Electronic Colloquium on Computational Complexity, Report No. 2 (1998)

ECCC

On Separating the Read-k-Times Branching Program Hierarchy

Jayram S. Thathachar

Abstract

We obtain an exponential separation between consecutive levels in the hierarchy of classes of functions computable by polynomial-size syntactic read-k-times branching programs, for *all* $k > 0$, as conjectured by various authors [Weg87, SS93, Pon95]. For every k, we exhibit two explicit functions that can be computed by linear-sized read-$(k+1)$-times branching programs but require size $\exp\left\{\Omega\left(n^{1/k+1}2^{-2k}k^{-4}\right)\right\}$ to be computed by any read-k-times branching program. The result actually gives the strongest possible separation — the exponential lower bound applies to both non-deterministic read-k-times branching programs and randomized read-k-times branching programs with 2-sided error ε, for some $\varepsilon > 0$. The only previously known results are the separation between $k = 1$ and $k = 2$ [BRS93] and a separation of non-deterministic read-k from deterministic read-$(k \ln k / \ln 2 + C)$, where C is some appropriate constant, for each k [Oko97]. A simple corollary of our results is that randomization is not more powerful than non-determinism for read-k-times branching programs. A combinatorial result that we prove along the way is a "hash-mixing lemma" (see [MNT93]) for families of hash functions that are *almost* universal, which may be of independent interest.

ISSN 1433-8092

1. Introduction

Branching programs and their many variants have long been a popular model for studying the complexity of functions (see, for example, the survey paper of Razborov [Raz91]). A *(boolean) branching program* is a directed acyclic graph with a source node and sink nodes labeled either "accept" or "reject." Each non-sink node has out-degree two and the two outgoing edges are labeled $x_i = 0$ and $x_i = 1$ for some input variable x_i. For any boolean assignment to $x_1, x_2, \ldots, x_n$, there is a unique computation path from the source node to one of the sink nodes in which all the edge labelings are consistent with the assignment. In this way, any branching program computes a boolean function. The two important complexity measures for a branching program are its *size*, which is the number of nodes, and *length*, which is the maximum length of a computation path from the source to a sink.

Branching programs are an important abstraction of many computing models and are closely related to other well studied models of computation (see, for e.g. [Weg87]). In a natural way, the size and length measure the space and time, respectively, used in a computation. A superpolynomial size lower bound for a function f would imply that f is not computable in non-uniform log-space. However, the best size lower bound for an explicit function in NP is $\Omega(\frac{n^2}{\log^2 n})$ due to Nečiporuk [Neč66] (see, also [BS90]). Branching programs are also useful for studying time-space tradeoffs (see [Bor93] for a survey of results and open problems). It is a major open problem to exhibit a function that requires super-polynomial size branching programs of linear length. Such a result would imply that there are functions that cannot be simultaneously computed in log-space and linear time. Since linear length implies that on average each variable is tested only a constant number of times on any path, researchers have looked at restricted variants of branching programs that capture this property in the hope of proving better bounds.

Read-k-times branching programs (see [Weg87]) have the restriction that along each path from source to sink, each variable may be tested at most k times. The case $k = 1$, i.e. read-once branching programs, is well understood and a variety of lower bound and separation results have been shown for it (see [Raz91] for an overview, [SS93] for an overview and summary of proof techniques and [Sau97b] for a discussion on the separation results). Wegener [Weg87] conjectured that the hierarchy of classes of functions computable by polynomial-size read-k-times branching programs, for $k \geq 1$, is proper. Borodin, Razborov and Smolensky [BRS93] observed (and clarified) that there are actually two potentially different classes of read-k-times branching programs that can be considered — the *semantic* type where the restriction on multiple reads applies only to computational paths and *syntactic* type where this restriction applies to the non-computational paths as well (for $k = 1$, the two classes coincide). For $k \geq 2$, no non-trivial bounds have been proved for semantic read-k-times branching programs. On the other hand, for the syntactic model, Borodin *et al.* [BRS93] were able to show exponential size bounds for some explicit function, for each $k \leq c \log n$, where c is some appropriate constant. Here again, a natural problem is to separate the syntactic read-k-times branching program hierarchy for $k \geq 2$. Various authors [Pon95, SS93, Weg87] have worked on this problem and conjectured candidate predicates for separating the hierarchy for $k \geq 2$; for example, Ponzio [Pon95] suggests looking at the generalization of the permutation matrix predicate in the k-dimensional hypercube. Okolnishnikova [Oko97] has made the only progress in this direction by showing that there are functions that can be computed by syntactic read-$(k \ln k / \ln 2 + C)$-times branching programs, for some constant C, but require exponential size non-deterministic syntactic read-k-times branching programs. However, the strict separation problem has been open even for $k = 2$, that is, is there a function that can be computed by a polynomial size read-thrice branching program but requires read-twice branching programs of superpolynomial size?

We resolve this question completely by obtaining an exponential separation between consecutive levels

in the hierarchy of classes of functions computable by polynomial size syntactic read-k-times branching programs, for all $k > 0$. For any odd prime q, we consider two predicates — the hyperplanar sum-of-products predicate (HSP_q^k) and the conjunction of hyperplanar sum-of-products predicate (CHSP_q^k) both of which are defined on the k-dimensional hypercube $[1,n]^k$ in terms of the $(k-1)$-dimensional hyperplanes orthogonal to the k dimensions. (There are a total of kn such hyperplanes. For example, when $k = 2$, these correspond to the rows and columns of a $n \times n$ matrix.) HSP_q^k is defined to be true if the number of hyperplanes that have odd parity is equivalent modulo q to the number of hyperplanes that have even parity. CHSP_q^k is true if and only if amongst the hyperplanes orthogonal to each dimension, the number of hyperplanes that have odd parity is equivalent modulo q to the number of hyperplanes that have even parity. (See Section 2 for a formal definition.) We prove:

Theorem 1: Let $k \geq 1$ and q be any odd prime. Then, HSP_q^{k+1} and CHSP_q^{k+1} can be computed by linear-sized read-$(k+1)$-times branching program whereas any non-deterministic read-k-times branching program for CHSP_q^{k+1} requires size at least $\exp\left\{\Omega\left(n^{1/k+1}2^{-2k}k^{-3}\right)\right\}$ and for HSP_q^{k+1} requires size at least $\exp\left\{\Omega\left(n^{1/k+1}2^{-2k}k^{-4}\right)\right\}$.

We also show that a similar size bound holds for randomized read-k-times branching programs with 2-sided error ε, for some $\varepsilon > 0$ that is constant for HSP_q^{k+1} but depends on k for CHSP_q^{k+1}. An easy corollary of our result is that randomization is not more powerful than non-determinism for read-k-times branching programs. Our results and proof techniques are inspired by the work of Borodin *et al.* [BRS93], Ponzio [Pon97] and Sauerhoff [Sau97b].

