On the Complexity of Integer Multiplication in Branching Programs with Multiple Tests and in Read-Once Branching Programs with Limited Nondeterminism

(Extended Abstract)

Philipp Woelfel*
FB Informatik, LS2
Univ. Dortmund
44221 Dortmund, Germany
woelfel@Ls2.cs.uni-dortmund.de

Abstract

Branching Programs (BPs) are a well-established computation and representation model for Boolean functions. Although exponential lower bounds for restricted BPs such as Read-Once Branching Programs (BP1s) have been known for a long time, the proof of lower bounds for important selected functions is sometimes difficult. Especially the complexity of fundamental functions such as integer multiplication in different BP models is interesting.

In [4], the first strongly exponential lower bound of $\Omega(2^{n/4})$ has been proven for the complexity of integer multiplication in the deterministic BP1 model. Here, we consider two well-studied BP models which generalize BP1s by allowing a limited amount of nondeterminism and multiple variable tests, respectively. More precisely, we prove a lower bound of $\Omega(2^{n/(7k)})$ for the complexity of integer multiplication in the $(\vee, k)$-BP model. As a corollary, we obtain that integer multiplication cannot be represented in polynomial size by nondeterministic BP1s, if the number of nondeterministic nodes is bounded by $\log n - \log\log n - \omega(1)$. Furthermore, we show that any $(1, +k)$-BP representing integer multiplication has a size of $\Omega\left(2^{\frac{n}{48(k+1)}}\right)$. This is not polynomial for $k = o(n/\log n)$.

1. Introduction

1.1 Branching Programs

Branching Programs (BPs) or equivalently binary decision diagrams (BDDs) belong to the most important nonuniform models of computation. Deterministic and nondeterministic BPs can be simulated by the corresponding Turing machines, and the BP complexity of a Boolean function is known to be a measure for the space complexity of the corresponding model of sequential computation. Therefore, one is interested in large lower bounds for the complexity of explicitly defined functions in BP models.

Definition 1 A (deterministic) *Branching Program* (BP) on the variable set $\mathcal{Z}_n = \{z_1, \ldots, z_n\}$ is a directed acyclic graph with one source and two sinks. The internal nodes are marked with variables in $\mathcal{Z}_n$ and the sinks are labeled with the Boolean constants 0 and 1. Further, each internal node has two outgoing edges, marked with 0 and 1, respectively. A *nondeterministic* Branching Program is a BP with some additional unmarked nodes having out-degree 2, which are called *nondeterministic nodes*.

Let G be a (possibly nondeterministic) BP on $\mathcal{Z}_n$ and $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ an assignment to the variables in $\mathcal{Z}_n$. A source-to-sink path in G is called *computation path* of a, if it leaves any node marked with z_i over the edge labeled with a_i. Note that in a deterministic BP each input defines exactly one computation path, while in a nondeterministic BP multiple computation paths are possible for each input.

Let B_n denote the set of Boolean functions $f : \{0,1\}^n \to \{0,1\}$. The BP G represents a function $f \in B_n$, for which $f(a) = 1$ if and only if there exists a computation path for a leading to the 1-sink.

The *size* of G, denoted by $|G|$, is the number of its nodes. The *Branching Program complexity* of a Boolean function f is the size of the smallest BP representing f.

Until today, no super-polynomial lower bounds for the size of general BPs representing an explicitly defined func-

*Supported in part by DFG grant We 1066

tion are known. Therefore, various types of restricted BPs have been investigated, and one is interested in refining the proof techniques in order to obtain lower bounds for less restricted BPs.

There are several possibilities to restrict BPs, among them restrictions on the multiplicity of variable tests or the ordering in which variables may be tested.

Definition 2 (i) A *Read-Once Branching Program* (short: BP1) is a BP where each variable appears on each computation path at most once.

(ii) A BP is called *oblivious*, if the nodes can be partitioned into levels such that edges point only from lower to higher levels and all internal nodes of one level are marked with the same variable. An *Ordered Binary Decision Diagram* (short: OBDD) is an oblivious BP1.

(iii) A $(\vee, k)$-BP1 G is a family of k deterministic BP1s $G_1, \ldots, G_k$. If $f_1, \ldots, f_k$ are the functions represented by $G_1, \ldots, G_k$, then G represents the function $f_1 \vee \ldots \vee f_k$. The size of G is $|G_1| + \ldots + |G_k|$.

OBDDs are probably the most popular Branching Programs, because they have found many applications, e.g. in the area of circuit verification and model checking. On the other hand, BP1s have been studied for a long time by complexity theorists. The first exponential lower bounds date back to the 80s [13, 16], and today, lower bounds for explicitly defined functions are as large as $2^{n-O(\log^2 n)}$ [1]. Borodin, Razborov and Smolensky [5] have presented a lower bound technique for nondeterministic BP1s, and until today, almost all exponential lower bounds for this model are based on their technique.

Note that $(\vee, k)$-BP1s can be regarded as nondeterministic BP1s with $k-1$ nondeterministic nodes which are located at the top of the BP1. Lower bounds for this BP-model have been proven e.g. by Savický and Sieling [11]. It is motivated by Jain, Bitner, Fussell and Abraham [8], who suggested to use so-called *Partitioned BDDs*, which are $(\vee, k)$-BP1s where the corresponding BP1s are in fact OBDDs.

In order to consider more general BP models than BP1s, one can allow to test variables more than once on the computation paths. Then, one can either restrict the number of tests for each variable or the number of variables which are allowed to be tested more than once.

Definition 3 (i) A *read-k-times Branching Program* (short: BPk) is a BP where each variable appears on each computation path at most k times.

(ii) A $(1, +k)$-BP is a BP where for each computation path p there exist at most k variables appearing on p more than once.

The technique of Borodin, Razborov and Smolensky [5] also allows exponential lower bounds for *syntactic* BPks, i.e. BPks, where the restriction that each variable can be tested at most k times holds even for each graph-theoretical path. Super-polynomial lower bound proofs for $(1, +k)$-BPs can be found in several papers, and have been proven for values of k up to $o(n/\log n)$ [9].

1.2. Integer Multiplication

In order to prove exponential lower bounds for arbitrary but explicitly defined Boolean functions, the functions are often "designed" in such a way that they fit the lower bound techniques. But besides the interest in finding lower bounds as large as possible or proving lower super-polynomial lower bounds for more and more general BP models, it is necessary to determine the complexity of *important* Boolean functions. Furthermore, it is generally a more difficult task to prove large lower bounds for some predefined and interesting function. Finding lower bound proofs for such functions may either help to develop new or refined proof-techniques, or can lead to new insights in the properties of the considered function.

Motivated by the applications the basic arithmetic functions are of particular interest.

Definition 4 The function $\mathrm{MUL}_{k,n} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the Boolean function which computes the bit z_k of the product $(z_{2n-1} \ldots z_0)$ of two n-bit integers $(x_{n-1} \ldots x_0)$ and $(y_{n-1} \ldots y_0)$.

Generally, the middle bit of integer multiplication (i.e. the $(n-1)$th) bit is the most difficult bit to compute. Hence, we consider the function $\mathrm{MUL} := \mathrm{MUL}_{n-1,n}$.

Bryant [6] has proven an exponential lower bound of $2^{n/8}$ for the function MUL in the OBDD model, and Gergov has shown an exponential lower bound for nondeterministic linear-length oblivious Branching Programs [7]. Ponzio later showed that the complexity of this function is $2^{\Omega(\sqrt{n})}$ for BP1s [10]. Recently, progress in determining the complexity of integer multiplication was obtained by a new approach in the analysis of MUL, which was based on universal hashing. Woelfel [14] improved Bryant's lower bound in the OBDD model to $\Omega\left(2^{n/2}\right)$ and Bollig and Woelfel [4] showed a lower bound of $\Omega\left(2^{n/4}\right)$ for general BP1s.

