A Read-Once Branching Program Lower Bound of $\Omega(2^{n/4})$ for Integer Multiplication Using Universal Hashing

Beate Bollig*
FB Informatik, LS2
Univ. Dortmund
44221 Dortmund, Germany
bollig@ls2.cs.uni-dortmund.de

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. Especially read-once branching programs (BP1s) have been studied intensively. Exponential lower bounds on the BP1 complexity of explicit functions have been known for a long time. Nevertheless, the proof of exponential lower bounds on the read-once branching program size of selected functions is sometimes difficult. Motivated by the applications the BP1 complexity of fundamental functions is of interest. It took quite a long time until Ponzio [16, 17] was able to prove a bound of $2^{\Omega(\sqrt{n})}$ for integer multiplication. Combining results and methods for universal hashing with lower bound techniques for BP1s a lower bound of $\Omega(2^{n/4})$ on the size of BP1s for integer multiplication is presented in this paper.

Keywords: Computational complexity, lower bounds, read-once branching programs, integer multiplication

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 $X_n = \{x_0, \ldots, x_{n-1}\}$ is a directed acyclic graph with one source and two sinks labeled by the constants 0 or 1, respectively. Each non-sink node (or inner node) is labeled by a Boolean variable and has two outgoing edges, one labeled by 0 and the other by 1. Each node v represents a Boolean function $f_v : \{0,1\}^n \to \{0,1\}$ in the following way. In order to evaluate $f_v(a)$, $a = a_{n-1} \cdots a_0 \in \{0,1\}^n$, start at v. After reaching an x_i-node choose the outgoing edge with label a_i until a sink is reached. The label of this sink defines $f_v(a)$. A BP with source q represents the Boolean function f_q. The computation path for the input a in a BP G is the sequence of nodes visited during the evaluation of a in G.

The size of a branching program G is the number of its nodes. $\mathrm{BP}(f)$ denotes the size of the smallest BP for a function f. The length of a branching program is the maximum length of a path from the source to one of the sinks.

The branching program size of a Boolean function f is known to be a measure for the space complexity of non-uniform Turing machines and is known to lie between the circuit size of f and its $\{\wedge, \vee, \neg\}$-formula size (see, e.g., [22]). Hence, one is interested in exponential lower bounds for more and more general types of BPs (for one of the latest breakthroughs for linear depth BPs see [1] and [6]). 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 oblivious if the node set can be partitioned into levels such that edges lead from lower to higher levels and all inner nodes of one level are labeled by the same variable.

Read-once branching programs (BP1s) are the most general model where each graph-theoretical path is the computation path for some input. Exponential lower bounds on the read-once branching program complexity of explicit functions have been known for a long time. The first bounds are of size $2^{\Omega(\sqrt{n})}$, where n is the number of variables ([23] and [21]). The later development follows different directions.

One aim was to prove lower bounds as large as possible. The first strongly exponential lower bounds of size $2^{\Omega(n)}$ are contained in [2] with a very small constant and in [15] with a constant of moderate size. The result for the Triangle-Parity function considered in [2] has been improved in [20]. Savický and Žák [18] have proven a lower bound of size $2^{n-3\sqrt{n}}$ which is almost optimal due to a general $O(2^n/n)$ upper bound. Finally, Andreev, Baskakov, Clementi, and Rolim [5] have proven an optimal bound of $2^{n-O(\log n)}$.

*Supported in part by DFG grant We 1066/9.

STOC'01, July 6-8, 2001, Heronissos, Crete, Greece.

Besides this complexity theoretical viewpoint people have used branching programs in applications where the complexity of fundamental functions is of interest. Representations of Boolean functions which allow efficient algorithms for many operations, in particular synthesis (combine two functions by a binary operation) and equality test (do two representations represent the same function?) are necessary. Bryant [9] introduced ordered binary decision diagrams (OBDDs) which are up to now the most popular representation for formal circuit verification. OBDDs can be seen as oblivious read-once branching programs. Unfortunately, several important and also quite simple functions have exponential OBDD size. Therefore, more general representations with good algorithmic behavior are necessary. Sieling and Wegener [19] have shown how read-once branching programs can be used for verification. They define so-called graph orderings. An arbitrary read-once branching program G is ordered with respect to a suitably chosen graph ordering. Now, there exist efficient algorithms for many operations for BP1s obeying the same ordering.

