Electronic Colloquium on Computational Complexity, Revision 01 of Report No. 073 (2001)

ECCC

Parity Graph-driven Read-Once Branching Programs and an Exponential Lower Bound for Integer Multiplication

Beate Bollig*[1], Stephan Waack[2], and Philipp Woelfel**[1]

[1] FB Informatik, LS2, Univ. Dortmund,
44221 Dortmund, Germany
bollig,woelfel@ls2.cs.uni-dortmund.de
[2] Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen
Lotzestr. 16-18, 37083 Göttingen, Germany
waack@math.uni-goettingen.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. Parity graph-driven read-once branching programs have been investigated intensively by Brosenne, Homeister, and Waack [8]. Here, 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 parity nonoblivious read-once branching program model for an explicitly defined Boolean function. In addition more insight into the structure of integer multiplication is yielded.

Keywords: Computational complexity, binary decision diagrams, branching programs, integer multiplication, lower bounds, parity nondeterminism.

* Supported in part by DFG We 1066/9.
** Supported in part by DFG We 1066/10.

ISSN 1433-8092

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. This graph represents a Boolean function $f_n : \{0,1\}^n \to \{0,1\}$ on X_n in the following way. In order to evaluate $f_n(a)$ for a given assignment $a \in \{0,1\}^n$ of the input variables, one follows a path starting at the source. At an internal node labeled by x_i, the path continues with the edge labeled by a_i (this is called a test of a variable). The output for a is the label of the sink which is finally reached.*

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 which 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 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 depth 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 nonuniform Turing machines and known to lie between the circuit size of f and its $\{\wedge, \vee, \neg\}$-formula size (see, e.g., [24]). Hence, one is interested in exponential lower bounds for more and more general types of BPs (for the latest breakthrough for semantic linear depth BPs see [1] 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* 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.*

Borodin, Razborov, and Smolensky [7] have proved one of the first exponential lower bounds for BPks. For oblivious branching programs of restricted depth exponential lower bounds have been proven, e.g., by Alon and Maass [3]. Nondeterminism is one of the most powerful concepts in computer science. In analogy to the definition for Turing machines, different modes of acceptance can be studied for branching programs (for more details we refer to [18]). There are several definitions of nondeterministic branching programs which are only slightly different. Here we use the following definition.

Definition 3. *A* nondeterministic branching program *is a generalized branching program where the number of edges leaving an internal node is not restricted. The function value of the represented function for a given assignment a of the input variables is 1, if and only if 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 if the number of its accepting paths is odd. The size of a nondeterministic branching program is the number of its nodes.*

Definitions of nondeterministic variants of restricted BPs are derived in a straightforward way. The results of Borodin, Razborov, and Smolensky [7] for BPks hold (and have been stated by the authors) also for nondeterministic-BPks.

Besides this complexity theoretical viewpoint people have used branching programs in applications. 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.

Definition 4. *Let $X_n = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. A variable ordering π on X_n is a permutation of $\{1, \ldots, n\}$ leading to the ordered list $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the variables. An* OBDD *is a read-once branching program where a variable ordering π is fixed. On each computation path the variables are tested according according to π, i.e., if an edge leads from an x_i-node to an x_j-node, the condition $\pi^{-1}(i) < \pi^{-1}(j)$ has to be fulfilled.*

Unfortunately, several important and also quite simple functions have exponential OBDD size. Therefore, more general representations with good algorithmic behavior are necessary. Gergov and Meinel [15] and Sieling and Wegener [22] have shown independently how read-once branching programs can be used for verification. In order to obtain efficient algorithms for many operations they have generalized the concept of variable orderings to graph orderings.

Definition 5. *A* graph ordering *is a branching program with a single sink. On each path from the source to the sink there is for each variable x_i exactly one node labeled by x_i.*

A graph-driven BP1 *with respect to a graph ordering G_0, G_0-*BP1 *for short, is a* BP1 *with the following additional property. For an arbitrary input $a \in \{0,1\}^n$, let $\mathcal{L}(a)$ be the list of labels at the nodes on the computation path for a in the* BP1 *and similarly let $\mathcal{L}_0(a)$ be the list of labels on the computation path for a in G_0. We require that $\mathcal{L}(a)$ is a subsequence of $\mathcal{L}_0(a)$.*

It is easy to see that an arbitrary read-once branching program is ordered with respect to a suitably chosen graph ordering.