But oblivious BPs are very restricted in the sense that the variable tests have to be performed in the same order on every computation path. A generalization of obliviousness for nondeterministic BP1s is obtained by restricting the order of variable tests in such a way that it equals for each input the order of variable tests being performed in a complete deterministic BP1 (the so-called *graph ordering*). The resulting restricted nondeterministic BP1s are called *graph-driven* and if the graph ordering is a tree, then they are

called *tree-driven*. An exponential lower bound for nondeterministic tree-driven BP1s representing MUL was proven by Bollig [2]. Using the results about integer multiplication from [4], Bollig, Waack, and Woelfel [3] have proven an exponential lower bound for the complexity of MUL in restricted parity-nondeterministic graph-driven BP1s (parity-nondeterminism means that the function value equals 1 iff an odd number of computation paths for the given input reaches the 1-sink).

1.3. Results

Nevertheless, there is yet no super-polynomial lower bound known for MUL in the general nondeterministic BP1 model or for deterministic BPks with $k > 1$. Furthermore, all the known lower bounds require that the BP model is either read-once and deterministic or that the order in which variable tests can be performed is somehow restricted.

One contribution of this paper consists of exponential lower bounds for nondeterministic BP1s which obey no restriction on the ordering of variable tests, but where the amount of nondeterminism is limited. More precisely, we prove that any $(\vee, k)$-BP1 for MUL has a size of $\Omega\left(2^{n/(7k)}\right)$. As a corollary, we obtain that any nondeterministic BP1 for MUL has super-polynomial size if the number of nondeterministic nodes is bounded by $\log n - \log\log n - \omega(1)$. Furthermore, we present the first exponential lower bound for a deterministic nonoblivious model allowing multiple variable tests. We show that any $(1, +k)$-BP for MUL has a size of $\Omega\left(2^{\frac{n}{48(k+1)}}\right)$ and hence is not polynomial if $k = o(n/\log n)$. This matches asymptotically the best known values of k, for which super-polynomial lower bounds have been proven [9].

In order to prove the results, we define properties of Boolean functions, and prove that these properties imply the desired lower bounds. This way, we obtain new lower bound techniques for the two models, which might be of independent interest.

2. Preliminaries

In the following text, we consider functions defined on the $n + m$ Boolean variables $\mathcal{X}_n = \{x_1, \ldots, x_n\}$ and $\mathcal{Y}_m = \{y_1, \ldots, y_m\}$. We denote by B_{n+m} the set of Boolean functions $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, and assume that an input to such a function is always specified by an assignment to the variables in $\mathcal{X}_n \cup \mathcal{Y}_m$. If not otherwise specified, then X_n is the set of all n-bit strings and Y_m is the set of all m-bit strings and the elements of X_n and Y_m are specified by assignments to the variables in $\mathcal{X}_n$ and $\mathcal{Y}_m$, respectively. Hence, a function $f \in B_{n+m}$ takes as input a pair $(a, b) \in X_n \times Y_m$.

If we do not want to distinguish between x- and y-variables, then we talk about z-variables, where we mean variables in $\mathcal{Z}_{n+m} = \{z_1, \ldots, z_{n+m}\}$. A *partial assignment* to $\mathcal{Z}_{n+m}$ is an element $\alpha = (\alpha_1, \ldots, \alpha_{n+m}) \in \{0, 1, *\}^{n+m}$. While a position α_i with value 0 or 1 means that the input variable z_i is fixed to the corresponding constant, a value of $*$ means that the input variable remains free. The set $S(\alpha) := \{z_i \mid \alpha_i \neq *\}$ is called the *support* of α. We say that a complete assignment $a = (a_1, \ldots, a_{n+m})$ to $\mathcal{Z}_{n+m}$ is *consistent* with a partial assignment α to the z-variables, if $a_i = \alpha_i$ for all $z_i \in S(\alpha)$. If $f \in B_{n+m}$ and α is a partial assignment to $\mathcal{Z}_{n+m}$, then $f|_\alpha$ is the subfunction of f obtained by restricting it to inputs consistent with α. If α and β are partial assignments with disjoint supports, then $\alpha\beta$ means the partial assignment for which

$$(\alpha\beta)_i = \begin{cases} \alpha_i & \text{if } z_i \in S(\alpha) \\ \beta_i & \text{if } z_i \in S(\beta) \\ * & \text{otherwise.} \end{cases}$$

If α and β are partial assignments with the same support, then $D(\alpha, \beta)$ denotes the set of variables $z_i \in S(\alpha)$ $(= S(\beta))$, for which $\alpha_i \neq \beta_i$. Finally, let p be a path in a BP G leading from the source to an arbitrary node. We say that a partial input α *induces* the path p, if $\alpha_i = c$ $(c \in \{0,1\})$ for any c-edge of p leaving a node marked with z_i.

3. Lower Bounds for $(\vee, k)$-BP1s

In [15], a property called *k-wise l-mixed* has been defined for Boolean functions. It was proven there, that any function having this property has a complexity of $2^{\Omega(l)}$ in the $(1, +(k-1))$-BP model and in the nondeterministic BP1 model, where the number of nondeterministic nodes is bounded by $k-1$. The reason why we cannot simply apply this technique to the function MUL is that it is not possible to distinguish between different variable types by using the technique. More precisely, the technique requires to consider different subfunctions which are obtained by setting arbitrary variables to constants. Hence, for integer multiplication, two different subfunctions might be obtained by assigning different constants to some x- and some y-variables. But the properties we know about integer multiplication allow us only to consider partial assignments, where either only the x-assignments or only the y-assignments differ.

Therefore, we introduce a somewhat related but more involved property of Boolean functions, which can cope with two types of variables, and is applicable to integer multiplication.

3.1. The Lower Bound Technique

We define this property by the description of a k-round game played by two players, Alice and Bob. We call the

game $\mathcal{G}^{\vee}(f,k,l,\varepsilon)$, where $f \in B_{n+m}$, k is the number of rounds, l is a positive integer, and $0 \le \varepsilon \le 1$.

Let $B_0 = Y_m$. Alice starts the ith round ($1 \le i \le k$) by choosing a subset $A_i \subseteq B_{i-1}$, $|A_i| \ge \varepsilon |B_{i-1}|$, a set $V_i \subseteq \mathcal{X}_n \setminus (V_1 \cup \ldots \cup V_{i-1})$, $|V_i| \le l$, and two partial assignments $\alpha_i \ne \beta_i$ with support V_i. After that, Bob finishes the round by choosing $B_i \subseteq A_i$ and $\gamma_i \in \{\alpha_i, \beta_i\}$.

We denote by γ_i^* the element in $\{\alpha_i, \beta_i\} \setminus \{\gamma_i\}$ (for $1 \le i \le k$), and after a k-round game we let $c = \gamma_1 \ldots \gamma_k$ and c_i be the partial input obtained from c by replacing γ_i with γ_i^*. Bob wins the game, if after k rounds there exists an element $b \in B_k$ and a partial assignment a to the variables in $\mathcal{X}_n \setminus (V_1 \cup \ldots \cup V_k)$ such that

$$f|_c(a,b) = 1 \quad \wedge \quad \Big(\bigvee_{j=1}^{k} f|_{c_j}(a,b) = 0 \Big).$$

Otherwise, Alice wins.

If Bob has a winning strategy for the game $\mathcal{G}^{\vee}(f,k,l,\varepsilon)$ for some function $f \in B_{n+m}$, then the function has a property which makes it hard for $(\vee,k)$-BP1s.