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

Definition 3. Integer multiplication $\mathrm{MULT}_n{:}\{0,1\}^{2n} \to \{0,1\}^{2n}$ is the Boolean function that computes the product of two n-bit integers, i.e., $\mathrm{MULT}_n(x,y) = z_{2n-1}\dots z_0$, where $x = x_{n-1}\dots x_0$ and $y = y_{n-1}\dots y_0$ and $xy = z = z_{2n-1}\dots z_0$. $\mathrm{MULT}_{i,n}$ computes the ith bit of MULT_n.

For some models integer multiplication is a quite simple function. It is contained in NC^1 and even in $TC^{0,3}$ (polynomial-size threshold circuits of depth 3) but neither in AC^0 (polynomial-size $\{\vee,\wedge,\neg\}$-circuits of unbounded fan-in and constant depth) nor in $TC^{0,2}$ [14]. Until now it is open whether there exist multiplication circuits of linear size. For OBDDs Bryant [10] has presented an exponential lower bound of size $2^{n/8}$ for $\mathrm{MULT}_{n-1,n}$. Incorporating Ramsey theoretic arguments of Alon and Maass [4] and using the rank method of communication complexity Gergov [13] has extended the lower bound to arbitrary linear-length oblivious BPs. It took quite a long time until Ponzio [16, 17] was able to prove an exponential lower bound of size $2^{\Omega(\sqrt{n})}$ for $\mathrm{MULT}_{n-1,n}$ for BP1s. Bollig [7] has presented the first (not strongly) exponential lower bound on the size of $\mathrm{MULT}_{n-1,n}$ for a nondeterministic nonoblivious branching program model. But this model, so-called nondeterministic tree-driven read-once branching programs, is very restricted such that methods known from communication complexity can be used for this lower bound. Until now exponential lower bounds on the size of $\mathrm{MULT}_{n-1,n}$ for general nondeterministic read-once branching programs or read k times branching programs with $k \geq 2$ are unknown. One reason for the difficulties in proving such lower bounds arise from the fact that integer multiplication can express many different shifting and adding combinations such that the effects of partial assignments and therefore the subfunctions are not easy to analyze.

Only recently Woelfel [27] has improved Bryant's OBDD lower bound up to $\Omega(2^{n/2})$. For this result he has used a new lower bound method which highly relies on a recently found universal family of hash functions [25]. Improving the lower bound on the size of read-once branching programs for multiplication to the first strongly exponential lower bound on the size of a nonoblivious branching program model we use a similar approach but the methods used are technically more involved. Here, the combination of results and methods for universal hashing with lower bound techniques for BP1s in order to prove large lower bounds is done for the first time. The analysis seems to be much easier than Ponzio's counting technique. Furthermore, our lower bound for $\mathrm{MULT}_{n-1,n}$ is of size $\Omega(2^{n/4})$, more precisely $2^{\lfloor (n-9)/4 \rfloor}$, and therefore even larger than Bryant's lower bound on the OBDD size.

The rest of the paper is organized as follows. In the next section, we present the lower bound method for read-once branching programs due to Simon and Szegedy [20]. In Section 3, we describe a property of universal hash families which turns out to be very useful for our purpose. Afterwards, we define a class of hash functions which is universal and closely related to our multiplication problem. Using facts from number theory we make important observations about integer multiplication. Finally, combining our observations we prove the strongly exponential lower bound on the size of read-once branching programs for $\mathrm{MULT}_{n-1,n}$ in Section 4.

2. A LOWER BOUND METHOD FOR READ-ONCE BRANCHING PROGRAMS

Simon and Szegedy [20] have presented a general method which summarizes most of the known lower bound techniques for read-once branching programs. Here, we present their main lemma in a slightly simpler form which has also been used by Ponzio [16, 17].

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function defined on n variables $X_n = \{x_0, \dots, x_{n-1}\}$. Let $\mathcal{F}$ be a filter on X_n, that means $\mathcal{F} \subseteq 2^{X_n}$ and $\mathcal{F}$ is closed upward, i.e., if $S \in \mathcal{F}$ then all supersets of S are in $\mathcal{F}$. A subset B of X_n is said to be in the *boundary* of $\mathcal{F}$ if $B \notin \mathcal{F}$ but $(B \cup \{x_i\}) \in \mathcal{F}$ for some x_i, $0 \leq i \leq n-1$. Replacing the variables of $\overline{B} := X_n \setminus B$ with constants we induce a subfunction on B.