Brosenne, Homeister, and Waack [8] have investigated parity graph-driven read-once branching programs which are a proper generalization of parity OBDDs and BP1s (for a formal definition see Definition 8 in Section 2). A similar nondeterministic model has been introduced by Bollig [5]. Obviously not each parity BP1 is a

graph-driven BP1. It is still open if there exist Boolean functions which need parity graph-driven BP1s of exponential size but can be represented by (general) parity BP1s of polynomial size.

For many restricted (nondeterministic) variants of branching programs exponential lower bounds are known. A survey of known lower bounds can be found in [20]. Moreover, Thathachar [23] was even able to prove an exponential gap between the size of nondeterministic BPks and deterministic BP$(k+1)$s for an explicitly defined Boolean function demonstrating that the lower bound techniques for this models are highly developed. But the problem of proving superpolynomial lower bounds for parity read-once branching programs is still open. Our results could be one step further towards an exponential lower bound for parity BP1s. Krause [17] has proved the first exponential lower bounds for oblivious parity branching programs with bounded depth. Later Savický and Sieling [21] 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 exponential lower bound of order $2^{\Omega(n^{1/2})}$ on the size of parity graph-driven BP1s representing the characteristic function of linear codes.

The proof of exponential lower bounds on the size of BDD models for *natural* functions is often a challenge.

Definition 6. Integer multiplication *is the Boolean function* $\mathrm{MULT}_n{:}\{0,1\}^{2n} \to \{0,1\}^{2n}$ *that maps two n-bit integers to their product. That is,* $\mathrm{MULT}_n(x,y) = z_{2n-1}\dots z_0$ *where* $x = x_{n-1}\dots x_0$ *and* $y = y_{n-1}\dots y_0$ *and* $x \cdot y = z = z_{2n-1}\dots z_0$. $\mathrm{MULT}_{i,n}$ *denotes the Boolean function defined as 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 $TC^{0,2}$ [16]. 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 [3] and using the rank method of communication complexity Gergov [14] has extended the lower bound to arbitrary nondeterministic linear-depth oblivious BPs. Recently Woelfel [27] has improved Bryant's lower bound up to $\Omega(2^{n/2})$. The first exponential lower bound on the size of deterministic read-once branching programs has been proven by Ponzio [19]. His lower bound is of order $2^{\Omega(n^{1/2})}$ and has been improved by Bollig and Woelfel [6] to the first strongly exponential lower bound of size $\Omega(2^{n/4})$ for $\mathrm{MULT}_{n-1,n}$. Bollig [5] has presented the first (not strongly) exponential lower bound on the size of $\mathrm{MULT}_{n-1,n}$ for so-called nondeterministic tree-driven read-once branching programs. Her result also holds (and has been stated by the author) for parity tree-driven read-once branching programs. But this model is very restricted. 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. Here, we present an exponential lower bound on the size of restricted parity graph-driven BP1s for $\mathrm{MULT}_{n-1,n}$. 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.

The rest of the paper is organized as follows. In Section 2 we carefully define parity graph-driven BP1s. Similar to the deterministic case [22] two different models of parity graph-driven BP1s are distinguished. Brosenne, Homeister, and Waack [8] have applied methods from linear algebra to present an exact characterization of the number of nodes in a well-structured parity graph-driven BP1 for a Boolean function f. (For a formal definition of the model see Section 2.) We restate some of their results and investigate the relationship between 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. Afterwards we describe a new lower bound criterion for the size of well-structured parity graph-driven BP1s representing a Boolean function f. In Section 3 we consider the function $\mathrm{MULT}_{n-1,n}$ in more detail. Finally, in Section 4 we apply the lower bound method presented in Section 2 to $\mathrm{MULT}_{n-1,n}$ and prove the first strongly exponential lower bound for well-structured parity graph-driven BP1s.

2 Algebraic Characterization and Lower Bounds for Parity Graph-driven BP1s

Sieling and Wegener [22] have introduced graph-driven BP1s as data structure for Boolean functions and have proved that the usual operations on OBDDs can be performed efficiently also for graph-driven BP1s. They have distinguished two different models, the second one, well-structured graph-driven BP1s, is a restricted variant of graph-driven BP1s.