Theorem 1 *Let $f \in B_{n+m}$ and k and l be two integers with $1 \le kl < n$. If Bob has a winning strategy for the game $\mathcal{G}^{\vee}(f,k,l,\varepsilon)$, where $\varepsilon \ge 1/2^{2l-1}$, then any $(\vee,k)$-BP1 representing f has a size of at least $2^{l-1}+2$.*

The theorem will be proven in the rest of this section. In order to do so, we state after the following definition a lemma which implies that if a $(\vee,k)$-BP1 $\{G_1,\ldots,G_k\}$ is too small, then Alice can make her choices in such a way that each computation path induced by c and c_j ($1 \le j \le k$) will meet at an edge in G_j .

Definition 5 Let $e = (u,v)$ be an edge of a deterministic BP G defined on some variable set $\mathcal{Z}$. The set $L_G(e)$ consists of all pairs (τ,τ') of partial assignments to the variables in $\mathcal{Z}$ for which the following two conditions are fulfilled.

1. $S(\tau) = S(\tau')$ and $\tau \ne \tau'$.
2. The partial assignments τ and τ' induce computation paths p and p' starting at the source and passing through e such that any variable in $D(\tau,\tau')$ is either tested on p and p' before v is reached or is not tested on any path from v to a sink.

We denote by L_G the union of all $L_G(e)$.

The idea behind the set $L_G(e)$ is that a computation reaching an edge e cannot distinguish two partial inputs τ, τ' with $(\tau,\tau') \in L_G(e)$. In other words, the BP "forgets" at the edge e that τ and τ' are different.

Lemma 1 *Let G be a deterministic BP1 representing a function $f \in B_{n+m}$ depending on the $n+m$ variables in $\mathcal{X}_n \cup \mathcal{Y}_m$, and let $Y^* \subseteq Y_m$. Let further $1 \le l < n$ be an integer such that $|G| \le 2^{l-1}+1$. Then there exist*

1. *an edge e in G,*
2. *a set $V \subseteq \mathcal{X}_n$, $|V| = l$,*
3. *partial assignments $\alpha \ne \beta$ to the x-variables with $S(\alpha) = S(\beta) = V$,*
4. *and a set $Y' \subseteq Y^*$ with $|Y'| \ge |Y^*|/2^{2l-1}$*

such that $\big((\alpha,b),(\beta,b)\big) \in L_G(e)$ for all $b \in Y'$.

This lemma is the main ingredient to the lower bound proof of Theorem 1. In order to prove it, we combine the idea behind the well-known lower bound technique for BP1s of Simon and Szegedy [12] with the following combinatorial statement, which helps us to cope with two different variable types.

Proposition 1 *Let $D = (d_{i,j})$ be an $(r \times c)$-matrix over $\{0,1\}$ having at least a fraction of ε ($0 \le \varepsilon \le 1$) 1-entries. Then there exist two rows $i \ne i'$ such that for at least $\varepsilon c(\varepsilon - 1/r)$ columns j the entries $d_{i,j}$ and $d_{i',j}$ are equal to 1.*

Proof. We define for a column j and for two different rows i, i'

$$\delta_j(i,i') := \begin{cases} 1 & \text{if } d_{i,j} = d_{i',j} = 1, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Let further γ_j be the total number of 1-entries in column j. Since $\sum_{j=1}^{c} \gamma_j \ge \varepsilon r c$ by the assumption that at least an ε-fraction of the matrix entries equals 1, we have

$$\sum_{j=1}^{c} \binom{\gamma_j}{2} = \sum_{j=1}^{c} \frac{\gamma_j(\gamma_j-1)}{2} \ge \sum_{j=1}^{c} \frac{1}{2} \frac{\varepsilon r c}{c} \Big(\frac{\varepsilon r c}{c} - 1 \Big) = \frac{\varepsilon r c(\varepsilon r - 1)}{2}. \quad (1)$$

For the inequality we have used the convexity of $\binom{x}{2}$. Therefore, we get

$$\sum_{\substack{1 \le i < i' \le r \\ 1 \le j \le c}} \delta_j(i,i') = \sum_{j=1}^{c} \sum_{1 \le i < i' \le r} \delta_j(i,i') = \sum_{j=1}^{c} \binom{\gamma_j}{2} \overset{(1)}{\ge} \frac{\varepsilon r c(\varepsilon r - 1)}{2}.$$

Hence, by the pigeon hole principle there exist two rows $i < i'$ such that

$$\sum_{j=1}^{c} \delta_j(i,i') \;\geq\; \sum_{\substack{1\leq i<i'\leq r\\ 1\leq j\leq c}} \frac{\delta_j(i,i')}{\binom{r}{2}} \;\geq\; \frac{\varepsilon r c(\varepsilon r-1)/2}{r(r-1)/2} \;\geq\; \varepsilon c(\varepsilon - 1/r),$$

which proves the claim. ∎

We are now ready to prove Lemma 1.

Proof of Lemma 1. For each node u of G except the root we let u^+ be the set of variables $x_i \in \mathcal{X}_n$ for which there exists a path from u to a sink passing over a node marked with x_i. Similarly, we let u^- be the set of variables $x_i \in \mathcal{X}_n$ for which there exists a path from the source to u passing over a node marked with x_i. (Note that if u is marked with x_i then $u^+ \cap u^- = \{x_i\}$). For the root r we let $r^+ = \mathcal{X}_n$ and $r^- = \{x_j\}$ if r is marked with x_j and $r^- = \emptyset$ if r is marked with a y-variable.

We define a cut $\mathcal{C}$ through the BP1, which is the set of edges $e = (u,v)$ for which $|u^+| > n-l$ and for which $|v^+| \leq n-l$. Since $r^+ = \mathcal{X}_n$ for the root r, $s^+ = \emptyset$ for each sink s, and $v^+ \subseteq u^+$ for each edge (u,v), it follows that each graph-theoretical source-to-sink path passes through exactly one edge in the cut $\mathcal{C}$.

For each such edge $e = (u,v) \in \mathcal{C}$, we let $V_e \subseteq \mathcal{X}_n$ be an arbitrary l-element set for which $u^- \subseteq V_e \subseteq \mathcal{X}_n \setminus v^+$. (Such a set exists for each edge in $\mathcal{C}$, because $|u^-| \leq l$, $|\mathcal{X}_n \setminus v^+| \geq l$ and $u^- \subseteq \mathcal{X}_n \setminus v^+$.) Hence, all x-variables being tested on any path from the source to e are contained in V_e, but no variable in V_e is tested on a path from e to a sink. Clearly, for any fixed $b \in Y^*$ the elements $a \in X$ for which the inputs (a,b) pass over e are uniquely defined by their setting of the variables in V_e.

Following standard graph theoretical arguments it can be seen that $\mathcal{C}$ contains at most $|G|-1$ edges. Therefore, by the pigeon-hole principle, there exists an edge $e \in \mathcal{C}$ such that at least $2^l|Y^*|/(|G|-1)$ different elements (α, b), where α is a partial assignment with support V_e and b is an element of Y^*, induce computation paths passing over e. Now assume that (α, b) and (β, b) are two different such elements. It is obvious that then $\big((\alpha,b),(\beta,b)\big) \in L_G(e)$.

We choose $V = V_e$ (recall that $|V_e| = l$). We have to show that there exist two different assignments α, β to the variables in V as well as a set $Y' \subseteq Y^*$ of large enough size such that all partial inputs (α, b) and (β, b) for $b \in Y'$ induce computation paths passing over e. In order to do so, we consider a $(2^l \times |Y^*|)$-matrix D, where each row is associated with a partial assignment to the x-variables having support V and where each column is associated with an element $b \in Y^*$. We write $d_{\alpha,b}$ for the entry in the row associated with the partial x-assignment α and in the column associated with b. We let each entry $d_{\alpha,b}$ be 1 if the partial input (α, b) passes over e and 0 otherwise.