It is evident that lower bounds, hierarchy and separation results contribute to our understanding of branching program based complexity classes. Okolnishnikova [Oko93] has given exponential lower bounds on the size of (deterministic) read-k-times branching programs computing some function, for $k \leq c \log n / \log\log n$, where c is some appropriate constant. For randomized read-k-times branching programs, Sauerhoff [Sau97a] has exhibited exponential size bounds for the same function considered by Borodin *et al.* [BRS93]. Some authors have considered further restrictions of read-k-times branching programs. For example, k-OBDDs [BSSW93] are branching programs that can be partitioned into k layers such that each layer uses the same order to access the variables. Although separation results are known for such models, they are even weaker than oblivious branching programs. Thus, the arguments do not apply to general syntactic read-k-times branching programs. Another important motivation for studying read-restricted variants is that some of them are used in practice as data structures for boolean functions. For example, ordered read-once branching programs [Bry86], also known as OBDDs, have been applied in computer-aided design as tools for hardware verification and symbolic model checking [BCL$^+$94]. Unordered read-once branching programs have also shown some promise in this regard [GM94, SW95].

Some of the important features of our techniques are (i) a combinatorial argument that reduces the lower bound problem to analyzing the structure of *pseudo-rectangles* (Section 3), (ii) interesting properties about the structure of the two hyperplanar predicates and (iii) an analogue of the "hash-mixing lemma" of Mansour, Nisan and Tiwary [MNT93] for *almost* universal families of hash functions (Section 4.2), which may be of independent interest.

For the rest of the paper, we deal only with the syntactic variant of read-k-times branching programs. In Section 2, we define HSP_q^k and CHSP_q^k for any $k > 1$ and odd prime q and motivate the lower bound problem by showing that they belong to a natural class of predicates that can be computed efficiently by read-k-times branching programs but are possibly hard for read-$(k-1)$-times branching programs.

Section 3 contains the combinatorial argument that reduces the lower bound problem on read-k-times branching programs to analyzing the structure of special types of functions called *pseudo-rectangles*. Applying this technique, in Section 4 we show exponential size lower bounds for non-deterministic and randomized read-k-times branching programs. We conclude with a discussion of our results.

2. The Hyperplanar Predicates

In order to motivate our candidate predicates HSP_q^k and CHSP_q^k, where $k > 1$ and q is an odd prime, we show that they belong to a natural class of predicates $N = n^k$ variables that contains many other potential candidates for separating the read-k-times hierarchy. Each such function is defined on the k-dimensional hypercube $[1, n]^k$ of side n. Let us call the n hyperplanes perpendicular to the d^{th} axis, $d \in [1, k]$ as *d-planes*. In other words, the i^{th} d-plane, for $i \in [1, n]$, is the set $\{v \in [1, n]^k \ : \ v_d = i\}$. Let X_i^d denote the set of variables corresponding to the i^{th} d-plane; observe that $\left|X_i^d\right| = n^{k-1}$. The predicates in this class are defined as follows: Suppose g and h are predicates which can be computed by read-once branching programs of polynomial size. Think of g and h as gates with unbounded fan-in. Consider the predicate Q defined by a depth-2 circuit consisting of an g-gate at the top connected to kn h-gates. For $d \in [1, k]$ and $i \in [1, n]$, the input to the $[k(d-1)+i]^{th}$ h-gate is X_i^d (See Figure 1). In this circuit, the function computed at the g-gate can be implemented by a read-k-times branching program of polynomial size because each variable is input to only k h-gates (Take any read-once branching program computing g and replace each node and its two outgoing edges with an appropriate branching program computing h.) For example, the generalization of the permutation predicate to k dimensions,

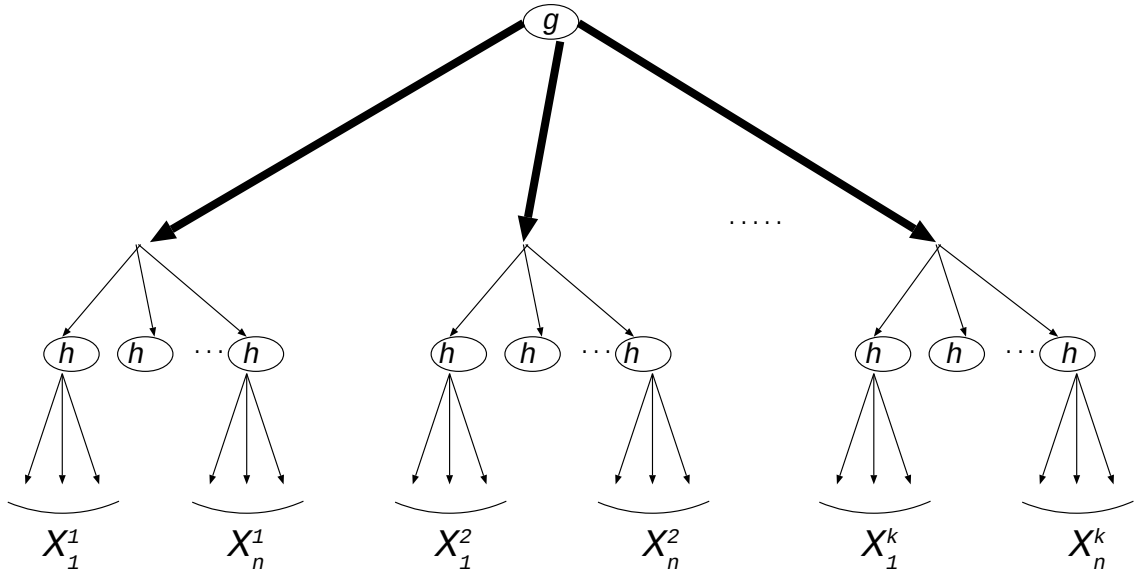

Figure 1: *The predicate Q described as a depth-2 circuit*

conjectured by Ponzio [Pon95] to separate the read-k-times hierarchy, can be easily seen to belong to this class. We will see shortly that HSP_q^k and CHSP_q^k also belong to this class.

To define HSP_q^k and CHSP_q^k, it will be convenient to use the Fourier representation for input assignments to X where -1 and 1 are identified with **true** and **false** respectively and also treated as elements of $GF(q)$.