LEMMA 1 (SIMON AND SZEGEDY [20]). *If for any B in the boundary of $\mathcal{F}$, at most $2^{|\overline{B}|}/L$ assignments to $\overline{B}$ induce the same subfunction on B, then any read-once branching program computing f has a size of at least L.*

The idea is the following one. Using the boundary of $\mathcal{F}$ a cut through the BP1 can be defined. This cut consists of so-called frontier edges and each computation path uses exactly one of those edges. Each frontier edge can be associated with a boundary set B and a computation path using a given frontier edge is uniquely defined by a setting to $\overline{B}$. Two paths use the same frontier edge only if the two corresponding settings to $\overline{B}$ induce the same subfunction. Since each frontier edge allows only a fraction of $1/L$ of the inputs to pass through it, there has to be at least L of those edges. Having fan-out 2 and only one source the BP1 has at least L nodes.

In the next section we provide tools to distinguish different subfuctions of $\mathrm{MULT}_{n-1,n}$.

3. UNIVERSAL HASHING AND INTEGER MULTIPLICATION

The concept of universal hashing introduced by Carter and Wegman in 1979 has been proven to be very successful in

a large number of applications, which range from complexity theoretical investigations over message authentication to standard applications like dictionary implementations or integer sorting. Universal hash families are usually defined by using the following notation. Let $\mathcal{H}$ be a family of hash functions $U \to R$. U and R are called *universe* and *range*, respectively. For arbitrary $x, x' \in U$ and $h \in \mathcal{H}$, we define

$$\delta_h(x,x') \ := \ \begin{cases} 1 & \text{if } x \neq x' \text{ and } h(x) = h(x'), \\ 0 & \text{otherwise.} \end{cases}$$

If h, x, and x' are replaced in $\delta_h(x,x')$ by sets, then the sum is taken over the elements from these sets, e.g., for $H \subseteq \mathcal{H}$, $V \subseteq U$, and $x \in U$

$$\delta_H(x,V) \ := \ \sum_{h \in H} \sum_{x' \in V} \delta_h(x,x').$$

Definition 4. A family $\mathcal{H}$ of hash functions $U \to R$ is universal if for any $x, x' \in U$ with $x \neq x'$

$$\delta_{\mathcal{H}}(x,x') \ \leq \ \frac{|\mathcal{H}|}{|R|}.$$

The property of universal hash families which is crucial for our lower bound proof can be described in the following way. If we choose a subset of the universe and this subset is not too small with respect to the size of the range, then the number of hash functions restricted on the chosen subset of the universe whose image is unequal to the whole range is small. Later this observation will be helpful to separate different subfunctions of $\mathrm{MULT}_{n-1,n}$. (A simpler version of this lemma has already been known (see, e.g., [3]).)

For a function $h : U \to R$ and a subset $V \subseteq U$ we define

$$h(V) \ := \ \{y \in R \mid \exists x \in V : h(x) = y\}.$$

LEMMA 2. *Let $\mathcal{H}$ be a universal family of hash functions $U \to R$, $r := |R|$, $V \subseteq U$, and $v := |V|$. Then it follows that*

$$\frac{|\{h \in \mathcal{H} \mid h(V) \neq R\}|}{|\mathcal{H}|} \ \leq \ \frac{(r-1)^2}{v}.$$

PROOF. Since $\mathcal{H}$ is universal, we obtain

$$\begin{aligned} \delta_{\mathcal{H}}(V,V) \ &= \ \sum_{x,x' \in V} \delta_{\mathcal{H}}(x,x') \ \leq \ \sum_{\substack{x,x' \in V \\ x \neq x'}} \frac{|\mathcal{H}|}{r} \\ &= \ \frac{|\mathcal{H}|}{r} v(v-1). \end{aligned}$$

Now, we define $F := \{h \in \mathcal{H} \mid h(V) \neq R\}$. We know by definition that

$$\forall h \in F \ \exists y_h \in R \ \forall x \in V : \ h(x) \neq y_h.$$

First, we prove two lower bounds for $\delta_F(V,V)$ and $\delta_{\mathcal{H} \setminus F}(V,V)$.

$$\begin{aligned} \delta_F(V,V) \ &= \ \sum_{h \in F} \delta_h(V,V) \\ &= \ \sum_{h \in F} \sum_{y \in R \setminus \{y_h\}} |h^{-1}(y) \cap V| \left(|h^{-1}(y) \cap V| - 1\right) \\ &\geq \ \sum_{h \in F} \sum_{y \in R \setminus \{y_h\}} \frac{v}{r-1} \left(\frac{v}{r-1} - 1\right) \\ &= \ |F| v \left(\frac{v}{r-1} - 1\right). \end{aligned}$$

In a similar way we obtain $\delta_{\mathcal{H} \setminus F}(V,V) \geq |\mathcal{H} \setminus F| v \left(\frac{v}{r} - 1\right)$.