By the discussion above, we know that the fraction of 1-entries in the matrix is at least $1/(|G|-1) \geq 1/2^{l-1}$. Applying Proposition 1 we obtain that there exist two different rows associated with α and β, respectively such that the number of elements $b \in Y^*$ with $d_{\alpha,b} = d_{\beta,b} = 1$ is bounded below by

$$1/2^{l-1} \cdot |Y^*|\Big(1/2^{l-1} - 1/2^l\Big) \;\geq\; |Y^*|/2^{2l-1}.$$

Let these α and β be fixed and let Y' be the set of the $|Y^*|/2^{2l-1}$ corresponding complete y-assignments. Then for each $b \in Y'$ we have $\big((\alpha,b),(\beta,b)\big) \in L_G(e)$. ∎

The following obvious fact helps us to apply Lemma 1

Fact 1 *Let G be a deterministic BP representing the function f, $(\tau,\tau') \in L_G$ and z be an assignment to the variables not in $S(\tau)$. Then either $f|_\tau(z) = f|_{\tau'}(z)$ or there exists a variable in $D(\tau,\tau')$ which is tested more than once on the computation path of τz (and on the computation path of $\tau' z$).*

We can finally prove the correctness of the lower bound technique.

Proof of Theorem 1. Let $G_1, \ldots, G_k$ be deterministic BP1s such that $G = \{G_1, \ldots, G_k\}$ is an $(\vee, k)$-BP1 with at most $2^{l-1}+1$ nodes representing a function f. We show the theorem by describing a winning strategy for Alice.

She will make her choices in such a way that after the ith round

$$\forall b \in B_i : \quad \big((\gamma_1 \ldots \gamma_i, b),\ (\gamma_1 \ldots \gamma_{i-1}\gamma_i^*, b)\big) \in L_{G_i}. \tag{2}$$

In the first round Alice achieves this by applying Lemma 1 to G_1 for $Y^* = B_0\ (= Y_m)$. This is possible because $|G_1| \leq |G| \leq 2^{l-1}+1$ as required by the lemma. Let α_1 and β_1 be assignments obtained this way and $A_1 \subseteq B_0$ be the resulting set of y-inputs. Note that $|A_1| \geq \varepsilon|B_0|$ for our choice of ε. Alice chooses $V_1 = S(\alpha_1)\ (= S(\beta_1))$. Then we have $|V_1| = l$, $\alpha_1 \neq \beta_1$, and after Bob's choice of $\gamma_1 \in \{\alpha_1, \beta_1\}$ and $B_1 \subseteq A_1$, (2) is fulfilled.

Assume now that (2) holds after the ith round $(1 \leq i < k)$. In the $(i+1)$th round we consider the BP1 $G_{(i+1)|_{\gamma_1\ldots\gamma_i}}$, which is obtained from G_i by restricting it to inputs according to $\gamma_1 \ldots \gamma_i$. Such a restriction is easily obtained by replacing any node with its d-successor, if it is marked with a variable x_j, which is set by $\gamma_1 \ldots \gamma_i$ to the Boolean constant d. We apply now Lemma 1 in the same way as in the first round, but now to the restricted BP1 $G^* := G_{(i+1)|_{\gamma_1\ldots\gamma_i}}$ and for $Y^* = B_i$. (Note that G^* is defined on at least $n - il > l$ x-variables.) Then there exists

an edge e, the set $A_{i+1} \subseteq B_i$ and the partial assignments α_{i+1} and β_{i+1} such that $\big((\alpha_{i+1}, b), (\beta_{i+1}, b)\big) \in L_{G^*}(e)$. Again it holds that $|A_{i+1}| \geq \varepsilon |B_i|$. As in the first round, Alice chooses $V_{i+1} = S(\alpha_{i+1})$ $(= S(\beta_{i+1}))$ and because $V_{i+1} = l$ and $\alpha_{i+1} \neq \beta_{i+1}$, the choice is made according to the rules of the game. Obviously, any input passing through the edge e in G^* also passes through the same edge in G. Hence, $\big((\gamma_1 \dots \gamma_i \alpha_{i+1}, b), (\gamma_1 \dots \gamma_i \beta_{i+1}, b)\big) \in L_G$ for all $b \in A_{i+1}$. Altogether, after Bob's choice of $\gamma_{i+1} \in \{\alpha_{i+1}, \beta_{i+1}\}$ (2) holds again.

Now assume that Alice and Bob have played the game k rounds such that (2) was fulfilled after each round. Let a be an arbitrary assignment to the variables in $\mathcal{X}_n \setminus (V_1 \cup \ldots \cup V_k)$ and let $b \in B_k$. We show that Alice has won the game, i.e.

$$f|_c(a,b) = 0 \quad \vee \quad \Big(\bigvee_{j=1}^{k} f|_{c_j}(a,b) = 1\Big), \qquad (3)$$

where c and c_j are defined as in the description of the game. We assume that $f|_c(a,b) = 1$ because otherwise there is nothing to show. In this case, there exists a deterministic BP1 $G_i \in \{G_1, \dots, G_k\}$ computing 1 for the input $(\gamma_1 \dots \gamma_k a, b)$. Note that $b \in B_k$ and hence by the rules of the game $b \in B_i$, too. Since (2) was fulfilled after each round, we know that $\big((\gamma_1 \dots \gamma_i, b), (\gamma_1 \dots \gamma_{i-1} \gamma_i^*, b)\big) \in L_G$. Then, because G_i is read-once, Fact 1 implies that $f|_{\gamma_1 \dots \gamma_i, b}(a) = f|_{\gamma_1 \dots \gamma_{i-1} \gamma_i^*, b}(a)$ and therefore

$$f(\gamma_1 \dots \gamma_k a, b) \;=\; f(\gamma_1 \dots \gamma_{i-1} \gamma_i^* \gamma_{i+1} \dots \gamma_k a, b).$$

Hence, $f|_{c_i}(a,b) = f|_c(a,b) = 1$, which proves (3). ∎

3.2. A Lower Bound for Integer Multiplication

In this section, we derive the lower bound for MUL in the $(\vee, k)$-BP model.

Theorem 2 *Any $(\vee, k)$-BP1 for* MUL *has a size of* $\Omega\big(2^{n/(7k)}\big)$.

Note that we can "simulate" any nondeterministic BP1 G with t nondeterministic nodes by a $(\vee, k)$-BP1, $k = 2^t$, which has a size of at most $k|G|$. This is done by hardwiring each of the at most 2^t possible nondeterministic decision-strings into G, which yields k deterministic BP1s. Hence, we obtain the following corollary from Theorem 2.

Corollary 1 *Any nondeterministic BP1 for* MUL *which contains at most t nondeterministic nodes has a size of $2^{\Omega(n/2^t)-t}$, which is not polynomial for $t = \log n - \log\log n - \omega(1)$.*

We prove Theorem 2 for the subfunction $\mathrm{MUL}' := \mathrm{MUL}|_{y_0=1}$ of MUL, where the y-inputs are restricted to odd integers.

Let $1 \leq l \leq (n-6)/(7k) - 3/7$ and $\varepsilon = 1/2^{2l-1}$. We show in the rest of the section that Bob has a winning strategy for the game $\mathcal{G}^{\vee}(\mathrm{MUL}', k, l, \varepsilon)$. Then Theorem 2 follows right away from Theorem 1 and from the fact that MUL′ is a subfunction of MUL.

