of a branching program G is the number of its nodes and is denoted by $|G|$. The branching program size of a boolean function f is the size of the smallest BP representing f. The length of a branching program is the maximum length of a path.

The branching program size of a boolean function f is known to be a measure for the space complexity of nonuniform Turing machines and known to lie between the circuit size of f and its $\{\wedge, \vee, \neg\}$-formula size (see, e.g., [19]). Hence, one is interested in exponential lower bounds for more and more general types of BPs (for the latest breakthrough for semantic super-linear length BPs see [1], [3] and [4]). In order to develop and strengthen lower bound techniques one considers restricted computation models.

Definition 2 i) A branching program is called (syntactically) *read k times* (BPk) if each variable is tested on each path at most k times.

ii) A BP is called *s-oblivious*, for a sequence of variables $s = (s_1, \ldots, s_l)$, $s_i \in X_n$, if the set of its internal nodes can be partitioned into disjoint sets V_i, $1 \le i \le l$, such that all nodes from V_i are labeled by s_i and the edges which leave V_i-nodes reach a sink or a V_j-node, $j > i$.

Bryant [9] has introduced ordered binary decision diagrams (OBDDs) which are up to now the most popular representation for formal circuit verification. OBDDs are oblivious BP1s, where on each path from the source to a sink the variables are tested accoding to a *variable ordering* given by a permutation π on the variable set. Unfortunately, several important and also quite simple functions have exponential OBDD size. Therefore, Gergov and Meinel [12] and Sieling and Wegener [17] have generalized independently the concept of variable orderings.

Definition 3 A *graph ordering* is a branching program with a single sink, where on each path from the source to the sink all variables appear exactly once. A (parity) *graph-driven* BP1 with respect to a graph ordering G_0, (parity) G_0-BP1 for short, is a (parity) BP1 with the following additional property. If for an input a, a variable x_i appears on the unique computation path of a in G_0 before the variable x_j, then x_i also appears on all computation paths of a in G before x_j.

(Note that the size of a (parity) G_0-BP1 G is the number of nodes in G and not in G and G_0.)

For many restricted (nondeterministic) variants of branching programs exponential lower bounds are known (for a survey see e.g. [15]). Moreover, Thathachar [18] has been able to prove an exponential gap between the size of nondeterministic BPks and deterministic BP$(k+1)$s for an explicitly defined boolean function. His results have demonstrated that the lower bound techniques for these models are highly developed. Nevertheless, the problem of proving superpolynomial lower bounds for parity read-once branching programs is still open. Krause [13] has proved the first exponential lower bounds for oblivious parity branching programs with bounded length. Later, Savický and Sieling [16] have presented exponential lower bounds for restricted parity read-once branching programs. In their model only at the top of the read-once branching program parity nodes are allowed. Recently, Brosenne, Homeister, and Waack [8] have proved the first (not strongly) exponential lower bound on the size of restricted parity graph-driven BP1s representing the characteristic function of linear codes.

Motivated by applications the analysis of *natural* functions like the basic arithmetic functions is of interest.

Definition 4 *Integer multiplication* MUL_n maps two n-bit integers $x = x_{n-1}\dots x_0$ and $y = y_{n-1}\dots y_0$ to their product $x \cdot y = z = z_{2n-1}\dots z_0$. $\mathrm{MUL}_{i,n}$ denotes the boolean function defined as the ith bit of MUL_n.

The middle bit of multiplication ($\mathrm{MUL}_{n-1,n}$) is known to be the *hardest* bit. Hence, in the following we only consider the function $\mathrm{MUL}_n := \mathrm{MUL}_{n-1,n}$. For OBDDs Bryant [10] has presented an exponential lower bound of size $2^{n/8}$ for MUL_n. Incorporating Ramsey theoretic arguments of Alon and Maass [2] and using the rank method of communication complexity, Gergov [11] has extended the lower bound to arbitrary nondeterministic linear-length oblivious BPs. Recently, Woelfel [21] has improved Bryant's lower bound up to $\Omega(2^{n/2})$. The first exponential lower bound on the size of deterministic BP1s has been proven by Ponzio [14]. His lower bound is of order $2^{\Omega(n^{1/2})}$ and has been improved by Bollig and Woelfel [7] to the first strongly exponential lower bound of size $\Omega(2^{n/4})$ for MUL_n. Bollig [5] has presented the first (not strongly) exponential lower bound on the size of MUL_n for so-called nondeterministic tree-driven BP1s. Her result also holds for parity tree-driven BP1s. Until now exponential lower bounds on the size of MUL_n for general nondeterministic BP1s or BPks with $k \geq 2$ are unknown. Here we present an exponential lower bound on the size of restricted parity graph-driven BP1s for MUL_n. This is the first strongly exponential lower bound for this branching program model. In addition, we yield more insight into the structure of integer multiplication.