Definition: For $d \in [1,k]$, let $H_d(X)$ denote the polynomial $\sum_{i\in[1,n]} \prod_{x\in X_i^d} x$ over $GF(q)$. Then

$$\mathrm{HSP}_q^k(X) = \mathbf{true} \iff \sum_d H_d(X) \equiv 0 \pmod q$$

$$\mathrm{CHSP}_q^k(X) = \mathbf{true} \iff \forall d \in [1,k] \quad H_d(X) \equiv 0 \pmod q.$$

It is easy to see that HSP_q^k and CHSP_q^k can be defined as depth-2 and depth-3 ACC^0 circuits, respectively, in the manner considered above: for both functions h is the parity predicate on n^{k-1} inputs; for HSP_q^k, g is the Mod_q predicate, and for CHSP_q^k, g is the AND of k terms where each term is a Mod_q predicate on n inputs. (For the Mod_q predicate -1 and 1 inputs correspond to **true** and **false** respectively.)

The following theorem gives upper bounds for constructing HSP_q^{k+1} and CHSP_q^{k+1} using read-k-times and read-$(k+1)$-times branching programs. Although the constructions are extremely simple, as we will see later, the exponential gap between these bounds is real.

Theorem 2: Both $\mathrm{HSP}_q^{k+1}(X)$ and CHSP_q^{k+1} can be computed by

1. deterministic read-$(k+1)$-times branching programs of linear size, and
2. deterministic read-k-times branching programs of size $\exp\{O(N^{1/k+1})\}$.

Proof (Sketch): The proof for Part 1 follows from earlier observations. For Part 2, the construction is quite simple and we will describe it only for HSP_q^{k+1}; the construction for CHSP_q^{k+1} uses similar ideas. The branching program is composed of k blocks, each of which is a read-once branching program. For $d \in [1,k-1]$, the d^{th} block computes polynomial $H_d(X)$ and the k^{th} block computes the sum of H_k and H_{k+1}. It is easy to compose them so that together they verify that $\sum_d H_d(X) \equiv 0 \pmod q$. The former part is easy: we can easily build a read-once branching program to compute H_d, for $d \in [1,k-1]$. Thus, the first $k-1$ blocks can be computed in polynomial size.

For the k^{th} block, we start off by reading the variables of each of the k-planes in order, for computing H_k (as was done in the first $k-1$ blocks). Simultaneously, we also keep track of the partial products for each of the n $(k+1)$-planes which are orthogonal to the k-planes. At the end, we have the products for all the $(k+1)$-planes from which we can also compute H_{k+1}. To keep this "state", the branching program needs at most 2^n nodes, from which the upper bound on the size follows. □

3. The Lower Bound Technique

Non-deterministic and randomized branching programs are natural generalizations of deterministic branching programs analogous to other computing models such as Turing machines. We refer the reader to [Raz91] for a formal definition of these and related models. Recall that in a non-deterministic branching program [BRS93], the edges are either unlabeled or labeled with "$x = 0$" or "$x = 1$", for some variable x. The inputs for which it computes a 1 are those that have at least one consistent path

from the source to an accepting sink. A randomized branching program [Sau97a] is syntactically similar to a deterministic branching program and has two types of variables — input variables and stochastic variables. We say that it has 2-sided error ϵ for computing a function f, if and only if for each input σ, it computes the correct value of $f(\sigma)$ for at least $1-\epsilon$ fraction of the settings to the stochastic variables. (We can also consider the more general model obtained by defining a probability distribution on deterministic read-k-times branching programs. Our bounds apply to this model as well.) In this section, we describe the basic technique that reduces the lower bound problem for non-deterministic and randomized read-k-times branching programs to analyzing the structure of special types of functions called *pseudo-rectangles*.

For disjoint sets $S, T \subseteq X$, we call a function $R(X)$ a *pseudo-rectangle with respect to the sets* $S, T \subseteq X$, if $R(X)$ can be expressed as $R'(X \backslash T) \wedge R''(X \backslash S)$. Intuitively, $R = R' \wedge R''$, where only R' depends on S and only R'' depends on T and they share common variables $X \backslash (S \cup T)$. The motivation for this definition is the following: Borodin *et al.* [BRS93] showed that one way to prove lower bounds on the sizes of non-deterministic read-k-times branching programs computing a boolean function f is to analyze the "covering" of the satisfying assignments of f by special types of functions called "(k,p)-rectangles". (Similar approaches were taken by Okolnishnikova [Oko93] and Sauerhoff [Sau97a] for deterministic and randomized branching programs, respectively.) Although this reduction is a significant step, these rectangles are still too complex for our purposes. By using a combinatorial argument, we show that for a suitable choice of parameters, any rectangle can be transformed into a pseudo-rectangle, which is structurally easier to analyze. Later, we will show that on the hypercube, pseudo-rectangles possess even nicer properties that essentially allow us to study the lower bound problem for the hyperplanar predicates in a restricted one-dimensional setting.

We call a function $R(X)$ a (k,p)-*rectangle* if there exists a family $\{X_1, X_2, \ldots, X_{kp}\} \subseteq 2^X$ such that

- $R(X) = \bigwedge_{i \in [1,kp]} g_i(X_i)$, for some g_i's,
- $|X_i| \leq \lceil N/p \rceil$, and
- each variable appears in at most k sets in the family.

The relationship between rectangles and branching program size is given in Proposition 3 below. (This is a special case of a general result involving probability distributions over input assignments.)

Proposition 3 ([BRS93, Sau97a]): Let f be an arbitrary boolean function on a variable set X of size N. Fix an integer k and let $p \in [1, N]$.

1. Suppose $\left|R^{-1}(1)\right| \leq t$ for any (k,p)-rectangle $R \leq f$. Then, any non-deterministic read-k-times branching program for f of size s satisfies $(2s)^{2kp} \geq \left|f^{-1}(1)\right|/t$.
2. Suppose any (k,p)-rectangle $R(X)$ satisfies $\left|R^{-1}(1) \cap f^{-1}(1)\right| \leq a\left|R^{-1}(1)\right| + t$. Then, any randomized read-k-times branching program for f of size s and 2-sided error ε satisfies $(2skn)^{kp} \geq \{(1-a)\left|f^{-1}(1)\right| - \varepsilon 2^N\}/t$.

We now show that (k,p)-rectangles can be transformed to pseudo-rectangles, for a suitable choice of p. This strengthens the arguments in Borodin *et al.* [BRS93] who obtain a similar but somewhat weaker result for functions that are defined on a pair of variable sets.

Lemma 4: Let X be a variable set of size N and set $p = 144 \cdot k \cdot 2^k$. Then, any (k,p)-rectangle $R(X)$ is a pseudo-rectangle with respect to some sets S and T each having at least $(2/3) \cdot N/2^k$ variables.