Let $\mathbb{Z}_{2^n}$ be the set of n-bit integers and $\mathbb{Z}_{2^n}^* = \{1, 3, \dots, 2^n - 1\}$ be the set of odd n-bit integers. For the ease of notation, we do not distinguish between n-bit strings and the corresponding integer values if we consider inputs for the function MUL′. E.g., we write $\mathrm{MUL}'(x,y)$ as well for $(x,y) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}^*$ as for $(x,y) \in \{0,1\}^n \times \{0,1\}^{n-1}$. Note also, that the set of x-variables specifying the x-input for MUL′ is now $\{x_0, \dots, x_{n-1}\} =: \mathcal{X}'_n$. If τ is a partial assignment to the variables in $\mathcal{X}'_n$, then $|\tau|$ denotes the integer in $\mathbb{Z}_{2^n}$ which corresponds to the bit string $t_{n-1} \dots t_0$, where $t_i = \tau_i$ if $x_i \in S(\tau)$ and $t_i = 0$ otherwise.

Fact 2 *If τ, τ' are partial assignments to the x-variables such that $S(\tau) \cap S(\tau') = \emptyset$, then $|\tau| + |\tau'| = |\tau\tau'|$.*

We first recall two results about integer multiplication, which have been obtained mainly in [4] but were stated in this form explicitly in [3].

Lemma 2 ([3, 4]) *Let $Y \subseteq \mathbb{Z}_{2^n}^*$, $1 \leq r \leq n-2$ and $(z_i, z'_i) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$, $1 \leq i \leq s$, where $z_i \neq z'_i$. Then there exists a subset $Y' \subseteq Y$, $|Y'| \geq |Y| - s \cdot 2^{n-r+2}$ such that*

$$\forall y \in Y', 1 \leq i \leq s: \quad 4 \cdot 2^{n-r} \;\leq\; \big((z_i - z'_i)y\big) \bmod 2^n \;\leq\; 2^n - 4 \cdot 2^{n-r}.$$

Lemma 3 ([4]) *Let $X \subseteq \mathbb{Z}_{2^n}$ and $Y \subseteq \mathbb{Z}_{2^n}^*$. If $|X| \cdot |Y| \geq 2^{n+2r+1}$ for some integer $r \geq 0$, then there exists an element $y \in Y$ such that for all $q \in \{0, \dots, 2^r - 1\}$*

$$\exists x \in X: \; q2^{n-r} \;\leq\; (xy) \bmod 2^n \;<\; (q+1)2^{n-r}.$$

The following proposition is obtained in an easy but somewhat technically way from the previous lemma. We state it in such a way that we can use it for the proof of Theorem 2 as well as for the lower bound proof for $(1, +k)$-BPs in the next section. The proof is omitted due to space restrictions.

Proposition 2 *Let $X \subseteq \mathbb{Z}_{2^n}$, $y \in \mathbb{Z}_{2^n}^*$, $z, z_1, \dots, z_k \in \mathbb{Z}_{2^n}$, and $r \in \mathbb{N}$ such that the following two properties are fulfilled.*

1. $\forall q \in \{0, \dots, 2^r - 1\}\, \exists x \in X:$

$$q2^{n-r} \leq (xy) \bmod 2^n < (q+1)2^{n-r}.$$

2. $\exists C \in \{-1, +1\} : \forall 1 \leq i \leq k:$

$$2^{n-1} \;\leq\; \big(C(z_i - z)y\big) \bmod 2^n \;\leq\; 2^n - 4 \cdot 2^{n-r}.$$

Then there exists $x \in X$ *such that for all* $1 \leq i \leq k$ $\mathrm{MUL}(x + z, y) > \mathrm{MUL}(x + z_i, y)$.

Let in the following $r = k(2l + 1) + 2$ and recall that $l \leq (n - 6)/(7k) - 3/7$. It is easy to derive the following inequality.

$$n + 2r + 1 \leq 2n - kl - k(2l + 1) - 1. \qquad (4)$$

We are now ready to describe Bob's strategy. Consider a partial assignment $\gamma_1 \dots \gamma_{i-1}$ obtained after the $(i-1)$th round as well as two assignments α_i and β_i to some x-variables in V_i. Further, let $A_i \subseteq B_{i-1}$ be chosen by Alice. Bob will make his choices in such a way that the following invariants are fulfilled after the ith round.

(i) $|B_i| \geq 2^{n-1-i(2l+1)}$.

(ii) $\forall b \in B_i$:
$2^{n-1} \leq \left(b\left|\gamma_i^*\right| - b|\gamma_i|\right) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r}$.

Obviously, the proof of the following two claims completes the proof of Theorem 2.

Claim 1 *Bob can achieve (i) and (ii) after the* i*th round for* $i = 1, \dots, k$.

Claim 2 *If (i) and (ii) hold after the* i*th round for* $i = 1, \dots, k$*, then Bob wins the game.*

Proof of Claim 1. For $i = 0$, we have $|B_0| = \left|\mathbb{Z}_{2^n}^*\right| = 2^{n-1}$. Hence (i) is fulfilled when the game starts or after the 0th round. We show now that if (i) holds after the $(i-1)$th round, then Bob can make his choices in such a way that (i) and (ii) also hold after the ith round.

Let $\alpha_i \neq \beta_i$ and A_i be the choices of Alice in this round. By the rules of the game, $|A_i| \geq \varepsilon |B_{i-1}| = 2^{-(2l-1)}|B_{i-1}|$, and hence because (i) was fulfilled in the previous round

$$|A_i| \geq 2^{n-1-(i-1)(2l+1)-(2l-1)} = 2^{n-i(2l+1)+1}. \qquad (5)$$

Because α_i and β_i are different, obviously $|\alpha_i| \neq |\beta_i|$. We apply Lemma 2 in order to obtain a set $Y' \subseteq A_i$, $\left|Y'\right| \geq |A_i| - 2^{n-r+2}$, such that

$\forall b \in Y'$:
$$4 \cdot 2^{n-r} \leq (b|\alpha_i| - b|\beta_i|) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r}.$$

Note that if $\left(b|\alpha_i| - b|\beta_i|\right) \bmod 2^n \leq 2^{n-1}$, then $\left(b|\beta_i| - b|\alpha_i|\right) \bmod 2^n \geq 2^{n-1}$ and vice versa. Hence, Bob may choose $\gamma_i \in \{\alpha_i, \beta_i\}$ such that for at least half of the elements $b \in Y'$ it holds $\left(b\left|\gamma_i^*\right| - b|\gamma_i|\right) \bmod 2^n \geq 2^{n-1}$. Bob then chooses B_i to be exactly the set of these elements in Y', so that we obtain altogether

$\forall b \in B_i$:
$$2^{n-1} \leq \left(b\left|\gamma_i^*\right| - b|\gamma_i|\right) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r},$$

and (ii) holds. Furthermore, we have

$$|B_i| \geq \left|Y'\right|/2 \geq \frac{|A_i| - 2^{n-r+2}}{2}.$$

Note that because $i \leq k$, we obtain from (5) that $|A_i|/2 \geq 2^{n-k(2l+1)} = 2^{n-r+2}$ (recall that $r = k(2l+1)+2$). Hence,

$$|B_i| \geq \frac{|A_i| - |A_i|/2}{2} = |A_i|/4 \overset{(5)}{\geq} 2^{n-1-i(2l+1)}.$$

Therefore, also (i) is fulfilled. ∎

Proof of Claim 2. Let the game have been played k rounds such that (i) and (ii) hold after each round. We let $z = |\gamma_1 \dots \gamma_k|$ and $z_j = \left|\gamma_1 \dots \gamma_{j-1}\gamma_j^*\gamma_{j+1} \dots \gamma_k\right|$.