Using the fact that $\delta_{\mathcal{H}}(V,V) = \delta_F(V,V) + \delta_{\mathcal{H} \setminus F}(V,V)$ it follows that

$$\frac{|\mathcal{H}|}{r} v(v-1) \ \geq \ |F| v \left(\frac{v}{r-1} - 1\right) + |\mathcal{H} \setminus F| v \left(\frac{v}{r} - 1\right).$$

Since $|\mathcal{H} \setminus F|$ equals $|\mathcal{H}| - |F|$ we get

$$|\mathcal{H}| \left(\frac{v-1}{r} - \left(\frac{v}{r} - 1\right)\right) \ \geq \ |F| \left(\frac{v}{r-1} - 1 - \left(\frac{v}{r} - 1\right)\right)$$

and thus

$$\begin{aligned} & |\mathcal{H}| \left(1 - \frac{1}{r}\right) \ \geq \ |F| v \frac{1}{r(r-1)} \\ \Rightarrow \ & |\mathcal{H}| r(r-1) \left(1 - \frac{1}{r}\right) \ \geq \ |F| v \\ \Rightarrow \ & |\mathcal{H}| (r-1)^2 \ \geq \ |F| v \\ \Rightarrow \ & |F| / |\mathcal{H}| \ \leq \ (r-1)^2 / v. \end{aligned}$$

□

Now, we consider hash functions which map the universe $U := \{1, 3, 5, \ldots, 2^n - 1\}$ to the $(k+1)$-bit range $R_{k+1} := \{0, \ldots, 2^{k+1} - 1\}$, $k \leq n-1$. For $a \in U$ and $b \in B := \{0, \ldots, 2^{n-k-1} - 1\}$ let

$$h_{a,b}^{k+1} : U \to R_{k+1}, \ x \mapsto \left((ax + b) \bmod 2^n\right) \operatorname{div} 2^{n-k-1},$$

where div is the integer division, i.e., $x \operatorname{div} y = \lfloor x/y \rfloor$. If the value of the linear function $ax + b$ is represented by $(y_{2n-1}, \ldots, y_0)$, then $h_{a,b}^{k+1}(x)$ is the integer that is represented by the $k+1$ bits $(y_{n-1}, \ldots, y_{n-k-1})$.

LEMMA 3 (WOELFEL [25, 26]). *The family of hash functions $\mathcal{H} := \left\{h_{a,b}^{k+1} \mid a \in U, b \in B\right\}$ is universal.*

Note, that the corresponding statement in [25] is somewhat more general in the sense that the universe consists of all n-bit integers and that the parameter a may also be chosen from a much smaller subset of U. Similar hash classes have been investigated by Dietzfelbinger [11], Dietzfelbinger, Hagerup, Katajainen, and Penttonen [12], and Woelfel [25, 26].

In the rest of the paper we use the following notation. Let $x \in \{0, \ldots, 2^n - 1\}$. Then $[x]_i$ denotes the ith bit in the binary representation of the integer x, i.e., $x = \sum_{i=0}^{n-1} [x]_i 2^i$. Furthermore, let $[x]_r^l$, $l \geq r$, denote the bits $x_l \ldots x_r$ in the binary representation of x. For the ease of description we use the notation $[x]_r^l = y$ if $(x_l, \ldots, x_r)$ is the binary representation of the integer $y \in \{0, \ldots, 2^{l-r+1} - 1\}$. Sometimes, we identify $[x]_r^l$ with y if the meaning is clear from the context.

Let $U := \{1, 3, \ldots, 2^n - 1\}$ in the rest of the paper. The following facts are not difficult to prove.

FACT 1. *For any $d \in \{0, \ldots, 2^n - 1\}$ with $2^s = \gcd(d, 2^n)$ the following two conditions hold:*

i) $\{(ad) \bmod 2^n \mid a \in U\} \ =$

$$\left\{1 \cdot 2^s, 3 \cdot 2^s, 5 \cdot 2^s, \ldots, \left(2^{n-s} - 1\right) \cdot 2^s\right\}$$

ii) $\forall i \in \{1, 3, \ldots, 2^{n-s} - 1\}$:

$$|\{a \in U \mid (ad) \bmod 2^n = i \cdot 2^s\}| \ = \ 2^s.$$

FACT 2. *Let* $x', x'' \in \{0, \ldots, 2^n - 1\}$ *and* $1 \leq k \leq n$. *Then for some* $\tau, \tau' \in \{0, 1\}$

i) $[x' + x'']_{n-k}^{n-1} \equiv \left([x']_{n-k}^{n-1} + [x'']_{n-k}^{n-1} + \tau\right) \bmod 2^k$

ii) $[|x' - x''|]_{n-k}^{n-1} \equiv \left(\left|[x']_{n-k}^{n-1} - [x'']_{n-k}^{n-1}\right| - \tau'\right) \bmod 2^k$