Due to the lack of space we have to omit some of the proofs. For a full version of the paper see [6].

2. The Lower Bound Criterion

In [17] a restricted variant of graph-driven BP1s has been investigated.

Definition 5 A graph-driven BP1 $G = (V, E)$ with respect to a graph ordering $G_0 = (V_0, E_0)$ is called *well-structured* if there exists a representation function $\alpha : V \rightarrow V_0$ with the following properties. The nodes v and $\alpha(v)$ are labeled by the same variable and for all inputs a such that v lies on the computation path for the input a the node $\alpha(v)$ lies on the path in G_0 which is activated by a.

Similar to the deterministic case well-structured parity G_0-BP1s are defined. The difference between graph-driven and well-structured graph-driven BP1s is the following one. In the general graph-driven model it is possible that two different inputs reach in G the same node labeled by x_i, whereas they reach in the graph-ordering G_0 different nodes labeled by x_i. This is not allowed in the well-structured case.

Brosenne, Homeister, and Waack [8] have realized how this restriction can be used to determine the number of nodes that is necessary to represent a boolean function f in a well-structured parity graph-driven BP1. A further observation which turns out to be very helpful in order to prove exponential lower bounds is the following one. The size of a well-structured parity graph-driven BP1 G and the size of a graph ordering G_0 of minimal size such that G is G_0-driven are polynomially related. First, we need the following lemma which is a slight generalization of a result from [17].

Lemma 1 ([8]) *Let G_0 be a graph ordering, v a node in a well-structured parity G_0-BP1 G, α the representation function, and $c \in \{0, 1\}$. If w is one of the c-successors of v in G then all paths to the sink in G_0 which leave $\alpha(v)$ via the c-edge pass through $\alpha(w)$.*

Proposition 1 *Let G be a well-structured parity graph driven* BP1 *on n variables. There exists a graph ordering G_0 such that G is G_0-driven and $|G_0| \leq 2n|G|$.*

Proof. Let G_0' be a graph ordering such that G is G_0'-driven and let $\mathcal{N}_v(G)$ be the set of nodes u in G such that $\alpha(u) = v$. First, we mark all nodes v in G_0' for which $\mathcal{N}_v(G)$ is not empty. Afterwards we eliminate all nodes which have not been marked in G_0'. An edge leading to one of theses nodes v is redirected to the first successor of v which has been marked. Because of Lemma 1 this node is uniquely determined. The resulting graph is a read-once branching program with one sink and at most $|G|$ nodes. Finally, we use the usual algorithm (see also [20]) to insert nodes such that on each path from the source to the sink there exist for each variable x_i exactly one node labeled by x_i. According to a topological ordering of the nodes, for each node v the set $V(v)$ of variables tested on some path from the source to v excluding the label of v is computed. Afterwards on each edge (v, w) dummy tests of the variables in $V(w) \setminus V(v)$ excluding the variable tested at v are added. A dummy test is a node where the 0- and the 1-edge lead to the same node.

The resulting graph ordering G_0 consists of at most $2n|G|$ nodes. It is easy to see that G is G_0-driven. $\square$

The proof of Proposition 1 cannot be generalized in a straightforward way for (general) parity graph-driven BP1s because the existence of the α-function is an essential part of the proof. Until now exponential lower bounds on the size of general parity graph-driven BP1s are unknown.