Definition 7. *A graph-driven* BP1 *with respect to a graph ordering G_0 is called a* well-structured graph-driven BP1 *if there exists a representation function $\alpha : V \to V_0$ 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.*

The reason for the two models is a time-space trade-off between graph-driven and well-structured graph-driven BP1s. A special property of the last one leads to the design of simpler and faster algorithms. The difference is the following one. If we reach the node v of a well-structured G_0-BP1 for some input a, then it follows that the node $\alpha(v)$ of the graph ordering G_0 is also reached for this input. In the general graph-driven model it is possible that the node v with label x_i is reached for the inputs a and b while the nodes with label x_i on the computation paths for the inputs a and b are different.

Brosenne, Homeister, and Waack [8] have realized how this property can be used for well-structured parity graph-driven BP1s to determine the number of nodes which is necessary to represent a Boolean function f. At each internal node v of a parity read-once branching program a function $f_v : \{0,1\}^n \to \{0,1\}$ is represented which depends syntactically on all variables of the function f. This function is defined as follows. The function f_v computes 1 on input a iff the number of paths activated by a which lead from v to the 1-sink is odd. More precisely, if an x_i-node v has 0-edges leading to $u_1, \ldots, u_k$ and 1-edges leading to $w_1, \ldots, w_l$, we define

$$f_v = \left[\overline{x_i} \wedge (f_{u_1} \oplus \cdots \oplus f_{u_k})\right] \oplus \left[x_i \wedge (f_{w_1} \oplus \cdots \oplus f_{w_l})\right].$$

Unlike the deterministic case the source of a parity BP1 can also have incoming edges. It is obvious that edges to the 0-sink can be eliminated and the constant 0 is represented by an empty branching program. Double edges can always be eliminated without changing the represented function. More precisely, if r edges with the same label lead from v to w, they could be replaced by (r mod 2) edges of the same kind. Parity branching programs may contain internal nodes without outgoing edges. These nodes represent the constant 0 and are always eliminated together with their incoming edges.

Definition 8. *A* parity graph-driven BP1 G with respect to a graph ordering G_0, *parity* G_0-BP1 *for short, is a parity* BP1 *with the following additional property. For an arbitrary input* $a \in \{0,1\}^n$, *let* $\mathcal{L}(a,p)$ *be the list of labels at the nodes on a computation path* p *for* a *in* G *and similarly let* $\mathcal{L}_0(a)$ *be the list of labels on the computation path for* a *in* G_0. *We require that* $\mathcal{L}(a,p)$ *is a subsequence of* $\mathcal{L}_0(a)$ *for each computation path* p *for* a. *The size of a parity* G_0*-driven BP1* G *is the number of nodes in* G *and is denoted by* $|G|$.

Similar to the deterministic case well-structured parity G_0-BP1s are defined. Brosenne, Homeister, and Waack [8] have shown that the maximal quotient of the minimal size well-structured parity G_0-BP1 and the minimal size parity G_0-BP1 representing a function f is bounded by $O(|G_0|)$, the size of the fixed graph ordering (similar to the deterministic case). But unlike the deterministic case it is not clear how the size of a parity graph-driven BP1 G and the minimal size of a graph ordering G_0 such that G is G_0-driven are related. The situation is different for well-structured parity graph-driven BP1s. Here we will see that for each well-structured parity graph-driven BP1 G there exists a graph ordering G_0 such that G is G_0-driven and $|G_0| \leq 2n|G|$. This property will be very helpful in order to prove exponential lower bounds on the size of well-structured parity graph-driven BP1s.

The following lemma is a slight generalization of a result from [22].

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* $\alpha(w)$.

Now we are able to prove the following proposition.

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 [25]) 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. □

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.

Now 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.

In the following, 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. It is known that on all paths from the source to v the same set of variables is tested. Without loss of generality let $x_1, \ldots, x_{i-1}$ be the previously tested variables. 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 $x_1, \ldots, x_{i-1}$, where a function g does not essentially depend on a variable x_j iff $g_{|x_j=0} = g_{|x_j=1}$.