Proof: Let $R(X) = \bigwedge_{i \in [1,kp]} g_i(X_i)$, for some g_i's be a (k,p)-rectangle. Arbitrarily define dummy singleton sets X_j, $j \in [kp+1, \ell]$, for some ℓ, so that each variable appears in *exactly* k sets. Let $\chi_1, \chi_2 \ldots \chi_\ell$ be random variables corresponding to independent Bernoulli trials. Set

$$S = \{x \in X \ : \ \forall i \in [1,\ell][x \in X_i \Longrightarrow \chi_i = 0]\}$$

$$T = \{x \in X \ : \ \forall i \in [1,\ell][x \in X_i \Longrightarrow \chi_i = 1]\}$$

It follows that S and T are disjoint. Moreover, for each i, either $X_i \cap S$ or $X_i \cap T$ is empty. Therefore, each $g_i(X)$ can be written as either $g_i(X_i \backslash T)$ or $g_i(X_i \backslash S)$, implying that $R(X)$ is a pseudo-rectangle with respect to S and T. To complete the proof, we show that with positive probability both S and T have at least $(1-\delta) \cdot N/2^k$ variables, where $\delta = 1/3$.

For $x \in X$, let Z_x be the indicator random variable for the event "$x \in S$" and let $Z = \sum_x Z_x = |S|$. Note that $x \in S$ if and only if $\chi_i = 0$, for each of the k sets X_i that contain x. Therefore, $\mathrm{E}[Z] = \sum_x \mathrm{E}[Z_x] = N/2^k$. Using Chebyschev's inequality, we will show that Z is close to its expected value with high probability.

The variance of Z is given by $\mathrm{var}[Z] = \sum_x \mathrm{var}[Z_x] + \sum_{x \neq y} \mathrm{cov}(Z_x, Z_y)$, where $\mathrm{cov}(Z_x, Z_y) = \mathrm{E}[Z_x Z_y] - \mathrm{E}[Z_x]\mathrm{E}[Z_y]$ denotes the covariance of Z_x and Z_y. For any x, Z_x is a Bernoulli random variable, so $\mathrm{var}[Z_x] \leq \mathrm{E}[Z_x] = 1/2^k$. Observe that if no set X_i contains both x and y, then the events Z_x and Z_y are independent implying that $\mathrm{cov}(Z_x, Z_y) = 0$. On the other hand, the number of pairs (x,y) such that some set X_i contains both x and y is at most $\lceil N/p \rceil^2 \cdot kp \leq 4kN^2/p$. (The singleton sets do not contribute to this bound.) For each such pair, $\mathrm{cov}(Z_x, Z_y) \leq \mathrm{E}[Z_x Z_y] \leq \mathrm{E}[Z_x] = 1/2^k$. For $N \geq 36 \cdot 2^k$, and for $p = 144 \cdot k \cdot 2^k$, we obtain

$$\mathrm{var}[Z] \leq \frac{N}{2^k} + \frac{4 \cdot k \cdot N^2}{144 \cdot k \cdot 2^k} \leq \frac{N^2}{36 \cdot 2^k} + \frac{N^2}{36 \cdot 2^k} = \frac{N^2}{18 \cdot 2^k} = \frac{\delta^2 \cdot \mathrm{E}[Z]^2}{2}.$$

Finally, using Chebyschev's inequality, we have

$$\Pr[Z \leq (1-\delta) \cdot E[Z]] \leq \frac{\mathrm{var}[Z]}{\delta^2 \cdot E[Z]^2} < \frac{1}{2}.$$

In a similar fashion, we obtain $\Pr[|T| \leq (1-\delta) \cdot N/2^k] < 1/2$. We conclude that there is at least one setting of the χ_i's in which both S and T have at least $(1-\delta) \cdot N/2^k = (2/3) \cdot N/2^k$ variables. □

In Theorem 6 below, we will derive two special forms for pseudo-rectangles on the $(k+1)$-dimensional hypercube, for arbitrary k. Suppose R is a pseudo-rectangle with respect to the sets S and T. For both these forms we will obtain variable sets $A = \{a_1, a_2, \ldots, a_m\} \subseteq S$ and $B = \{b_1, b_2, \ldots, b_m\} \subseteq T$, for sufficiently large m. The first form states that in some dimension d, there is a set of m d-planes such that the i^{th} d-plane in this set contains $\{a_i, b_i\}$. The second form states that there is a set of m hyperplanes such that the i^{th} d-plane in this set contains $\{a_i, b_i\}$ and every other hyperplane in the hypercube contains at most one variable of $A \cup B$. The first form will be useful for proving lower bounds for CHSP_q^k whereas the second form will be useful for HSP_q^k. The following combinatorial lemma forms the basis for the proof of Theorem 6; it is inspired by the $k = 2$ case due to Ponzio [Pon97].

Lemma 5: Let S and T be disjoint sets of variables in the $(k+1)$-dimensional hypercube $[1,n]^{k+1}$ each having size at least $(2/3) \cdot N/2^k$, where $N = n^{k+1}$. Then, there are at least $\epsilon(k+1)n/2$ hyperplanes each containing ϵn^k variables of S and $\epsilon n^k/2$ variables of T, where $\epsilon = [3(k+1)2^{k+1}]^{-1}$.

Proof: Call any hyperplane *S-dense* (*T-dense*) if it contains at least ϵn^k (respectively, $\epsilon n^k/2$) variables of S (respectively, T).

Let s_d denote the fraction of d-planes that are S-dense. Set $s = (\sum_d s_d)/(k+1)$ so that by the arithmetic-geometric mean inequality, $\prod_{d\in[1,k+1]} s_d \leq s^{k+1}$. By counting the variables of S in the d-planes, and summing over all d, we see that

$$|S| \leq \prod_{d\in[1,k+1]} (s_d n) \quad + \sum_{d\in[1,k+1]} \epsilon n^k (1-s_d) n \leq N\left[s^{k+1} + \epsilon(k+1)\right].$$

Rearranging the terms in the equation above and substituting the value of ϵ and the bound on $|S|$,

$$s^{k+1} \geq |S|/N - \epsilon(k+1) \geq (2/3)\cdot(1/2^k) - (1/3)\cdot(1/2^{k+1}) = 1/2^{k+1},$$

implying that $s \geq 1/2$.