In the following, we consider the representation of a boolean function f by its value table as an element of $(\mathbb{Z}_2)^{2^n}$. This set is a $\mathbb{Z}_2$ vector space where addition is component-wise parity and scalar multiplication by 0 or 1 is defined in the obvious way. Before we state our lower bound criterion, we have to introduce some notations. Let v be a node in the graph ordering G_0, G a well-structured parity G_0-driven BP1, $\mathcal{N}_v(G)$ the set of nodes u in G such that $\alpha(u) = v$, and f a boolean function. On all paths from the source to v the same set of variables has to be tested. W.l.o.g. let $x_1, \ldots, x_{i-1}$ be the previously tested variables and let v be labeled by x_i. Let $A(v) \subseteq \{0,1\}^{i-1}$ be the set of vectors $(a_1, \ldots, a_{i-1})$ such that v is reached for all inputs a starting with $(a_1, \ldots, a_{i-1})$. We define $\mathcal{F}_v := \{f_{|x_1=a_1,\ldots,x_{i-1}=a_{i-1}} | (a_1, \ldots, a_{i-1}) \in A(v)\}$. The functions of $\mathcal{F}_v$ depend syntactically on all variables $x_1, \ldots, x_n$ but they do not depend essentially on $x_1, \ldots, x_{i-1}$. (A function g essentially depends on a variable x_j iff $g_{|x_j=0} \neq g_{|x_j=1}$.) Now let $\mathcal{P}_v$ be the set of all nodes that lie on a path leaving v in G_0 including v. Then we define $\mathbb{B}^{G_0}_{f,v}$ as the boolean vector space spanned by all functions in $\bigcup_{w \in \mathcal{P}_v} \mathcal{F}_w$.

Let V be a vector space and V_1, V_2 be sub vector spaces of V. V_1 is said to be *linearly independent modulo* V_2, if $V_1 \cap V_2 = \{\mathrm{o}\}$, i.e., $\dim V_1 + \dim V_2 = \dim(V_1 + V_2)$.

Lemma 2 *Let $A'(v)$ be a subset of $A(v)$ such that the subfunctions $f_{|x_1=a_1,\ldots,x_{i-1}=a_{i-1}}$, $(a_1, \ldots, a_{i-1}) \in A'(v)$, are linearly independent, and let $\mathbb{B}^{G_0}_{f,A'}$ be the vector space spanned by these subfunctions. If $\mathbb{B}^{G_0}_{f,A'}$ is linearly independent modulo the vector space of all subfunctions in $\mathbb{B}^{G_0}_{f,v}$ not essentially depending on x_i, then $|\mathcal{N}_v(G)| \geq |A'(v)|$.*

3. Integer Multiplication and the Matrix Game

We start our investigations with two technical lemmas which provide important properties of the function MUL_n.

In the rest of the paper $[x]_{n-k}^{n-1}$ denotes the bits at position $n-1$ to $n-k$ in the binary representation of the integer x. Using universal hashing Bollig and Woelfel [7, proof of Lemma 5] have shown the following.

Lemma 3 (Covering Lemma) *Let $X \subseteq \mathbb{Z}_{2^n}$ and $Y \subseteq \mathbb{Z}_{2^n}^* := \{1,3,\ldots,2^n-1\}$. If $|X|\cdot|Y| \geq 2^{n+2k+1}$, $k \geq 0$, then there exists an element $y^* \in Y$ such that*

$$\forall z \in \{0,\ldots,2^k-1\}\ \ \exists x \in X:\ [xy^*]_{n-k}^{n-1} = z.$$

The lemma states that if X and Y are large enough sets of (odd) n-bit integers, then by choosing an appropriate $y \in Y$, the possible outcomes in the bits $n-1,\ldots,n-k$ of the products xy for $x \in X$ cover all possible k-bit values. (Note that Bollig and Woelfel [7] have proved this statement only implicitly in a non-parameterized form.) We now state another important lemma about integer multiplication, which is a generalization of Lemma 6 from [7].

Lemma 4 (Distance Lemma) *Let $Y \subseteq \mathbb{Z}_{2^{n-1}}^*$, $1 \leq k \leq n-3$, and $(z_i, z_i') \in \mathbb{Z}_{2^{n-1}} \times \mathbb{Z}_{2^{n-1}}$, where $z_i \neq z_i'$, $1 \leq i \leq t$. Then there exists a subset $Y' \subseteq Y$ with*

$$\forall y \in Y': 4\cdot 2^{n-k-1} \leq ((z_i - z_i')y) \bmod 2^{n-1} \leq 2^{n-1} - 4\cdot 2^{n-k-1}$$

such that $|Y'| \geq |Y| - t\cdot 2^{n-k+1}$.