Now let $\mathcal{P}_v$ be the set of all nodes which lie on a path leaving v in G_0. (Note, that v itself lies on all paths from v to the sink.) Then we define $\mathbb{B}_{f,v}^{G_0}$ as the Boolean vector space spanned by all functions in

$$\bigcup_{w \in \mathcal{P}_v} \mathcal{F}_w.$$

Lemma 2 (Brosenne, Homeister, and Waack [8]). *Let G be a well-structured parity G_0-driven* BP1 *representing f, v a node in G_0, and $J(v)$ the first successor of v in G_0, $J(v) \neq v$, which lies on all paths leaving v. Then*

$$|\mathcal{N}_v(G)| = dim_{\mathbb{Z}_2} \mathbb{B}_{f,v}^{G_0} - dim_{\mathbb{Z}_2} \mathbb{B}_{f,J(v)}^{G_0}.$$

The following observation will be helpful. W.l.o.g. let $x_1, \ldots, x_{i-1}$ be the set of variables tested on any path from the source of G_0 to v and let v be labeled by the variable x_i. Then it follows that the vector space $\mathbb{B}_{f,J(v)}^{G_0}$ is a subspace of the vector space spanned by all Boolean functions not essentially depending on $x_1, \ldots, x_i$.

Now our idea to prove stronger lower bounds is the following one. 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 $V_1 \cap V_2 = \{o\}$, i.e., $\dim V_1 + \dim V_2 = \dim(V_1 + V_2)$. This means that no vector in $V_1 \setminus \{o\}$ can be represented by a linear combination in V_2 and vice versa.

Lemma 3. *Let $A'(v)$ be a subset of $A(v)$ such that the subfunctions $f_{|x_1=a_1,\ldots,x_{i-1}=a_{i-1}}$, where $(a_1,\ldots,a_{i-1}) \in A'(v)$, 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*

$$|\mathcal{N}_v(G)| = dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,v} - dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,J(v)} \geq |A'(v)|.$$

Proof. Since all functions in $\mathbb{B}^{G_0}_{f,J(v)} \subseteq \mathbb{B}^{G_0}_{f,v}$ are obviously not essentially depending on x_i, $\mathbb{B}^{G_0}_{f,A'}$ is also linearly independent modulo $\mathbb{B}^{G_0}_{f,J(v)}$. Hence,

$$\dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,A'} + \dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,J(v)} = \dim_{\mathbb{Z}_2}\left(\mathbb{B}^{G_0}_{f,A'} + \mathbb{B}^{G_0}_{f,J(v)}\right) \leq \dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,v}, \qquad (1)$$

where the inequality follows from the fact that $\mathbb{B}^{G_0}_{f,A'} + \mathbb{B}^{G_0}_{f,J(v)} \subseteq \mathbb{B}^{G_0}_{f,v}$. Since all the subfunctions defined by $A'(v)$ as above are linearly independent, $\dim_{\mathbb{Z}_2}\mathbb{B}^{G_0}_{f,A'}$ equals $|A'(v)|$ and the desired result follows from inequality (1). □

In the last section we will apply Lemma 3 to prove the first strongly exponential lower bound for well-structured parity graph-driven BP1s.

3 Integer Multiplication

We start our investigations with two technical lemmas which provide important properties of the function $\mathrm{MULT}_{n-1,n}$.

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 $[x]_r^l = z$ if $(x_l,\ldots,x_r)$ is the binary representation of the integer $z \in \{0,\ldots,2^{l-r+1}-1\}$. Sometimes, we identify $[x]_r^l$ with z if the meaning is clear from the context.

3.1 The Covering Lemma

Using universal hashing Bollig and Woelfel [6, proof of Lemma 5] have shown the following.

Lemma 4. *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*

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

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 [6] have proved this statement only implicitly in a non-parameterized form. Therefore, we provide a complete proof in the appendix.

3.2 The Distance Lemma

We now state another important lemma about integer multiplication, which is a generalization of Lemma 6 from [6].

Lemma 5. *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

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

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 [6, 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 $|Y| - t \cdot 2^{n-k+1}$. □

3.3 The Matrix Game

First, we motivate our investigations. (In Section 4 we will make the following ideas more precise.) 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 $\mathrm{MULT}_{n-1,n}$ 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 3 that the size of parity G_0-BP1s representing $\mathrm{MULT}_{n-1,n}$ is large.