FACT 3. *Let* $x', x'', b \in \{0, \ldots, 2^n - 1\}$. *Then*

$$\begin{aligned} & b \leq (x' - x'') \bmod 2^n \leq 2^n - b \\ \Leftrightarrow \quad & b \leq \left|x' - x''\right| \leq 2^n - b. \end{aligned}$$

The following lemma is needed for some technical reasons. Later, we will consider two different assignments to some of the x-variables, which lead to two distinct values x' and x''. We have to avoid assignments a to the y-variables for which the value of $\left|[ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right|$ is close to 0 or close to 2^k. The lemma shows that this is only the case for a small fraction of possible assignments to the y-variables.

LEMMA 4. *Let* $A \subseteq U$ *with* $|A| \geq 2^{n-m}$, $m \leq n/2 - 2$, *and* $k = m + 3$. *Then for all distinct* $x', x'' \in U$, *there exists a subset* $A' \subseteq A$ *such that*

i) $|A'| \geq 2^{n-m-1}$,

ii) $\forall a \in A' : \; 2 \leq \left|[ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right| \leq 2^k - 2.$

PROOF. Let $d = (x'' - x') \bmod 2^n$, $2^s = \gcd(d, 2^n)$ and

$$A' = \left\{a \in A \;\middle|\; 4 \cdot 2^{n-k} \leq (ad) \bmod 2^n \leq 2^n - 4 \cdot 2^{n-k}\right\}.$$

We show that the conditions i) and ii) hold for A'.

In order to get a lower bound on the cardinality of A' we first bound the number of $a \in U$ such that $(ad) \bmod 2^n$ is contained in $M' \cup M''$, where

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

Using Fact 1 we obtain

$$\begin{aligned} & \left|\left\{a \in U \;\middle|\; (ad) \bmod 2^n \in M' \cup M''\right\}\right| \\ &= 2^s \cdot \left|\left(M' \cup M''\right) \cap \left\{1 \cdot 2^s, 3 \cdot 2^s, \ldots, (2^{n-s} - 1) \cdot 2^s\right\}\right| \\ &= 2^s \cdot 2 \cdot \left\lfloor \frac{2^{n-k+2}}{2^{s+1}} \right\rfloor \leq 2^{n-k+2}. \end{aligned}$$

Clearly, A' consists of all $a \in A$ such that $(ad) \bmod 2^n \notin M' \cup M''$. Thus, using $k = m + 3$ it follows that

$$\left|A'\right| \geq |A| - 2^{n-k+2} \geq 2^{n-m} - 2^{n-m-1} = 2^{n-m-1}$$

and condition i) is fulfilled.

By definition of A' and using Fact 3 we can conclude that for all $a \in A'$:

$$4 \cdot 2^{n-k} \leq \left|(ax'') \bmod 2^n - (ax') \bmod 2^n\right| \leq 2^n - 4 \cdot 2^{n-k}.$$

(Here we use that

$$\begin{aligned} (ad) \bmod 2^n &\equiv (ax'' - ax') \bmod 2^n \\ &\equiv ((ax'') \bmod 2^n - (ax') \bmod 2^n) \bmod 2^n.) \end{aligned}$$

It follows that

$$4 \leq \left|(ax'') \bmod 2^n - (ax') \bmod 2^n\right| \text{ div } 2^{n-k} \leq 2^k - 4$$

and using Fact 2 we conclude that

$$4 \leq \left|[ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right| \leq 2^k - 3.$$

Therefore, for all $a \in A'$ the condition

$$2 \leq \left|[ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right| \leq 2^k - 2$$

is fulfilled. □

The following lemma is the crucial one for the strongly exponential lower bound for $\text{MULT}_{n-1,n}$.

LEMMA 5. *Let* $A, X \subseteq U$, *where* $|X| = 2^{n-m-1}$ *and* $|A| \geq 2^{n-m}$ *with* $m := \lfloor \frac{n-9}{4} \rfloor$. *Then for all distinct* $x', x'' \in U$ *there exist an element* $a \in A$ *and an element* $x \in X$ *such that*

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

PROOF. Let $k := m + 3$. From Lemma 4 it follows that there exists a subset $A' \subseteq A$, where $|A'| \geq 2^{n-m-1}$ and for all elements $a \in A'$: $2 \leq \left|[ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right| \leq 2^k - 2$.