Proof. Let $\delta_i := (z_i - z_i') \bmod 2^{n-1}$, $1 \leq i \leq t$, and

$$\begin{aligned} M' &:= \{0,\ldots,4\cdot 2^{n-k-1}-1\} \text{ and} \\ M'' &:= \{2^{n-1} - 4\cdot 2^{n-k-1}+1,\ldots,2^{n-1}-1\}. \end{aligned}$$

Let Y' be the set of all $y \in Y$ where $(y\delta_i) \bmod 2^{n-1} \notin M' \cup M''$ for all $i \in \{1,\ldots,t\}$. Bollig and Woelfel [7, proof of Lemma 4] have shown that the number of $y \in Y$ with $(y\delta_i) \bmod 2^{n-1} \in M' \cup M''$ for a fixed $i \in \{1,\ldots,t\}$ is bounded above by 2^{n-k+1}. Therefore, for at most $t\cdot 2^{n-k+1}$ elements $y \in Y$ there exists at least one element $i \in \{1,\ldots,t\}$ such that $(y\delta_i) \bmod 2^{n-1} \in M' \cup M''$. Altogether, we have proved that the size of Y' is at least $|Y| - t\cdot 2^{n-k+1}$. □

Before we state our main lemma about properties of integer multiplication, we motivate our investigations. Let G_0 be a graph ordering which is not too large. Then we can prove that there exists a node v such that w.l.o.g. at least as many x- as y-variables have been tested from the source to v, v is labeled by a variable x_i, and there is a partial assignment a^* to the y-variables tested on the paths to v such that many paths which agree for the tested y-variables with a^* lead to v. Let $A'(v)$ be the set of these assignments. Now our aim is to prove that the boolean vector space spanned by the subfunctions of MUL_n according to $A'(v)$ is linearly independent modulo the vector space spanned by all subfunctions not essentially depending on V^*, where V^* contains x_i and the variables which have been tested on the paths to v. Then we can conclude using Lemma 2 that the size of well-structured parity G_0-BP1s representing MUL_n is large.

In the following, we investigate integer multiplication of two binary numbers $x = (x_{n-1}, \ldots, x_0)$ and $y = (y_{n-1}, \ldots, y_0)$, where $x_{n-1} = y_{n-1} = 0$ and $x_0 = y_0 = 1$. Let $V_x = \{x_1, \ldots, x_{n-2}\}$ and $V_y = \{y_1, \ldots, y_{n-2}\}$. Furthermore, let $V'_x \subseteq V_x$ ($V'_y \subseteq V_y$) be a set of m x-variables (y-variables), where $m \leq \lfloor (n-17)/6 \rfloor$. We fix an arbitrary assignment of the V'_y-variables. Now we consider a $2^m \times 2^{2n-2m-4}$ matrix M. Each row is associated with one assignment of the V'_x-variables and each column with an assignment of the variables from $V_x \setminus V'_x$ and $V_y \setminus V'_y$. Together with the fixed assignment of the V'_y-variables, $x_{n-1} = y_{n-1} = 0$ and $x_0 = y_0 = 1$, we obtain two well-defined n-bit numbers $x_{r,c}$ and y_c for each pair (r, c) of a row and a column. We define $M_{r,c}$ as $\mathrm{MUL}_n(x_{r,c}, y_c)$. Finally, we define for an arbitrary fixed variable $x_i \in V_x \setminus V'_x$ and a column c the column c' as the one which only differs from c by the assignment to the variable x_i.

Now our aim is to show that for an arbitrary choice of different rows $r^1, \ldots, r^l$, there exists a column c such that