Now apply the arithmetic-geometric inequality again to obtain $\prod_d (1-s_d) \leq (1-s)^{k+1} \leq 1/2^{k+1}$. Thus, the number of variables of T that lie in S-dense hyperplanes is at least

$$|T| - N/2^{k+1} \geq (2/3)\cdot N/2^k - 1/2^{k+1} = \epsilon(k+1)N.$$

It follows that at least $\epsilon(k+1)n/2$ S-dense hyperplanes are also T-dense because otherwise the number of variables of T that lie in S-dense hyperplanes is at most

$$N\left[\left(\sum_d s_d - \epsilon(k+1)/2\right)\cdot(\epsilon/2) + \epsilon(k+1)/2\right] \leq N(k+1)\left[(1-\epsilon/2)\cdot(\epsilon/2) + \epsilon/2\right] < \epsilon(k+1)N.$$

□

Theorem 6: Let X be the variable set of the $(k+1)$-dimensional hypercube of size $N = n^{k+1}$. Set $p = 144\cdot k\cdot 2^k$ and $\epsilon = [3(k+1)2^{k+1}]^{-1}$. Then, any (k,p)-rectangle $R(X)$ can be expressed as a pseudo-rectangle with respect to sets $A = \{a_1, a_2, \ldots a_m\}$ and $B = \{b_1, b_2, \ldots b_m\}$ in two ways as described below:

(i) Here $m = \epsilon n/2$ and for some dimension d, each pair $\{a_i, b_i\}$ lies in some unique d-plane, for $i \in [1,m]$

(ii) Here $m = \epsilon n/(4k)$ and for each $i \in [1,m]$, there is exactly one hyperplane containing $\{a_i, b_i\}$, but every other hyperplane contains at most one element of $A \cup B$.

Proof: Applying Lemma 4, $R(X)$ is a pseudo-rectangle with respect to disjoint sets S and T, each of size at least $(2/3)\cdot N/2^k$. Part (i) of the Lemma follows easily by applying Lemma 5 because for some dimension d, at least $\epsilon n/2$ d-planes must be dense in both S and T.

To derive part (ii), similarly to part (i), we first fix a d and a set Γ of $\epsilon n/2$ d-planes each containing ϵn^k variables of S and $\epsilon n^k/2$ variables of T. Now we choose the variables $a_i \in A$ and $b_i \in B$ in stages as follows: In the first stage, we pick an arbitrary d-plane P in Γ. We let b_1 be an arbitrary element of T in P and eliminate the variables of the other k hyperplanes containing b_1 from our consideration. (This also eliminates some variables in P.) Next, we choose a_1 to be an arbitrary element of S in P and repeat the same process with respect to a_1. Finally, we delete all the variables of P. Observe that this process eliminates $2kn^{k-1}$ variables from each of the other d-planes in Γ. More generally, in the i^{th} stage, we only consider those elements that have not been eliminated in the previous stages and in the manner described above, we choose pairs $b_i \in T$, $a_i \in S$ sharing some d-plane in Γ and then delete the variables of this plane and the other $2k$ hyperplanes containing a_i and b_i. We can continue this process for $\epsilon n/(4k)$ stages after which we obtain the required sets A and B. □

4. Lower Bounds for Read-k-times Branching Programs

In this section, we give *exponential* size bounds for non-deterministic and randomized read-k-times branching programs computing $\mathrm{HSP}_q^{k+1}(X)$ and $\mathrm{CHSP}_q^{k+1}(X)$. Consider the easy case $k = 1$ and $q = 3$ for the purposes of motivating the proof of the exponential size lower bound for read-once branching programs computing $\mathrm{C}HSP_3^2$. The following notation will be useful: for a polynomial $F(X)$, and a partial assignment ρ to the variables $Z \subseteq X$, let $F\lceil_\rho(Y)$ denote the polynomial depending on the variables $Y = X \backslash Z$, obtained by restricting the variables of Z to values specified by ρ.

When $k = 1$, X corresponds to a matrix of size $N = n^2$. It can be shown that any assignment that sets all the variables in X except for those in some fixed 5×5 sub-matrix can always be extended to a complete assignment that satisfies $\mathrm{C}HSP_3^2$, implying that the number of satisfying assignments is at least $2^N/2^{25}$.

In order to apply to apply Proposition 3 part 1, we consider an arbitrary (k, p)-rectangle $R \leq \mathrm{C}HSP_3^2$. Setting $p = 2$, it follows that $R(X)$ is of the form $R'(S) \wedge R''(T)$, for *disjoint* S and T of size $N/2$. Thus, we get pseudo-rectangles for free without having to resort to the machinery of Lemma 4. (For $k \geq 2$, we do not obtain such a form directly.) Lemma 5 implies that there must be $m = \Omega(n)$ rows or columns each containing at least one element of S and one element of T. Let $A = \{a_1, a_2, \ldots, a_m\} \subseteq S$ and $B = \{b_1, b_2, \ldots, b_m\} \subseteq T$ be variables occurring, say, in the first m rows. Since $R \leq \mathrm{C}HSP_3^2$, any assignment that satisfies $R(X)$ must set $H_1(X)$ to 0 mod q. For any assignment ρ to $X \backslash (A \cup B)$, we obtain $H_1\lceil_\rho(A \cup B) = \sum_i u_i(a_i b_i) + c$, where $u \in \{-1, 1\}^m$ and $c \in GF(q)$ are determined by ρ. Suppose we can show for some fixed $\lambda < 1$ that $R\lceil_\rho(A \cup B)$ can cover only $(2\lambda)^m$ of the assignments to $A \cup B$ that satisfy the above equation. Since ρ was picked arbitrarily, the number of satisfying assignments of $R(X)$ is therefore at most $2^N \lambda^m$, which is exponentially small in the number of satisfying assignments for $\mathrm{C}HSP_3^2$. Applying Proposition 3 part 1 gives the exponential size lower bound for read-once branching programs computing $\mathrm{C}HSP_3^2$.

The two key observations in the description above are that (a) $R\lceil_\rho(A \cup B) = R'\lceil_\rho(A) \wedge R''\lceil_\rho(B)$ is a standard rectangle as considered in communication complexity and (b) but for the coefficients c and u_i's, the polynomial $H_1\lceil_\rho(A \cup B)$ is similar to the standard inner product function over $GF(q)$ — a function which has been studied extensively in communication complexity (for example, see [KN97]). An important difference that makes the problem non-trivial is that the assignments to the a_i's and the b_i's are distributed over $\{-1, 1\}^m$ rather than $GF(q)^m$. Thus our goal is essentially studying the rectangular complexity of an inner-product-like function in a non-standard setting.

For the predicate HSP_q^{k+1}, an argument similar to the one outlined above can be carried out using Theorem 6 part (ii). In this case, the restricted polynomial has an additional term — a linear function of the a'a and b's. We will consider this general form in the proofs below.