We consider the universal family of hash functions $\mathcal{H}$ from Lemma 3. Recall that $R_{k+1} = \{0, \ldots, 2^{k+1} - 1\}$, $B = \{0, \ldots, 2^{n-k-1} - 1\}$,

$$h_{a,b}^{k+1} : U \to R_{k+1}, \quad x \mapsto \left((ax + b) \bmod 2^n\right) \text{div } 2^{n-k-1},$$

and $\mathcal{H} = \left\{h_{a,b}^{k+1} \;\middle|\; a \in U, b \in B\right\}$.

Let $r := |R_{k+1}|$ and $F := \left\{h_{a,b}^{k+1} \;\middle|\; a \in A', b \in B\right\}$. Then

$$\frac{|F| \cdot |X|}{|B|} \geq 2^{n-m-1} \cdot 2^{n-m-1} = 2^{2n-2m-2}.$$

Furthermore,

$$\frac{|\mathcal{H}| \cdot r^2}{|B|} = 2^{n-1} \cdot 2^{2k+2} = 2^{n+2m+7}.$$

Therefore, we can conclude that

$$\frac{|F| \cdot |X|}{|\mathcal{H}| \cdot r^2} \geq 2^{n-4m-9} \geq 1.$$

It follows that $|F|/|\mathcal{H}| > (r-1)^2/|X|$. Now, using Lemma 2 we can conclude that there exists a hash function $h_{a,b}^{k+1} \in F$ such that $\left\{h_{a,b}^{k+1}(x) \;\middle|\; x \in X\right\} = R_{k+1}$. Hence, there exist an $a \in A'$ and an element $b \in B$ such that

$$\left\{[ax + b]_{n-k-1}^{n-1} \;\middle|\; x \in X\right\} = \left\{0, \ldots, 2^{k+1} - 1\right\}. \quad (1)$$

Let these a and b be fixed. We define

$$d := \left([ax'']_{n-k}^{n-1} - [ax']_{n-k}^{n-1}\right) \bmod 2^k.$$

Since $a \in A'$, we know that $2 \leq d \leq 2^k - 2$ using Fact 3. W.l.o.g. we assume that $d \leq 2^{k-1}$ (otherwise we exchange x' and x'' in the rest of the proof). Finally, let

$$z := \left(2^{k-1} - 2 - [ax']_{n-k}^{n-1}\right) \bmod 2^k.$$

Now, we choose an element $x \in X$ such that

$$[ax + b]_{n-k-1}^{n-1} = z_{k-1} \ldots z_0 1,$$

where $(z_{k-1}, \ldots, z_0)$ is the binary representation of z. Equation (1) ensures the existence of such an element x. Since $b < 2^{n-k-1}$, it follows that $[ax]_{n-k-1}^{n-1} = z_{k-1} \ldots z_0 q$, where

$q \in \{0,1\}$, and we get $[ax]_{n-k}^{n-1} = z$. By the definition of z we obtain

$$[ax']_{n-k}^{n-1} + [ax]_{n-k}^{n-1} \equiv 2^{k-1} - 2 \mod 2^k,$$

and therefore by Fact 2 we know that

$$[ax' + ax]_{n-k}^{n-1} \equiv 2^{k-1} - 2 + \tau \mod 2^k$$

for some $\tau \in \{0,1\}$. Hence, $[ax' + ax]_{n-1} = 0$. Furthermore, by the definition of d we get

$$\begin{aligned}[ax'']_{n-k}^{n-1} + [ax]_{n-k}^{n-1} &\equiv [ax']_{n-k}^{n-1} + [ax]_{n-k}^{n-1} + d \\ &\equiv 2^{k-1} - 2 + d \bmod 2^k.\end{aligned}$$

Using the assumption that $2 \leq d \leq 2^{k-1}$ we obtain the result

$$2^{k-1} \leq [ax'' + ax]_{n-k}^{n-1} \leq 2^k - 1.$$

by Fact 2. Therefore, $[ax'' + ax]_{n-1} = 1$ and $[ax'' + ax]_{n-1} \neq [ax' + ax]_{n-1}$ which proves the claim. □

4. A STRONGLY EXPONENTIAL LOWER BOUND FOR INTEGER MULTIPLICATION

Now, we combine the lower bound technique for BP1s presented in Section 2 with Lemma 5 in order to prove the first strongly exponential lower bound on the size of a nonoblivious branching program model representing $\mathrm{MULT}_{n-1,n}$.