$$\bigoplus_{j=1}^{l} M_{r^j,c} \neq \bigoplus_{j=1}^{l} M_{r^j,c'}. \tag{1}$$

Before we show (1) we illustrate how this property can be used to prove lower bounds using Lemma 2. The set of all possible assignments of the V'_x- and V'_y-variables is a superset of the set $A(v)$. By fixing the V'_y-variables by an arbitrary assignment, we obtain a set $A^*(v)$ which determines the matrix M. The number of a row of M identifies an assignment α determined by an element in $A^*(v)$ and the row itself represents the function vector of the subfunction $\mathrm{MUL}_{|\alpha}$. In this setting, (1) is the following. If we take an arbitrary linear combination of subfunctions (represented by the rows $r^1, \ldots, r^l$), then there exist two assignments to the variables in $(V_x \setminus V'_x) \cup (V_y \setminus V'_y)$ differing only in their setting to x_i such that the function value of the linear combination is different for both assignments. Hence, no subfunction not essentially depending on the V'_x- and V'_y-variables and x_i can be represented as a linear combination of the subfunctions determined by $A^*(v)$. By Lemma 2, this allows the conclusion that $|\mathcal{N}_v(G)| \geq |A'(v)|$, where $A'(v) \subseteq A^*(v)$.

We return to the proof of (1). Let $x_{r,c}$ be the number in $\mathbb{Z}^*_{2^{n-1}}$ defined by the choice of a row r and a column c and y_c the number in $\mathbb{Z}^*_{2^{n-1}}$ defined by the choice of the column c and the fixed assignment of the V'_y-variables. Therefore,

$$M_{r,c} = [x_{r,c} \cdot y_c]_{n-1}.$$

The number $x_{r,c}$ can be written as the sum of two components $x_r^{row} + x_c^{col}$, where x_r^{row} is the number defined by the partial assignment of the V_x'-variables given by the row r and the 0-assignment of the variables from $V_x \setminus V_x'$ and x_c^{col} is the number defined by the partial assignment of the variables from $V_x \setminus V_x'$, $x_0 = 1$, and the 0-assignment of the V_x'-variables. It follows that $M_{r,c} = [(x_r^{row} + x_c^{col}) \cdot y_c]_{n-1}$.

We take a look at the columns where for an arbitrary i the variable x_i is set to 0. Obviously the set of all pairs (x_c^{col}, y_c) of theses columns c corresponds to a set $X \times Y$ where $X, Y \subseteq \mathbb{Z}^*_{2^{n-1}}$, $|X| = 2^{n-m-3}$, and $|Y| = 2^{n-m-2}$. Furthermore, $x_{c'}^{row} - x_c^{row} = 2^i$. Finally, the choice of l rows $r^1, \ldots, r^l$ corresponds to the numbers $x_{r^1}^{row}, \ldots, x_{r^l}^{row}$. For the ease of description we denote these numbers by $x^1, \ldots, x^l$.

Summarizing, our aim is to prove that, under the assumption discussed above, for arbitrarily chosen $x^1, \ldots, x^l$ there exists a pair $(x, y) \in X \times Y$ such that the number of indices $j \in \{1, \ldots, l\}$ for which

$$[(x^j + x)y]_{n-1} \neq [(x^j + x + 2^i)y]_{n-1}$$

is odd. Formally this leads to the statement of Lemma 5.

Lemma 5 *Let $m \leq \lfloor (n-17)/6 \rfloor$, $1 \leq l \leq 2^m$, $X, Y \subseteq \mathbb{Z}^*_{2^{n-1}}$, $d \neq 0$, and let $x^1, \ldots x^l$ be elements from $\mathbb{Z}_{2^n-1}$ with the following properties:*

i) $|X| \geq 2^{n-m-3}$ *and* $|Y| \geq 2^{n-m-2}$,

ii) $\forall\, 2 \leq j \leq l:\ x^1 \neq x^j \quad$ *and* $\quad \forall\, 1 \leq j \leq l:\ x^1 \neq x^j + d$,

iii) for all $x \in X$ *and all* $1 \leq j \leq l$: $x + x^j + d < 2^{n-1}$.

Let $(x, y) \in X \times Y$ and let $\sigma(x, y)$ be the number of indices $j \in \{1, \ldots, l\}$ where $[(x^j + x)y]_{n-1} \neq [(x^j + x + d)y]_{n-1}$. Then there exists a pair $(x, y) \in X \times Y$ such that $\sigma(x, y)$ is odd.

Obviously, the conditions of Lemma 5 are fulfilled for $d = 2^i$ and our choice of $x^1, \ldots, x^l$ and X and Y as described above. (Note, that we have achieved (iii) by setting $x_{n-1} = y_{n-1} = 0$.)