4.1. Non-Deterministic Read-k-times Branching Programs

We show that any rectangle (in the communication complexity sense) can cover only an exponentially small set of the solutions to any equation involving an inner product-like function (Lemma 8) but the set of satisfying assignments of $\mathrm{HSP}_q^{k+1}(X)$ or $\mathrm{CHSP}_q^{k+1}(X)$ is "dense" (Lemma 10). Finally we use Proposition 3 and the arguments outlined above to get the desired lower bound. The proof of Lemma 8 uses the following simple fact from linear algebra.

Lemma 7: Let S be a set of vectors in $GF(q)^n$, for some n, and let $r = \dim(\text{span(S)})$. If there is an integer ℓ such that for each coordinate i, every vector in S assumes only ℓ possible values, then $|S| \leq \ell^r$.

Proof: Choose a basis $v^1, v^2, \ldots, v^r$ for the subspace generated by S and assume without loss of generality for each $i, j \in [1, r]$ that $v^i_j = 1$, if $i = j$, and 0 otherwise. Since each vector $w \in S$ can be expressed as some linear combination $\sum_i a_i v^i$, it follows that $a_i = w_i$, for $i \in [1, r]$. Therefore, only ℓ possible values are allowed for each a_i, implying that $|S| \leq \ell^r$. □

Lemma 8: Let m be an integer and q an odd prime. Fix any $u \in (GF(q)^*)^m$, $v, w \in GF(q)^m$, and $c \in GF(q)$. Let $\Sigma, \Pi \subseteq \{-1, 1\}^m$ and let R denote $\Sigma \times \Pi$. If every assignment $(\sigma, \pi) \in R$ satisfies $\sum_i u_i(\sigma_i - v_i)(\pi_i - w_i) + c \equiv 0 \pmod{q}$, then $|R| \leq 2^m$.

Proof: Pick any $\sigma^0 \in \Sigma$ and $\pi^0 \in \Pi$. Treating the elements of Σ and Π as vectors over $GF(q)$, set

$$\tilde{\Sigma} = \{\tilde{\sigma} \ : \ \tilde{\sigma}_i = u_i(\sigma_i - \sigma^0_i), \sigma \in \Sigma\} \qquad \text{and} \qquad \tilde{\Pi} = \{\pi - \pi^0 \ : \ \pi \in \Pi\}.$$

Note that $\left|\tilde{\Sigma}\right| = |\Sigma|$ and $\left|\tilde{\Pi}\right| = |\Pi|$. Moreover, for any $\tilde{\sigma} \in \tilde{\Sigma}$ and $\tilde{\pi} \in \tilde{\Pi}$, it can be verified that $\tilde{\sigma} \cdot \tilde{\pi} = 0$. (For two vectors $u, v \in GF(q)^m$, $u \cdot v = \sum_i u_i v_i \bmod q$ denotes the standard inner product over $GF(q)$.) If we let V = vector space generated by $\tilde{\Sigma}$ and W = vector space generated by $\tilde{\Pi}$, then by the above property, V and W are orthogonal subspaces. Therefore, $\dim(V) + \dim(W) \leq m$.

Because $\Sigma \subseteq \{-1, 1\}^m$, for any fixed coordinate there are at most two fixed values that every vector in $\tilde{\Sigma}$ can assume. Applying Lemma 7 with $S = \tilde{\Sigma}$ and $\ell = 2$, we obtain $\left|\tilde{\Sigma}\right| \leq 2^{\dim(V)}$. Similarly $\left|\tilde{\Pi}\right| \leq 2^{\dim(W)}$, from which it follows that

$$|R| = |\Sigma|\,|\Pi| \leq 2^{\dim(V)+\dim(W)} \leq 2^m,$$

proving the lemma. □

We now show that the number of satisfying assignments of HSP^{k+1}_q and of CHSP^{k+1}_q are "close" to the maximum possible. The proofs are based on the following simple lemma:

Lemma 9: Let q be any odd prime. For any $u \in \{-1, 1\}^{2q-1}$, $c \in GF(q)$, there exists $v \in \{-1, 1\}^{2q-1}$ such that $u \cdot v + c \equiv 0 \pmod{q}$ and $\prod_i v_i = 1$.

Proof: It suffices to produce $y \in \{-1, 1\}^{2q-1}$ such that (i) $\sum_i y_i + c \equiv 0 \pmod{q}$ and (ii) $\prod_i y_i = \prod_i u_i$. Then we can set $v_i = u_i y_i$ for $i \in [1, 2q-1]$ so that $u \cdot v + c = \sum_i y_i + c \equiv 0 \pmod{q}$ and $\prod_i v_i = \prod_i u_i \cdot \prod_i y_i = (\prod_i u_i)^2 = 1$.

Clearly, there exists at least one $y \in \{-1, 1\}^{2q-1}$ satisfying (i). Suppose $\prod_i y_i = -\prod_i u_i$, violating (ii). Since at least q of the y_i's must be equal, assume without loss of generality that $y_1 = y_2 = \cdots = y_q$, so that $\sum_{i \in [1,q]} y_i \equiv 0 \pmod{q}$. Now, replace y_i by $-y_i$, for $i \in [1, q]$. It can be verified that (i) still holds. Because q is odd, the value of $\prod_i y_i$ will be negated implying that (ii) holds as well. □

Lemma 10: For any k and any odd prime q, the number of input assignments satisfying $\text{CHSP}^k_q(X)$ is at least $2^N/2^{(2q-1)^k}$ and the number satisfying $\text{HSP}^k_q(X)$ is at least $2^N/2^{2q-1}$, where $|X| = N$.

Proof: For each predicate, we will fix a particular subset of the variables Y. Let ρ denote an arbitrary assignment to the variables of $X \backslash Y$. We will show that ρ always be extended by an assignment ϕ to Y such that the complete assignment (ϕ, ρ) satisfies the predicate. It follows that the number of satisfying assignments is at least $2^N/2^{|Y|}$.

$\mathbf{CHSP}_q^k(X)$: In this case Y is the set of variables corresponding to the sub-hypercube $[1, 2q-1]^k$. Note that $|Y| = (2q-1)^k$, as required by the lemma. For two assignments σ and π to a set of variables, let their point-wise product $\sigma \odot \pi$ be the assignment such that $\sigma \odot \pi(x) = \sigma(x) \cdot \pi(x)$. We have to construct ϕ such that (ϕ, ρ) sets each polynomial $H_d(X)$ to 0 (mod q). We first construct k intermediate assignments $\phi_1, \phi_2, \ldots, \phi_k$. Loosely speaking, each ϕ_d will satisfy the property that (ϕ_d, ρ) sets $H_d(X)$ to 0 (mod q) but does not "influence" the value of $H_{d'}(X)$, for $d' \neq d$.