In the following, we investigate integer multiplication for 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 be the set of variables $x_1, \ldots, x_{n-2}$ 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 $\mathrm{MULT}_{n-1,n}(x_{r,c}, y_c)$. 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 rows $r^1, \ldots, r^l$ there exists a column c such that

$$\bigoplus_{j=1}^{l} M_{r^j,c} \neq \bigoplus_{j=1}^{l} M_{r^j,c'}, \tag{2}$$

which means that the number of rows r^j, $1 \leq j \leq l$, where $M_{r^j,c} \neq M_{r^j,c'}$, is odd.

Before we show (2) we illustrate how this property can be used to prove lower bounds using Lemma 3. 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 $\mathrm{MULT}_{|\alpha}$. In this setting, (2) 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 3 this allows the conclusion that

$$|\mathcal{N}_v(G)| \geq |A'(v)|,$$

where $A'(v) \subseteq A^*(v)$.

Let $x_{r,c}$ be the number $x \in \mathbb{Z}^*_{2^{n-1}}$ defined by the choice of a row r and a column c and y_c the number $y \in \mathbb{Z}^*_{2^{n-1}}$ 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

$$M_{r,c} = [(x_r^{row} + x_c^{col}) \cdot y_c]_{n-1}.$$

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 6.

Lemma 6. *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) for all $x \in X$ *and all* $2 \leq j \leq l$: $x^1 \neq x^j$ *and all* $1 \leq j \leq l$: $x^1 \neq x^j + d$,
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 6 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), $z \in Z$,

$$4 \cdot 2^{n-k-1} \leq ((z - x^1)y) \bmod 2^{n-1} \leq 2^{n-1} - 4 \cdot 2^{n-k-1}. \tag{3}$$

According to Lemma 5

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

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 4. 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

$$[(x^1 + x)y]_{n-k}^{n-1} = 2^{k-1} - 1 \quad \text{and} \quad [(x^1 + x')y]_{n-k}^{n-1} = 2^{k-1}. \tag{4}$$

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 (4). 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 (4) it follows that

$$2^{n-1} - 2^{n-k} \;\leq\; ((x^1 + x)y) \bmod 2^n \;<\; 2^{n-1} \quad \text{and} \tag{5}$$
$$2^{n-1} \;\leq\; ((x^1 + x')y) \bmod 2^n \;<\; 2^{n-1} + 2^{n-k}. \tag{6}$$

From this it follows that

$$1 \;\leq\; ((x' - x)y) \bmod 2^n \;<\; 2 \cdot 2^{n-k}. \tag{7}$$

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 (7)

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

Together with inequality (5) 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 (3) and we are done. □

Altogether, we have proved that the vector space spanned by all subfunctions of $\text{MULT}_{n-1,n}$ 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 $\text{MULT}_{n-1,n}$ 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

In this section, we combine the lower bound technique for well-structured parity graph-driven BP1s presented in Section 2 with Lemma 6 in order to 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* $\mathrm{MULT}_{n-1,n}$ *is bounded below by* $2^{(n-46)/12}/n$.

Proof. Let G be a well-structured parity graph-driven BP1 representing $\mathrm{MULT}_{n-1,n}$ 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 $\mathrm{MULT}_{n-1,n}$ 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, 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 6 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

$$|\mathcal{N}_v(G)| = \dim_{\mathbb{Z}_2} \mathbb{B}^{G_0}_{f,v} - \dim_{\mathbb{Z}_2} \mathbb{B}^{G_0}_{f,J(v)} \geq |A'(v)| \geq 2^{1/2\lfloor(n-17)/6\rfloor}.$$

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 read-once branching programs representing $\mathrm{MULT}_{n-1,n}$. □

Acknowledgement

Thanks to Stefan Droste, Detlef Sieling, and Ingo Wegener for fruitful discussions.