Proof. Let $k = 2m + 5$ and $X' := \{x^1 + x \mid x \in X\}$. Clearly $|X'| = |X| \geq 2^{n-m-3}$. Because of condition (iii), X' is a subset of $\mathbb{Z}_{2^{n-1}}$. First, we consider the $2l - 1$ pairs (x^1, z) where $z \in Z := \{x^2, \ldots, x^l\} \cup \{x^1 + d, \ldots, x^l + d\}$. Because of condition (iii), all $z \in Z$ are elements of $\mathbb{Z}_{2^{n-1}}$ and, because of condition (ii), they are all different from x^1. Let Y' be the set of all $y \in Y$ such that for all pairs (x^1, z), $z \in Z$,

$$4 \cdot 2^{n-k-1} \leq ((z - x^1)y) \bmod 2^{n-1} \leq 2^{n-1} - 4 \cdot 2^{n-k-1}. \tag{2}$$

According to Lemma 4

$$\begin{aligned} |Y'| &\geq |Y| - (2l-1)2^{n-k+1} > |Y| - 2^{m+1+n-k+1} \\ &\geq 2^{n-m-2} - 2^{n-m-3} = 2^{n-m-3}. \end{aligned}$$

Here we have used the fact that $2l \leq 2^{m+1}$. Using $m \leq \lfloor (n-17)/6 \rfloor$ we can conclude that

$$|X'| \cdot |Y'| \;\geq\; 2^{2n-2m-6} \;\geq\; 2^{2n-n/3+17/3-6} \;=\; 2^{n+(2/3)n-1/3}.$$

Since $k = 2m+5$, it follows that

$$2^{n+2k+1} \;=\; 2^{n+4m+11} \;\leq\; 2^{n+(2/3)n-34/3+11} \;=\; 2^{n+(2/3)n-1/3}$$

such that we obtain $|X'| \cdot |Y'| \;\geq\; 2^{n+2k+1}$. Now we can apply Lemma 3. According to this there exist an element $y^* \in Y'$ and $x^*, x^{**} \in X'$ such that

$$[x^* y^*]_{n-k}^{n-1} \;=\; 2^{k-1} - 1 \quad \text{and} \quad [x^{**} y^*]_{n-k}^{n-1} \;=\; 2^{k-1}.$$

Let $y = y^*$. According to the definition of X' we can write x^* as $x^1 + x$ and x^{**} as $x^1 + x'$ for two elements $x, x' \in X$ such that

$$[(x^1 + x)y]_{n-k}^{n-1} \;=\; 2^{k-1} - 1 \quad \text{and} \quad [(x^1 + x')y]_{n-k}^{n-1} \;=\; 2^{k-1}. \tag{3}$$

Next we prove the following claims for x and x':

(C1) $[(x^1 + x)y]_{n-1} \neq [(x^1 + x')y]_{n-1}$.

(C2) For all $2 \leq i \leq l$: $[(x^i + x)y]_{n-1} = [(x^i + x')y]_{n-1}$.

(C3) For all $1 \leq i \leq l$: $[(x^i + x + d)y]_{n-1} = [(x^i + x' + d)y]_{n-1}$.

Using these claims we can prove in the following way that either $\sigma(x,y) = \sigma(x',y) - 1$ or $\sigma(x,y) = \sigma(x',y) + 1$. From (C1) and (C3) for $i = 1$ we can conclude that

$$[(x^1 + x)y]_{n-1} = [(x^1 + x + d)y]_{n-1} \;\Leftrightarrow\; [(x^1 + x')y]_{n-1} \neq [(x^1 + x' + d)y]_{n-1},$$

and from (C2) and (C3) that

$$[(x^i + x)y]_{n-1} = [(x^i + x + d)y]_{n-1} \;\Leftrightarrow\; [(x^i + x')y]_{n-1} = [(x^i + x' + d)y]_{n-1}$$

for $i = 2, \ldots, l$.

Therefore, exactly one of the values $\sigma(x,y)$ or $\sigma(x',y)$ is odd and we can complete our proof by proving (C1)-(C3). (C1) follows immediately from equation (3). To prove (C2) and (C3) we reconsider the pairs (x^1, z), $z \in Z = \{x^2, \ldots, x^l, x^1 + d, \ldots, x^l + d\}$. Obviously, it is sufficient to prove that $[(z + x)y]_{n-1} = [(z + x')y]_{n-1}$, for all $z \in Z$. We assume that this is not the case, w.l.o.g. $[(z + x)y]_{n-1} = 0$ and $[(z + x')y]_{n-1} = 1$ (the other case follows similarly).