Lemma 11: Let Y be the the set of variables corresponding the sub-hypercube $[1, 2q-1]^k$ and let ρ be an arbitrary assignment to the variables of $X \backslash Y$. Then there exist assignments $\phi_1, \phi_2, \ldots \phi_k$ to Y such that for any $d \in [1, k]$, (i) $H_d\lceil_\rho(\phi_d) \equiv 0 \pmod q$ and (ii) for any $d' \neq d$, and for any assignment σ to Y, $H_{d'}\lceil_\rho(\sigma \odot \phi_d) \equiv H_{d'}\lceil_\rho(\sigma) \pmod q$.

Proof: Fix any d and substitute for ρ in the polynomial $H_d(X)$. We have

$$H_d\lceil_\rho(Y) = \sum_{i \in [1,2q-1]} \underbrace{\prod_{x \in X_i^d \backslash Y} \rho(x)}_{u_i} \cdot \prod_{x \in Y \cap X_i^d} x + \underbrace{\sum_{i \in [2q,n]} \prod_{x \in X_i^d} \rho(x)}_{c} \quad (1)$$

where $u \in \{-1, 1\}^{2q-1}$ and $c \in GF(q)$. Apply Lemma 9 to obtain $v \in \{-1, 1\}^{2q-1}$ such that $u \cdot v + a \equiv 0 \pmod q$ and $\prod_i v_i = 1$. Define the assignment ϕ_d to Y by assigning v_i to *every* variable in $Y \cap X_i^d$, for $i \in [1, 2q-1]$.

To verify (i), substitute the values assigned by ϕ_d for the variables of Y in Equation (1). We have $\prod_{x \in Y \cap X_i^d} \phi_d(x) = v_i^{\left|Y \cap X_i^d\right|}$, which simplifies to v_i, since $\left|Y \cap X_i^d\right| = (2q-1)^k$ is odd. Therefore, Equation (1) simplifies as $H_d\lceil_\rho(\phi_d) = u \cdot v + c \equiv 0 \pmod q$.

For proving (ii), fix any $d' \neq d$ and observe that we can write a expression for $H_{d'}\lceil_\rho(Y)$ similar to Equation (1), that is, $H_{d'}\lceil_\rho(Y) = (\sum_i u'_i \prod_{x \in Y \cap X_i^d} x) + c'$, for some $u' \in \{-1, 1\}^{2q-1}$ and $c' \in GF(q)$. Substitute for σ and $\sigma \odot \phi_d$ in this equation. We have

$$H_{d'}\lceil_\rho(\phi_d) = \sum_i u'_i \prod_{x \in Y \cap X_i^d} \phi_d(x) + c'$$

$$H_{d'}\lceil_\rho(\sigma \odot \phi_d) = \sum_i u'_i \prod_{x \in Y \cap X_i^d} (\sigma \odot \phi_d)(x) + c' = \sum_i u'_i \prod_{x \in Y \cap X_i^d} \sigma(x)\phi_d(x) + c'$$

We show that for each $i \in [1, 2q-1]$, $\prod_{x \in Y \cap X_i^{d'}} \phi_d(x) = 1$, which would imply the desired equality $H_{d'}\lceil_\rho(\sigma \odot \phi_d) = H_{d'}\lceil_\rho(\phi_d)$. This can be seen as follows:

$$\prod_{x \in Y \cap X_i^{d'}} \phi_d(x) = \prod_{j \in [1,2q-1]} \prod_{x \in Y \cap X_i^{d'} \cap X_j^d} \phi_d(x) = \prod_{j \in [1,2q-1]} v_j^{\left|Y \cap X_i^{d'} \cap X_j^d\right|} = \prod_{j \in [1,2q-1]} v_j = 1,$$

because for any $d \neq d'$, $\left|Y \cap X_i^{d'} \cap X_j^d\right| = (2q-1)^{k-1}$ is odd. □

(Proof of Lemma 10 continued): Apply Lemma 11 to obtain ϕ_d, for $d \in [1, k]$ and set ϕ to be their point-wise product. For any $d \in [1, k]$, by applying Lemma 11 part (ii) above repeatedly, we obtain $H_d\lceil_\rho(\phi) \equiv H_d\lceil_\rho(\phi_d) \pmod q$, which by part (i) is equivalent to 0 (mod q).

$\mathbf{HSP}_q^k(X)$: Here $Y = \{y_1, y_2, \ldots, y_{2q-1}\}$, where y_i corresponds to the element $(i, 1, 1, \ldots, 1)$ in the hypercube $[1, n]^k$. Observe that $Y \subseteq X_d^1$ for any $d \neq 1$. On the other hand, the variables of Y are split amongst the first $2q-1$ 1-planes. As before, substitute for ρ in each of the polynomials $H_d(X)$ and separate the sub-expressions that depend on ρ only. When $d = 1$, we have

$$H_1\lceil_\rho(Y) = \sum_{i\in[1,2q-1]} y_i \underbrace{\prod_{x\in X_i^1\setminus\{y_i\}} \rho(x)}_{u_i} \quad + \quad \underbrace{\sum_{i\in[2q,n]} \prod_{x\in X_i^1} \rho(x)}_{b}.$$

Thus, $H_1\lceil_\rho(Y) = \sum_i u_i y_i + b = u \cdot y + b$ (where $y = (y_1, y_2, \ldots, y_{2q-1})$).

When we substitute for ρ in $H_d(X)$, where $d \neq 1$, and sum over all such d, we obtain

$$\begin{aligned}\sum_{d\in[2,k]} H_d\lceil_\rho(Y) &= \sum_{d\in[2,k]} \left(\prod_{j\in[1,2q-1]} y_j \prod_{x\in X_1^d\setminus Y} \rho(x) \quad + \quad \sum_{i\in[2,n]} \prod_{x\in X_i^d} \rho(x) \right) \\ &= \prod_{j\in[1,2q-1]} y_j \underbrace{\sum_{d\in[2,k]} \prod_{x\in X_1^d\setminus Y} \rho(x)}_{a} \quad + \quad \underbrace{\sum_{\substack{d\in[2,k]\\ i\in[2,n]}} \prod_{x\in X_i^d} \rho(x)}_{a'}\end{aligned}$$

It follows that $H\lceil_\rho(Y) = \sum_d H_d\lceil_\rho(Y) = u \cdot y + b + a\prod_j y_j + a'$.

Apply Lemma 9 with u and $c = b + a + a'$ to obtain $v \in \{-1, 1\}^{2q-1}$ such that $u \cdot v + c \equiv 0 \pmod q$ and $\prod_j v_j = 1$. Define ϕ by assigning values v to y. The lemma follows because

$$H\lceil_\rho(\phi) = u \cdot v + b + a\prod_j v_j + a' = u \cdot v + b + a + a' = u \cdot v + c \equiv 0 \pmod q.$$

□

We now have all the pieces to prove Theorem 1.