References

1. Ajtai, M. (1999). A non-linear time lower bound for Boolean branching programs. Proc. of 40th FOCS, 60–70.
2. Alon, N., Dietzfelbinger, M., Miltersen, P.B., Petrank, E., and Tardos, G. (1999). Linear hash functions. Journal of the ACM 46, 667–683.
3. Alon, N. and Maass, W. (1988). Meanders and their applications in lower bound arguments. Journal of Computer and System Sciences 37, 118–129.
4. Beame, P., Saks, M., Sun, X., and Vee, E. (2000). Super-linear time-space tradeoff lower bounds for randomized computation. Proc. of 41st FOCS, 169–179, and ECCC Report TR 00-025.
5. Bollig, B. (2001). Restricted nondeterministic read-once branching programs and an exponential lower bound for integer multiplication. RAIRO Theoretical Informatics and Applications 35, 149–162.
6. Bollig, B. and Woelfel, P. (2001). A read-once branching program lower bound of $\Omega(2^{n/4})$ for integer multiplication using universal hashing. Proc. of 33rd STOC, 419–424.
7. Borodin, A., Razborov, A., and Smolensky, R. (1993). On lower bounds for read-k-times branching programs. Comput. Complexity 3, 1–18.
8. Brosenne, H., Homeister, M., and Waack, St. (2001). Graph-driven free parity BDDs: algorithms and lower bounds. Proc. of 26th MFCS, LNCS 2136, 212–223.
9. Bryant, R.E. (1986). Graph-based algorithms for Boolean manipulation. IEEE Trans. on Computers 35, 677–691.
10. Bryant, R.E. (1991). On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. IEEE Trans. on Computers 40, 205–213.
11. Carter, J.L. and Wegman, M.N. (1979). Universal classes of hash functions. Journal of Computing and System Science 18, 143–154.
12. Dietzfelbinger, M. (1996). Universal hashing and k-wise independent random variables via integer arithmetic without primes. Proc. 13th STACS, Lecture Notes in Computer Science 1046, 569–580.
13. Dietzfelbinger, M., Hagerup, T., Katajainen, J., and Penttonen, M. (1997). A reliable randomized algorithm for the closest-pair problem. Journal of Algorithms, Vol. 25, 19–51.
14. Gergov, J. (1994). Time-space trade-offs for integer multiplication on various types of input oblivious sequential machines. Information Processing Letters 51, 265–269.
15. Gergov, J. and Meinel, C. (1994). Efficient Boolean manipulation with OBDDs can be extended to FBDDs. IEEE Trans. on Computers 43, 1197–1209.
16. Hajnal, A. and Maass, W. and Pudlák, P. and Turán, G. (1987). Threshold circuits of bounded depth. Proc. 28th FOCS, 99–110.
17. Krause, M. (1992). Separating $\oplus$L, from NL, co-NL and AL (=P) for oblivious Turing machines of linear access time. RAIRO Theoretical Informatics and Applications 26, 507-522.
18. Meinel, C. (1990). Polynomial size Ω-branching programs and their computational power. Information and Computation 85, 163–182.
19. Ponzio, S. (1998). A lower bound for integer multiplication with read-once branching programs. SIAM Journal on Computing 28, 798–815. (A preliminary version has been appeared in Proc. of STOC 1995.)
20. Razborov, A. A. (1991). Lower bounds for deterministic and nondeterministic branching programs. Proc. of Fundamentals of Computation Theory (FCT), LNCS 529, 47–60.
21. Savický, P. and Sieling, D. (2000). A hierarchy result for read-once branching programs with restricted parity nondeterminism. Proc. of 25th MFCS, LNCS 1893, 650–659.
22. Sieling, D. and Wegener, I. (1995). Graph driven BDDs - a new data structure for Boolean functions. Theoretical Computer Science 141, 283–310.
23. Thathachar, J. (1998). On separating the read-k-times branching program hierarchy. Proc. of 30th STOC, 653–662.
24. Wegener, I. (1987). *The Complexity of Boolean Functions.* Wiley-Teubner.
25. Wegener, I. (2000). *Branching Programs and Binary Decision Diagrams - Theory and Applications.* SIAM Monographs on Discrete Mathematics and Applications.
26. Woelfel, P. (1999). Efficient strongly universal and optimally universal hashing. Proc. 24th MFCS, Lecture Notes in Computer Science 1672, 262–272.
27. Woelfel, P. (2001). New bounds on the OBDD-size of integer multiplication via universal hashing. Proc. of 18th STACS, LNCS 2010, 563–574.