We have to show how to choose an element $b \in B_k$ and a partial assignment a to the variables in $\mathcal{X}_n' \setminus (V_1 \cup \dots \cup V_k)$ such that for all $1 \leq j \leq k$

$$\mathrm{MUL}'(|\gamma_1 \dots \gamma_k a|, b) > \mathrm{MUL}'(|\gamma_1 \dots \gamma_{j-1}\gamma_j^*\gamma_{j+1} \dots \gamma_k a|, b).$$

Because of Fact 2, we can write this inequality as

$$\mathrm{MUL}'(z + |a|, b) > \mathrm{MUL}'(z_j + |a|, b). \qquad (6)$$

Let X' be the set of all integers $|a|$, where a is an assignment to the variables in $\mathcal{X}_n' \setminus (V_1 \cup \dots \cup V_k)$. Obviously, $\left|X'\right| \geq 2^{n-kl}$. Furthermore, it is $|B_k| \geq 2^{n-1-k(2l+1)}$, because invariant (i) is fulfilled after the kth round. Hence, we get

$$\left|X'\right| \cdot |B_k| \geq 2^{2n-kl-k(2l+1)-1} \overset{(4)}{\geq} 2^{n+2r+1}.$$

Applying Lemma 3, we can choose an element $b \in B_k$ such that

$$\forall q \in \left\{0, \dots, 2^r - 1\right\} \exists x \in X' : q2^{n-r} \leq (xb) \bmod 2^n < (q+1)2^{n-r}. \qquad (7)$$

Let this b be fixed from now on. Recall that $z = |\gamma_1 \dots \gamma_k|$ and $z_j = \left|\gamma_1 \dots \gamma_{j-1}\gamma_j^*\gamma_{j+1} \dots \gamma_k\right|$. Hence, by Fact 2 $z_j - z = \left|\gamma_j^*\right| - \left|\gamma_j\right|$, and thus because of (ii) and because $b \in B_k \subseteq B_j$

$$\forall 1 \leq j \leq k : 2^{n-1} \leq (z_j b - zb) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r}. \qquad (8)$$

By (7) and (8) all preconditions of Proposition 2 are fulfilled, and therefore there exists an element $x \in X'$ such that for all $1 \leq j \leq k$

$$\mathrm{MUL}(z + x, b) > \mathrm{MUL}(z_j + x, b).$$

Since b is odd, the same result holds also for MUL′ and thus (6) is proven for all $1 \leq j \leq k$. ∎

4. Lower Bounds for $(1, +k)$-BPs

Similarly as in Section 3 we define now a property of Boolean functions, which allows us to prove exponential lower bounds in the $(1, +k)$-BP model. As in the previous section the main difficulty lies in the task to cope with two different variable types. Moreover, it seems not possible to prove something similar as Lemma 1 for $(1, +k)$-BPs. Therefore, the property we describe now differs from the one used in the previous section. We describe it though, again by a game played by two players, Alice and Bob.

Let $f \in B_{n+m}$. The game we describe below is called $\mathcal{G}^+(f, k, l, t)$, where k and l are positive integers and $t \geq k$ is the number of rounds ($tl \leq m + n$).

Alice starts the ith round ($1 \leq i \leq t$) by choosing a set $V_i \subseteq (\mathcal{X}_n \cup \mathcal{Y}_m) \setminus (V_1 \cup \ldots \cup V_{i-1})$ such that $|V_i| \leq l$. Further, Alice chooses two partial assignments $\alpha_i \neq \beta_i$ with support V_i which differ only in their settings to variables of one type, i.e. $D(\alpha_i, \beta_i) \subseteq \mathcal{X}_n$ or $D(\alpha_i, \beta_i) \subseteq \mathcal{Y}_m$. After that, Bob chooses $\gamma_i \in \{\alpha_i, \beta_i\}$. The game is finished after the tth round.

Let $c = (\gamma_1 \ldots \gamma_t)$ and c_j be the partial assignment obtained from c by replacing γ_j with γ_j^*, where as in the previous sections γ_j^* is the element in $\{\alpha_j, \beta_j\} \setminus \{\gamma_j\}$. Bob wins the game, if there exist a set $I \subseteq \{1, \ldots, t\}$ of at least k indices and a partial assignment λ to the variables in $(\mathcal{X}_n \cup \mathcal{Y}_n) \setminus (V_1 \cup \ldots \cup V_t)$ such that

$$\forall j \in I: \; f|_c(\lambda) \neq f|_{c_j}(\lambda).$$

Otherwise, Alice wins.

Theorem 3 *Let $f \in B_{n+m}$, $l \geq 4$, and $tl \leq n + m$. If Bob has a winning strategy for the game $\mathcal{G}^+(f, k, l, t)$, then any $(1, +(k-1))$-BP representing f has a size of at least $2^{l/4-2} + 2$.*

In order to prove the theorem by describing a winning strategy for Alice, we use the following lemma.

Lemma 4 *Let $f \in B_{n+m}$ and $4 \leq l \leq n + m$. If G is a deterministic BP representing f with at most $2^{l/4-2}+1$ nodes, then there exist two partial assignments α, β with support V such that the following three conditions are fulfilled.*

1. $|V| \leq l$.
2. $D(\alpha, \beta) \subseteq \mathcal{X}_n \;\vee\; D(\alpha, \beta) \subseteq \mathcal{Y}_n$.
3. $(\alpha, \beta) \in L_G$.

Proof. Let $l' = \lfloor (l+3)/4 \rfloor \geq l/4$. We denote by $p_{a,b}$ the computation path of the input $(a, b) \in X_n \times Y_m$. On each path $p_{a,b}$ we determine the edge $e_{a,b}$ through which the path passes after either exactly l' x-variables or exactly l' y-variables have been tested. If $p_{a,b}$ is a path testing less than l' x- and less than l' y-variables, then $e_{a,b}$ is the last edge on the path (i.e. the edge pointing to a sink).

Since G has two sinks with out-degree 0 and $|G| - 2$ internal nodes with out-degree 2, G contains exactly $2(|G| - 2)$ edges. Hence, by the pigeon hole principle, there exists an edge e such that for at least

$$\frac{|X||Y|}{2(|G|-2)} > \frac{2^{n+m}}{2 \cdot 2^{l/4-2}} \geq 2^{n+m-l/4+1} \geq 2^{n+m-l'+1}$$

inputs (a, b) it holds $e_{a,b} = e$. Furthermore, there exists one type of variables – w.l.o.g. the y-variables – such that at least half of the inputs (a, b) with $e_{a,b} = e$ test on their paths to the edge e less than l' y-variables. Considering only these inputs, we can then find (again by the pigeon hole principle) an element $b \in Y_m$ and a set $X' \subseteq X_n$, $|X'| > 2^{n-l'}$, such that for all $a \in X'$ it holds $e_{a,b} = e$ and all paths $p_{a,b}$ test less than l' y-variables before edge e is reached. Let this b be fixed from now on.

Now assume first that there exists an element $a \in X'$ such that $p_{a,b}$ tests also less than l' x-variables before edge $e = e_{a,b}$ is reached. Then by definition, e points to a sink. Let τ be the partial assignment which defines exactly the path $p_{a,b}$. (I.e. the ith bit of τ is set to the ith bit of (a, b) if this bit is tested on the path $p_{a,b}$ and set to $*$ otherwise.) Since less than l' x- and less than l' y-variables are tested on $p_{a,b}$, we have $S(\tau) \leq 2(l' - 1)$. We choose an arbitrary x-variable $x_i \notin S(\tau)$ and let α and β be the partial inputs extending τ by the additional assignment $x_i = 0$ and $x_i = 1$, respectively. Clearly, we then have $\alpha \neq \beta$, $S(\alpha) = S(\beta)$ and the only variable in $D(\alpha, \beta)$, namely x_i, is tested on no path from e to a sink. Hence, $(\alpha, \beta) \in L_G(e)$. Furthermore, letting $V = S(\alpha)$ we get $|V| \leq 2l' - 1 \leq l$, and since finally $D(\alpha, \beta) = \{x_i\} \subseteq \mathcal{X}_n$, the claim is proven for this case.