THEOREM 1. *The* BP1 *size of* $\mathrm{MULT}_{n-1,n}$ *is bounded below by* $2^{\lfloor (n-9)/4 \rfloor}$.

PROOF. We prove the lower bound for the subfunction $\mathrm{MULT}'_{n-1,n}$ of $\mathrm{MULT}_{n-1,n}$, where the variables x_0 and y_0 are set to 1, since then we can use our results from Section 3 for odd numbers x and y. The lower bound for $\mathrm{MULT}_{n-1,n}$ follows immediately. Let $m := \lfloor (n-9)/4 \rfloor$, $X'_n := \{x_1, \ldots, x_{n-1}\}$, and $Y'_n := \{y_1, \ldots, y_{n-1}\}$. The filter $\mathcal{F}$ is defined in the following way:

$$\mathcal{F} := \{V \subseteq X_n \cup Y_n : \quad |V \cap X'_n| > n - m - 1 \text{ and } |V \cap Y'_n| > n - m - 1\}.$$

We will show that for any B in the boundary of $\mathcal{F}$ at most $2^{|\overline{B}|}/2^m$ assignments to the variables of $\overline{B}$ lead to the same subfunction for $\mathrm{MULT}'_{n-1,n}$. Using Lemma 1 we obtain the desired lower bound of size 2^m. Fix any B in the boundary of $\mathcal{F}$ and let $S := \overline{B}$. It is $|S \cap X'_n| = m$ and $|S \cap Y'_n| = m'$ with $m' < m$ (or vice versa). W.l.o.g. there are m x-variables in S.

Our aim is to apply Lemma 5 and to show that two arbitrary assignments p and q to the variables in S which have the same partial assignment a^* to the y-variables of S lead to different subfunctions. First, we define A as the set of all odd integers in $\{0, \ldots, 2^n - 1\}$ whose binary representations $(y_{n-1}, \ldots, y_0)$ agree for the y-variables of S with the partial assignment a^*. The cardinality of A is at least 2^{n-m}. Since the assignments p and q are different, there exists at least one x-variable in S which is fixed in a different way according to p and q. Let x' and x'' be the two odd integers whose binary representations agree for the x-variables in S with p and q, respectively, and for which all x-variables in $X'_n \setminus S$ are set to 0. Let X be the set of all odd integers in $\{0, \ldots, 2^n - 1\}$ whose binary representations have the property that all x-variables of S are set to 0. It follows that $|X| = 2^{n-1-m}$. Using Lemma 5 we obtain the result that p and q lead to different subfunctions. Altogether, we have proven a lower bound of $2^{\lfloor (n-9)/4 \rfloor}$ on the size of read-once branching programs for $\mathrm{MULT}_{n-1,n}$. □

Similar exponential lower bounds for other arithmetic functions can be obtained by so-called read-once projections $\leq_{rop}$ ([8, 17]).

Definition 5. i) Squaring $\mathrm{SQU}_n{:}\{0,1\}^n \to \{0,1\}^{2n}$ computes the square of an n-bit integer, i.e., $\mathrm{SQU}_n(x) = z_{2n-1} \ldots z_0$, where $x = x_{n-1} \ldots x_0$ and $x^2 = z = z_{2n-1} \ldots z_0$. $\mathrm{SQU}_{i,n}$ computes the ith bit of SQU_n.

ii) Inversion $\mathrm{INV}_n{:}\{0,1\}^{n-1} \to \{0,1\}^n$ computes the n most significant bits of the multiplicative inverse of an n-bit number, i.e., $\mathrm{INV}_n(x) = 1z_{-1} \ldots z_{-n}$, where $x = 1x_{-2} \ldots x_{-n}$ and $x^{-1} = z = 1z_{-1} \ldots z_{-n}$. $\mathrm{INV}_{i,n}$ computes the ith bit of INV_n.

iii) Division $\mathrm{DIV}_n{:}\{0,1\}^{2n} \to \{0,1\}^n$ computes the n most significant bits of the quotient of two n-bit integers, i.e., $\mathrm{DIV}_n(x,y) = z_{n-1} \ldots z_0$ where $x = x_{n-1} \ldots x_0$ and $y = y_{n-1} \ldots y_0$ and $\lfloor x/y \rfloor = z = z_{n-1} \ldots z_0$. $\mathrm{DIV}_{i,n}$ computes the ith bit of DIV_n.