According to equation (3) it follows that

$$2^{n-1} - 2^{n-k} \;\leq\; ((x^1 + x)y) \bmod 2^n \;<\; 2^{n-1} \quad \text{and} \tag{4}$$

$$2^{n-1} \;\leq\; ((x^1 + x')y) \bmod 2^n \;<\; 2^{n-1} + 2^{n-k}. \tag{5}$$

From this it follows that

$$1 \leq ((x' - x)y) \bmod 2^n < 2 \cdot 2^{n-k}. \tag{6}$$

From our assumption $[(z+x)y]_{n-1} = 0$ and $[(z+x')y]_{n-1} = 1$ we know that

$$((z+x)y) \bmod 2^n < 2^{n-1} \leq ((z+x')y) \bmod 2^n.$$

Since $((z+x')y) \bmod 2^n - ((z+x)y) \bmod 2^n = ((x'-x)y) \bmod 2^n$, we can conclude using inequality (6)

$$2^{n-1} - 2 \cdot 2^{n-k} \leq ((z+x)y) \bmod 2^n < 2^{n-1}.$$

Together with inequality (4) we obtain

$$-2 \cdot 2^{n-k} < ((z+x)y) \bmod 2^n - ((x^1+x)y) \bmod 2^n < 2^{n-k}.$$

Considering all terms in this inequality modulo 2^{n-1} it follows that

$$((z-x^1)y) \bmod 2^{n-1} < 2^{n-k} \quad \text{or} \quad ((z-x^1)y) \bmod 2^{n-1} > 2^{n-1} - 2 \cdot 2^{n-k}.$$

But this is a contradiction to inequality (2) and we are done. □

Altogether, we have proved that the vector space spanned by all subfunctions of MUL_n according to all assignments of the m V'_x-variables and an arbitrary assignment a^* of the m V'_y-variables is linearly independent modulo the vector space spanned by all subfunctions of MUL_n according to all assignments of the V'_x- and V'_y-variables not essentially depending on a variable x_i from $V_x \setminus V'_x$.

4. A Strongly Exponential Lower Bound for Integer Multiplication

Combining the new lower bound technique for well-structured parity graph-driven BP1s with Lemma 5 we prove the first strongly exponential lower bound on the size of a nonoblivious parity branching program model.

Theorem 1 *The size of well-structured parity graph-driven BP1s representing* MUL_n *is bounded below by* $2^{(n-46)/12}/n$.

Proof. Let G be a well-structured parity graph-driven BP1 representing MUL_n and G_0 be a graph ordering of minimal size such that G is G_0-driven. We may assume that the size of G_0 is at most $2^{1/2\lfloor (n-17)/6 \rfloor}$, because otherwise using Proposition 1 we can conclude that the size of parity graph-driven BP1s representing MUL_n is bounded below by

$$2^{1/2\lfloor (n-17)/6 \rfloor}/(4n) \geq 2^{(1/2)\cdot(n-22)/6}/(4n) = 2^{(n-46)/12}/n.$$

Let $m := \lfloor (n-17)/6 \rfloor$, $V_x = \{x_1, \ldots, x_{n-2}\}$, and $V_y = \{y_1, \ldots, y_{n-2}\}$. Since on all paths in G_0 all variables have to be tested, it is obvious that on all paths from the source to a node v the same set of variables is tested. In the following we only investigate paths where $x_0 = y_0 = 1$ and $x_{n-1} = y_{n-1} = 0$. We define a cut in the graph ordering G_0 in the following way. The cut consists of all nodes v where v is labeled by a V_x-variable and on all paths to v exactly m V_x-variables and at most m V_y-variables have been tested (or vice versa). On each path in G_0 there is exactly one node of the cut. Using the pigeonhole principle there exists one node v which lies on at least $2^{2n-4}/|G_0|$ paths from the source to the sink. W.l.o.g. v is labeled by x_i, and m V_x-variables and m' V_y-variables, $m' \leq m$, have been tested. Using the pigeonhole principle again there exists one partial assignment a^* to the V_y-variables tested on the paths from the source to v such that there are at least $2^m/|G_0|$ paths to v which agree for the V_y-variables with the partial assignment a^*. Let $A'(v)$ be the set of all assignments associated with these paths, V'_x (V'_y) be the set of the x-variables (y-variables) which have been tested, and let v be labeled by x_i. Clearly the requirements from Lemma 5 are fulfilled and we can conclude that the vector space spanned by all subfunctions according to $A'(v)$ is linearly independent modulo the vector space of all subfunctions not essentially depending on the V'_x- and the V'_y-variables and x_i. Therefore, we obtain the result