We now consider the other case, namely that for all $a \in X'$ the path $p_{a,b}$ tests exactly l' x-variables before it reaches edge e. Assume that all $a \in X'$ differ in no variable being tested on a path $p_{a,b}$ before edge e is reached. Then clearly all paths $p_{a,b}$ are equal before edge e, and hence all $a \in X'$ have the same support S such that $|S \cap \mathcal{X}_n| = l'$. But this is impossible, because there are at most $2^{n-l'}$ possible assignments to the x-variables not in S, while we have $|X'| > 2^{n-l'}$.

Hence, there exist two different elements $a', a'' \in X'$ such that there is at least one variable $x_i \in D(a', a'')$ being tested on $p_{a',b}$ and $p_{a'',b}$ before edge e. Let V' be the set of variables being tested on $p_{a',b}$ and V'' be the set of variables being tested on $p_{a'',b}$ before edge e is reached on these paths. Note that V' and V'' may contain different x- and y-variables, but they have at least the variable x_i in common. Because by construction $|V'|, |V''| \leq 2l' - 1$, we obtain for $V := V' \cup V''$ that $|V| \leq 4l' - 3$. We choose α to be the partial assignment with support V which equals (a', b) on the variables in V' and equals (a'', b) on the variables in

$V \setminus V'$. Similarly, we choose β to be the partial assignment with support V which equals (a'', b) on the variables in V'' and equals (a', b) on the variables in $V \setminus V''$. This way, α and β are different partial assignments with support V, and have an equal setting to all variables in $V \setminus (V' \cap V'')$. Hence, $D(\alpha, \beta) \subseteq V' \cap V''$. Furthermore, α and β induce the sub-paths of $p_{a',b}$ and $p_{a'',b}$, respectively, which lead from the source to edge e. Since by construction, each variable in $D(\alpha, \beta) \subseteq V' \cap V''$ is tested on both paths before edge e is reached, we obtain $(\alpha, \beta) \in L_G(e)$. Furthermore, α and β are defined for the same b, and hence we have $D(\alpha, \beta) \subseteq \mathcal{X}_n$. Together with $|V| \leq 4l' - 3 \leq l$, all three conditions of the claim are fulfilled. ∎

We can now use this lemma in order to prove Theorem 3.

Proof of Theorem 3. Assume that G is a $(1, +(k-1))$-BP for the function f and that $|G| \leq 2^{l/4-2} + 1$. We describe a winning strategy for Alice.

In the first round, Alice chooses α_1, β_1 with support V_1 according to Lemma 4. Hence, $|V_1| \leq l$, and α_1 and β_1 differ only in their settings to the x-variables or only in their settings to the y-variables and are therefore chosen in accordance with the rules of the game. Furthermore, $(\alpha_1, \beta_1) \in L_G$. After that, Bob chooses $\gamma_1 \in \{\alpha_1, \beta_1\}$.

Consider now the $(i+1)$th round, $i+1 \leq t$. We let $G|_{\gamma_1 \ldots \gamma_i}$ be the $(1, +(k-1))$-BP obtained from G by restricting it to the partial assignment $\gamma_1 \ldots \gamma_i$. (See the proof of Theorem 1 for how to do this.) Note that the function $f|_{\gamma_1 \ldots \gamma_i}$ represented by $G|_{\gamma_1 \ldots \gamma_i}$ still depends on at least l variables. Applying Lemma 4, Alice can find analogously to the first round $\alpha_{i+1}, \beta_{i+1}$ such that $D(\alpha_{i+1}, \beta_{i+1})$ is a subset of $\mathcal{X}_n$ or of $\mathcal{Y}_m$ and $(\alpha_{i+1}, \beta_{i+1}) \in L_{G|_{\gamma_1 \ldots \gamma_i}}$. It is easy to see that then $(\gamma_1 \ldots \gamma_i \alpha_{i+1}, \gamma_1 \ldots \gamma_i \beta_{i+1}) \in L_G$ and of course $D(\gamma_1 \ldots \gamma_i \alpha_{i+1}, \gamma_1 \ldots \gamma_i \beta_{i+1})$ is also a subset of either $\mathcal{X}_n$ or $\mathcal{Y}_m$.

Altogether, we obtain that Alice can make her choices in such a way that after t rounds the following holds (recall the definition of c and c_j at the description of the game).

$$\forall 1 \leq j \leq t : \ (c, c_j) \in L_G. \tag{9}$$

Let then I be a set of k indices in $\{1, \ldots, t\}$ and λ be an arbitrary assignment to the variables in $(\mathcal{X}_n \cup \mathcal{Y}_m) \setminus (V_1 \cup \ldots \cup V_t)$. We consider the computation path p of the input $c\lambda$ $(= \gamma_1 \ldots \gamma_t \lambda)$. Since at most $k-1$ variables can be tested more than once on p, and since the k variable sets V_i, $i \in I$, are all disjoint, there is an index $j \in I$ such that each variable in V_j is tested at most once on p. But clearly $D(c, c_j) \subseteq V_j$, and hence Fact 1 together with (9) tells us that $f|_c(\lambda) = f|_{c_j}(\lambda)$. Therefore, Alice has won the game. ∎

Using this technique, one may finally obtain an exponential lower bound for MUL in the $(1, +k)$-BP model.

Theorem 4 *Any deterministic* $(1, +k)$*-BP for* MUL *has a size of* $\Omega\left(2^{\frac{n}{48(k+1)}}\right)$.

Proof of Theorem 4. We prove the theorem for the sub-function $\mathrm{MUL}'' := \mathrm{MUL}|_{x_0=1, y_0=1}$, which computes the middle bit of the product of two odd n-bit integers. Let $k \geq 1$, $l = \lfloor n/(12k) - 4/3 \rfloor$, $t = 4k$ and $r = 4k + \lceil \log k \rceil + 5$. We show that Bob has a winning strategy for the game $\mathcal{G}^+(\mathrm{MUL}'', k, l, t)$. The theorem then follows right away from Theorem 3.

We even show that Bob wins the game independently of his choices. Let all the partial assignments γ_i, γ_i^* with support V_i $(1 \leq i \leq t)$ be somehow determined according to the rules of the game, and let c and c_i be defined as usual. Since $D(c, c_i) = D(\gamma, \gamma_i)$ is either a subset of $\mathcal{X}_{n-1}$ or of $\mathcal{Y}_{n-1}$, there exists a set J of at least $t/2 = 2k$ indices in $\{1, \ldots, t\}$ such that all sets $D(c, c_j)$, $j \in J$, contain only variables of one and the same type. Hence, assume w.l.o.g. that $D(c, c_j) \subseteq \mathcal{X}_n$ for all $j \in J$.

We let $z \in \mathbb{Z}_{2^n}^*$ be the odd integer obtained by fixing $z_0 = 1$ and setting all bits z_i, $1 \leq i < n$, according to the setting of the x-variables x_i in c $(= \gamma_1 \ldots \gamma_t)$ if $x_i \in S(c)$ and letting $z_i = 0$ if $x_i \notin S(c)$. In an analogous way we obtain the integers z_j, $j \in J$, from the partial assignments c_j.