A Appendix: Proof of the Covering Lemma (Lemma 4)

The concept of universal hashing introduced by Carter and Wegman in 1979 [11] 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 9. *A family $\mathcal{H}$ of hash functions $U \to R$ is universal if for any $x, x' \in U$ with $x \neq x'$*

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

In order to prove Lemma 4, we show a similar covering lemma for universal hash families. Consider a universal hash family $\mathcal{H}$ and a subset V of the universe. The following lemma states that there is a large fraction of hash functions h in $\mathcal{H}$ under which the function values of the elements from V cover the whole range R, i.e.

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

Using this result and the known fact that the mappings $x \mapsto [ax+b]_{n-k}^{n-1}$ (for a and b being elements from appropriate sets) form a universal hash family (see Lemma 8), we can then easily derive Lemma 2.

Note that a simpler version of the following lemma has already been known (see, e.g., [2]).

Lemma 7. *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 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 $\delta_{\mathcal{H}\setminus F}(V,V) \geq |\mathcal{H}\setminus F| \cdot v\left(\frac{v}{r} - 1\right)$. 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 n-bit universe $U := \mathbb{Z}_{2^n}$ to the k-bit range $R_k := \{0, \ldots, 2^k - 1\}$, $1 \leq k \leq n$. For $a, b \in U$ let

$$h^k_{a,b} : U \to R_k, \quad x \mapsto \big((ax + b) \bmod 2^n\big) \operatorname{div} 2^{n-k},$$

where div is the integer division, i.e., $x \operatorname{div} y = \lfloor x/y \rfloor$. Note that in our notation we can write $h_{a,b}(x)$ as $[ax + b]^{n-1}_{n-k}$.

Lemma 8 (Woelfel [26]). *Let $B = \{0, \ldots, 2^{n-k} - 1\} \subseteq U$. The family of hash functions $\mathcal{H}_k := \left\{h^k_{a,b} \,\middle|\, a \in \mathbb{Z}^*_{2^n},\ b \in B\right\}$ is universal.*

Similar hash classes have been investigated by Dietzfelbinger [12], Dietzfelbinger, Hagerup, Katajainen, and Penttonen [13], and Woelfel [26].

We can now use Lemma 7 together with the hash family $\mathcal{H}_k$ to prove the Covering Lemma.

Proof of Lemma 4. Let $1 \leq k < n/2$ and consider the hash family $\mathcal{H}_{k+1}$ as defined in Lemma 8. Let $r := |R_{k+1}|$ and $F := \left\{h_{y,b}^{k+1} \,\middle|\, y \in Y,\, b \in B\right\} \subseteq \mathcal{H}_{k+1}$. Then by definition

$$\frac{|F| \cdot |X|}{|B|} \;=\; |Y| \cdot |X| \;\geq\; 2^{n+2k+1}.$$

Furthermore,

$$\frac{|\mathcal{H}_{k+1}| \cdot (r-1)^2}{|B|} \;<\; 2^{n-1} \cdot 2^{2k+2} = 2^{n+2k+1}.$$

Therefore, it follows that $|F|/|\mathcal{H}_{k+1}| > (r-1)^2/|X|$. Using Lemma 7 we can conclude that there exists a hash function $h_{y,b}^{k+1} \in F$ such that $\left\{h_{y,b}^{k+1}(x) \,\middle|\, x \in X\right\} = R_{k+1}$. Hence, there exist an $y \in Y$ and an element $b \in B$ such that

$$\left\{[xy+b]_{n-k-1}^{n-1} \,\middle|\, x \in X\right\} \;=\; \left\{0, \ldots, 2^{k+1}-1\right\}. \tag{8}$$

Let these y and b be fixed. It suffices to show that for any $z \in \left\{0, \ldots, 2^k - 1\right\}$ we can find an $x \in X$ such that $[xy]_{n-k}^{n-1} = z$.

Let the binary representation of z be $z_{k-1} \ldots z_0$ and $x \in X$ such that

$$[xy+b]_{n-k-1}^{n-1} \;=\; z_{k-1} \ldots z_0 1.$$

Equation (8) ensures the existence of such an element x. Since $b < 2^{n-k-1}$, it follows that $[xy]_{n-k-1}^{n-1} = z_{k-1} \ldots z_0 q$, where $q \in \{0,1\}$, and we get $[xy]_{n-k}^{n-1} = z$. □

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'