$$|\mathcal{N}_v(G)| \geq |A'(v)| \geq 2^{1/2\lfloor (n-17)/6 \rfloor}.$$

Altogether, we have proved a lower bound of $2^{1/2\lfloor (n-17)/6 \rfloor}/4n$,which is at least $2^{(n-46)/12}/n$, on the size of well-structured parity graph-driven BP1s representing MUL_n. □

Acknowledgments

We would like to thank Stefan Droste and Ingo Wegener for proofreading and fruitful discussions on the subject of the paper.

References

[1] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings of the 40th Annual IEEE Symposium on Fountations of Computer Science*, pp. 60–70. 1999.

[2] N. Alon and W. Maass. Meanders and their applications in lower bounds arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.

[3] P. Beame, M. Saks, X. Sun, and E. Vee. Super-linear time-space tradeoff lower bounds for randomized computation. In *Proceedings of the 41st Annual IEEE Symposium on Fountations of Computer Science*, pp. 169–179. 2000.

[4] P. Beame and E. Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. 2002. To appear.

[5] B. Bollig. Restricted nondeterministic read-once branching programs and an exponential lower bound for integer multiplication. *RAIRO Theoretical Informatics and Applications*, 35:149–162, 2001.

[6] B. Bollig, S. Waack, and P. Woelfel. Parity graph-driven read-once branching programs and an exponential lower bound for integer multiplication. Technical Report TR01-73, Electronic Colloquium on Computational Complexity, 2001.

[7] B. Bollig and P. Woelfel. A read-once branching program lower bound of $\Omega(2^{n/4})$ for integer multiplication using universal hashing. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 419–424. 2001.

[8] H. Brosenne, M. Homeister, and S. Waack. Graph-driven free parity BDDs: Algorithms and lower bounds. In *Mathematical Foundations of Computer Science: 26th International Symposium*, volume 2136 of *Lecture Notes in Computer Science*, pp. 212–223. 2001.

[9] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35:677–691, 1986.

[10] R. E. Bryant. On the complexity of VLSI implementations and graph representations of boolean functions with applications to integer multiplication. *IEEE Transactions on Computers*, 40:205–213, 1991.

[11] J. Gergov. Time-space tradeoffs for integer multiplication on various types of input oblivious sequential machines. *Information Processing Letters*, 51:265–269, 1994.

[12] J. Gergov and C. Meinel. Efficient analysis and manipulation of OBDDs can be extended to FBDDs. *IEEE Transactions on Computers*, 43:1197–1209, 1994.

[13] M. Krause. Separating $\oplus$L from L, NL, co-NL, and AL (=P) for oblivious turing machines of linear access time. *RAIRO Theoretical Informatics and Applications*, 26:507–522, 1992.

[14] S. Ponzio. A lower bound for integer multiplication with read-once branching programs. *SIAM Journal on Computing*, 28:798–815, 1998.

[15] A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *Proc. of Fundamentals in Computation Theory*, volume 529 of *Lecture Notes in Computer Science*, pp. 47–60. 1991.

[16] P. Savický and D. Sieling. A hierarchy result for read-once branching programs with restricted parity nondeterminism. In *Mathematical Foundations of Computer Science: 25th International Symposium*, volume 1893 of *Lecture Notes in Computer Science*, pp. 650–659. 2000.

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

[18] J. S. Thathachar. On separating the read-k-times branching program hierarchy. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 653–662. 1998.

[19] I. Wegener. *The Complexity of Boolean Functions*. Wiley-Teubner, 1987.

[20] I. Wegener. *Branching Programs and Binary Decision Diagrams - Theory and Applications*. SIAM, first edition, 2000.

[21] P. Woelfel. New bounds on the OBDD-size of integer multiplication via universal hashing. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pp. 563–574. 2001.