Consider now an arbitrary partial assignment $\lambda = (\lambda_x, \lambda_y)$ to the variables in $(\mathcal{X}_{n-1} \cup \mathcal{Y}_{n-1}) \setminus (V_1 \cup \ldots \cup V_t)$. (Here λ_x denotes the partial assignment to the x-variables and λ_y the partial assignment to the y-variables.) Let $x = |\lambda_x|$. The x-integer represented by the input $c\lambda$ then obviously is $z + x$ and the x-integer represented by $c_j\lambda$, $j \in J$, is $z_j + x$. Similarly, λ_y together with the setting of the y-variables defined by c defines a unique integer $y \in \mathbb{Z}_{2^n}^*$. Note that for $j \in J$ this integer is the same, if we replace c with c_j, because then γ_j and γ_j^* do not differ in the y-variables. Hence, the integers x and y are uniquely defined by the choice of (λ_x, λ_y).

Let X' be the set of all the integers x and Y^* be the set of all the integers y that can be obtained this way from the partial assignments (λ_x, λ_y) with support $(\mathcal{X}_{n-1} \cup \mathcal{Y}_{n-1}) \setminus (V_1 \cup \ldots \cup V_t)$. In order to show that Bob has won the game, we have to prove that there exists a partial assignment $\lambda = (\lambda_x, \lambda_y)$ with support $(\mathcal{X}_{n-1} \cup \mathcal{Y}_{n-1}) \setminus (V_1 \cup \ldots \cup V_t)$ and a set $I \subseteq J$, $|I| \geq k$, such that

$$\forall j \in I : \ \mathrm{MUL}''|_c(\lambda) \ \neq \ \mathrm{MUL}''|_{c_j}(\lambda).$$

In our notation, this is equivalent to showing that there exists a pair $(x, y) \in X' \times Y^*$ and a set $I \subseteq J$, $|I| \geq k$, such that

$$\begin{aligned} &|I| \geq k \quad \text{and} \\ &\forall j \in I : \ \mathrm{MUL}(z + x, y) \ \neq \ \mathrm{MUL}(z_j + x, y). \end{aligned} \tag{10}$$

Note that $V_1 \cup \ldots \cup V_t$ contains at most tl variables. Since λ_x and λ_y are partial assignments to the x- and y- variables not in $V_1 \cup \ldots \cup V_t$, we have $\left|X'\right|, |Y^*| \geq 2^{n-1-tl}$ and $\left|X'\right| \cdot |Y^*| \geq 2^{2n-2-tl}$. We consider now the $|J| = 2k$ pairs (z, z_j), $j \in J$. By Lemma 2, there exists a subset $Y' \subseteq Y^*$ such that

$$\forall y \in Y',\ j \in J: \quad 4 \cdot 2^{n-r} \leq \big((z_j - z)y\big) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r}, \tag{11}$$

and $\left|Y'\right| \geq |Y^*| - |J| \cdot 2^{n-r+2}$. Recalling that $r = 4kl + \lceil \log k \rceil + 5$ and $t = 4k$, we obtain

$$|J| \cdot 2^{n-r+2} = 2k \cdot 2^{n-4kl-\lceil \log k \rceil - 3} \leq 2^{n-tl-2} \leq \left|Y^*\right|/2.$$

Hence, $\left|Y'\right| \geq |Y^*| - |Y^*|/2 \geq |Y^*|/2$. Using $l \leq n/(12k) - 4/3$, this yields

$$\begin{aligned} \left|X'\right| \cdot \left|Y'\right| &\geq \left|X'\right| \cdot \left|Y^*\right|/2 \geq 2^{2n-tl-3} = 2^{2n-4kl-3} \\ &= 2^{(5/3)n+(16/3)k-3} \geq 2^{(5/3)n+7/3}. \end{aligned}$$

(For the last inequality we have used $k \geq 1$.) Using this inequality, we obtain further

$$\begin{aligned} 2^{n+2r+1} &= 2^{n+8kl+2\lceil \log k \rceil + 11} \\ &= 2^{(5/3)n-(32/3)k+2\lceil \log k \rceil + 11} \leq \left|X'\right| \cdot \left|Y'\right|. \end{aligned}$$

Hence, we may apply Lemma 3, and obtain an element $y \in Y'$ such that

$$\forall q \in \left\{0, \ldots, 2^r - 1\right\} \exists x \in X': \quad q2^{n-r} \leq (xy) \bmod 2^n < (q+1)2^{n-r}. \tag{12}$$

Let this y be fixed from now on.

Note that for any z_j, $j \in J$, we have either $2^{n-1} \leq \big((z_j - z)y\big) \bmod 2^n$ or $2^{n-1} \leq \big((z - z_j)y\big) \bmod 2^n$. Hence, either for $C = -1$ or for $C = 1$ we obtain for at least half of the indices $j \in J$ that $2^{n-1} \leq \big(C(z_j - z)y\big) \bmod 2^n$. Let I be the set of these indices, $|I| \geq |J|/2$. Invoking (11), we get then altogether

$$\forall j \in I: \quad 2^{n-1} \leq \big(C(z_j - z)y\big) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-r}. \tag{13}$$

We may apply now Proposition 2, noting that due to (12) and (13) all preconditions of this proposition are fulfilled for the elements $z_j \in I$. Hence, we obtain that there exists $x \in X$ such that $\mathrm{MUL}(x + z, y) > \mathrm{MUL}(x + z_j, y)$ for all $j \in I$. Since $|I| \geq |J|/2 \geq k$, this proves (10). ∎

Acknowledgment

I thank Ingo Wegener for helpful comments on this text and Beate Bollig for fruitful discussions on Branching Programs and Integer Multiplication.

References

[1] A. Andreev, J. Baskakov, A. Clementi, and J. Rolim. Small pseudo-random sets yield hard functions: New tight explict lower bounds for branching programs. In *Proc. of 26th ICALP*, volume 1644 of *LNCS*, pages 179–189, 1999.

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

[3] 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, ECCC, 2001.

[4] B. Bollig and P. Woelfel. A read-once branching program lower bound of $\Omega(2^{n/4})$ for integer multiplication using universal hashing. In *Proc. of 33rd ACM STOC*, pages 419–424, 2001.

[5] A. Borodin, A. Razborov, and R. Smolensky. On lower bounds for read-k-times branching programs. *Comput. Complex.*, 3:1–18, 1993.

[6] 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(2):205–213, 1991.

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

[8] J. Jain, J. Bitner, D. S. Fussell, and J. A. Abraham. Functional partitioning for verification and related problems. In *Brown MIT VLSI Conference*, pages 210–226, 1992.

[9] S. Jukna and A. Razborov. Neither reading few bits twice nor reading illegally helps much. *Discrete Applied Mathematics*, 85:223–238, 1998.

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

[11] P. Savický and D. Sieling. A hierarchy result for read-once branching programs with restricted parity nondeterminism. In *Proc. of 25th MFCS*, volume 1893 of *LNCS*, pages 650–659, 2000.

[12] J. Simon and M. Szegedy. A new lower bound theorem for read-only-once branching programs and its applications. In *Advances in Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 183–193. AMS, 1993.

[13] I. Wegener. On the complexity of branching programs and decision trees for clique functions. *J. of the ACM*, 35(2):461–471, 1988.

[14] P. Woelfel. New bounds on the OBDD-size of integer multiplication via universal hashing. In *Proc. of 18th STACS*, volume 2010 of *LNCS*, pages 563–574, 2001.

[15] P. Woelfel. A lower bound technique for restricted branching programs and applications. In *Proc. of 19th STACS*, volume 2285 of *LNCS*, pages 431–442, 2002.

[16] S. Žák. An exponential lower bound for one-time-only branching programs. In *Proc. of 11th MFCS*, volume 176 of *LNCS*, pages 562–566, 1984.