In order to prove optimal lower bounds on the depth of polynomial-size threshold circuits Wegener [24] has presented the following reductions

$$\mathrm{MULT} \leq_{rop} \mathrm{SQU} \leq_{rop} \mathrm{INV} \leq_{rop} \mathrm{DIV}.$$

The lower bound for $\mathrm{MULT}_{n-1,n}$ implies directly lower bounds for the appropriate bits of the other three functions.

COROLLARY 1. *The* BP1 *size of the arithmetic functions* $\mathrm{SQU}_{n,n}$, $\mathrm{INV}_{-n,n}$, *and* $\mathrm{DIV}_{0,n}$ *is bounded below by* $2^{\Omega(n)}$.

Conclusion

The effect of partial assignments of the input is hard to understand for integer multiplication but using methods which rely on universal hashing we have been able to analyse the subfunctions of $\mathrm{MULT}_{n-1,n}$. We have learned that even if we replace almost a quarter of the variables of each factor by constants each result of the product bits between the positions $n-1$ to $3/4n$ will still be possible.

Acknowledgement

Thanks to Martin Dietzfelbinger and Ingo Wegener for several valuable hints and fruitful discussions.

5. REFERENCES

[1] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proc. of 40th FOCS*, pages 60–70, 2000.

[2] M. Ajtai, L. Babai, P. Hajnal, J. Komlós, P. Pudlák, V. Rödl, E. Szemerédi, and G. Turán. Two lower bounds for branching programms. In *Proc. 18th STOC*, pages 30–38, 1986.

[3] N. Alon, M. Dietzfelbinger, P. Miltersen, E. Petrank, and G. Tardos. Linear hash functions. *Journal of the ACM*, 46:667–683, 1999.

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

[5] A. Andreev, J. Baskov, E. Clementi, and R. Rolim. Small pseudo-random sets yield hard functions: new tight explicit lower bounds for branching programs. In *Proc. of ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 179–189, 1999.

[6] P. Beame, M. Saks, X. Sun, and E. Vee. Super-linear time-space tradeoff lower bounds for randomized computation. In *Proc. 41st FOCS*, 2000.

[7] B. Bollig. Restricted nondeterministic read-once branching programs and an exponential lower bound for integer multiplication. In *Proc. 25th MFCS*, volume 1893 of *Lecture Notes of Computer Science*, pages 222–231, 2000.

[8] B. Bollig and I. Wegener. Read-once projections and formal circuit verification with binary decision diagrams. In *Proc. 23rd STACS*, volume 1046 of *Lecture Notes of Computer Science*, pages 491–502, 1996.

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

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

[11] M. Dietzfelbinger. Universal hashing and k-wise independent random variables via integer arithmetic without primes. In *Proc. 13th STACS*, volume 1046 of *Lecture Notes in Computer Science*, pages 569–580, 1996.

[12] M. Dietzfelbinger, T. Hagerup, J. Katajainen, and M. Penttonen. A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, 25:19–51, 1997.

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

[14] A. Hajnal, W. Maass, P. Pudlák, and G. Turán. Threshold circuits of bounded depth. In *Proc. 28th FOCS*, pages 99–110, 1987.

[15] K. Kriegel and S. Waack. Lower bounds on the complexity of real-time branching programs. In *Proc. FCT*, volume 278 of *Lecture Notes in Computer Science*, pages 263–267, 1987.

[16] S. Ponzio. A lower bound for integer multiplication with read-once branching programs. In *Proc. 27th STOC*, pages 130–139, 1995.

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

[18] P. Savický and S. Žák. A read-once lower bound and a $(1,+k)$-hierarchy for branching programs. *Theoretical Computer Science*, 238:347–362, 2000.

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

[20] 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 (ed. J. Cai)*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 183–193, 1993.

[21] S. Žák. An exponential lower bound for one-time-only branching programs. In *Proc. 9th MFCS*, volume 176 of *Lecture Notes in Computer Science*, pages 562–566, 1984.

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

[23] I. Wegener. On the complexity of branching programs and decision trees for clique functions. *Journal of the ACM*, 35:461–471, 1988.

[24] I. Wegener. Optimal lower bounds on the depth of polynomial-size threshold circuits for some arithmetic functions. *Information Processing Letters*, 46:85–87, 1993.

[25] P. Woelfel. Efficient strongly universal and optimally universal hashing. In *Proc. 24th MFCS*, volume 1672 of *Lecture Notes in Computer Science*, pages 262–272, 1999.

[26] P. Woelfel. *Klassen universeller Hashfunktionen mit ganzzahliger Arithmetik*. Diploma thesis, Univ. Dortmund, 2000.

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