Proof (of Theorem 1): The upper bounds follow from Theorem 2. For proving the lower bounds via Proposition 3 part 1, first consider any (k, p)-rectangle $R \leq \mathrm{CHSP}_q^{k+1}$, where $p = 144 \cdot k \cdot 2^k$, and apply Theorem 6 part (i) to obtain d and sets A and B each of size $m = \Omega(n/(k \cdot 2^k))$, so that $R(X) = R'(X\setminus B) \wedge R''(X\setminus A)$. By the symmetry in the definition of $\mathrm{CHSP}_q^{k+1}(X)$, and we can assume without loss of generality that $d = 1$. Pick an arbitrary assignment ρ to the variables of $X\setminus(A \cup B)$. Let Γ denote the set of assignments that satisfy $R\lceil_\rho(A \cup B)$. In other words, $\Gamma = \Sigma \times \Pi$, where $\Sigma = \{\sigma \ : \ R'\lceil_\rho(\sigma) = 1\}$ and $\Pi = \{\pi \ : \ R''\lceil_\rho(\pi) = 1\}$.

Because $R \leq \mathrm{CHSP}_q^{k+1}$, for any $(\sigma, \pi) \in \Gamma$, we have $H_1\lceil_\rho(\sigma, \pi) \equiv 0 \pmod q$. Since $H_1\lceil_\rho(\sigma, \pi) = \sum_i u_i\sigma_i\pi_i + c$, for some $u \in \{-1, 1\}^m$ and $c \in GF(q)$, Lemma 8 implies that $|\Gamma| \leq 2^m$. Since ρ was chosen arbitrarily, it follows that $|R^{-1}(1)| \leq 2^{N-2m}2^m = 2^{N-m}$. Combining this with Lemma 10 and applying Proposition 3 part 1, the size of any non-deterministic read-k-times branching program computing CHSP_q^{k+1} is at least

$$\left(2^{N-(2q-1)^k}/2^{N-m}\right)^{1/(2kp)} = \left(2^{m-(2q-1)^k}\right)^{1/(2kp)} = 2^{\Omega(nk^{-3}2^{-2k})},$$

proving the theorem for CHSP_q^{k+1}.

For HSP_q^{k+1}, a similar argument can be carried out using Theorem 6 part (ii). In this case, we consider the polynomial $H\lceil_\rho(A \cup B)$, which by the property satisfied by A and B can be written as $\sum_i u_i(a_i - v_i)(b_i - w_i) + c$, for some $u \in (GF(q)^*)^m$, $v, w \in GF(q)^m$ and $c \in GF(q)$ that depend only on ρ. Thus Lemma 8 can be applied again from which the lower bound follows by similar arguments. □

4.2. Randomized Read-k-times Branching Programs

Since we have already proved Lemma 10, to apply Proposition 3 part 2, we only need to show that the density of satisfying assignments of $\mathrm{HSP}_q^{k+1}(X)$ or CHSP_q^{k+1} in any (k,p)-rectangle $R(X)$ is significantly small. Consider the predicate CHSP_q^{k+1}. Similar to the proof of Theorem 1, $R(X) = R'(X\backslash B) \wedge R''(X\backslash A)$, and we pick an arbitrary assignment ρ to $X\backslash(A \cup B)$ which defines Σ and Π. We want to show that in the "rectangle" defined by Σ and Π, the distribution of the values of $H_1\lceil_\rho(A \cup B)$ is almost uniform on $GF(q)$. Because it would then imply that roughly $1/q$ fraction of the values of $H_1\lceil_\rho(A \cup B)$ are $\equiv 0 \pmod q$. Since ρ was picked arbitrarily, it follows that the density of satisfying assignments of $\mathrm{CHSP}_q^{k+1}(X)$ in $R(X)$ is roughly at most $1/q$.

The polynomial $H_1\lceil_\rho(A \cup B) = \sum_i u_i a_i b_i + c$, for some $u \in \{-1,1\}^m$, $c \in GF(q)$, can be viewed as a family of hash functions $\mathcal{H}$ of size 2^m mapping $\{-1,1\}^m$ to $GF(q)$. Each assignment σ to A determines a hash function $h_\sigma \in \mathcal{H}$ such that for any assignment π to B

$$h_\sigma(\pi) = H_1\lceil_\rho(\sigma, \pi) = \sum_i u_i \sigma_i \pi_i + c.$$

Ideally, we want to say that $\mathcal{H}$ is a universal family of hash functions and then argue that the distribution in any rectangle is balanced (as done in [MNT93], for example). But $\mathcal{H}$ is not a universal family; for example, when $m = 1$, $h \in \mathcal{H}$ is not uniform over $GF(q)$ because it can output only 2 values in $GF(q)$. However, as m grows large, we will see that $\mathcal{H}$ behaves almost like a universal family. This prompts the following definition.

Definition: Let $\mathcal{H} = \{h : \mathcal{I} \to \mathcal{O}\}$ be a family of hash functions. For $0 \le \delta \le 1$, we say that $\mathcal{H}$ is *δ-almost universal on $\mathcal{I}$*, if

(A) For any $x \in \mathcal{I}$ and $a \in \mathcal{O}$,

$$\left| \Pr_{h \in \mathcal{H}}[h(x) = a] - \frac{1}{|\mathcal{O}|} \right| \le \frac{1}{|\mathcal{O}|}(\delta/3)$$

(B) For $x \neq y$, we call (x, y) a *good pair* if for all $a, b \in \mathcal{O}$,

$$\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b] \le \frac{1}{|\mathcal{O}|^2}(1 + \delta/3),$$

and a *bad pair* otherwise. Then, for each fixed $x \in \mathcal{I}$, $\Pr_{y \in \mathcal{I}}[(x, y) \text{ is bad} \mid y \neq x] \le \delta/2$.

Note: When $\delta = 0$, we obtain the standard universal family of hash functions. In our applications, δ will be exponentially small in $|\mathcal{I}|$.

The following lemma, which generalizes the well-known "hash-mixing lemma" [MNT93], shows that rectangles corresponding to almost universal families of hash functions are "balanced". It will be proved in the next section.

Lemma 12: Let $\mathcal{H} = \{h : \mathcal{I} \to \mathcal{O}\}$ be a δ-almost universal family of hash functions. Let $\mathcal{A} \subseteq \mathcal{I}$, $\mathcal{G} \subseteq \mathcal{H}$, and $\mathcal{B} \subseteq \mathcal{O}$. If $p = |\mathcal{B}| / |\mathcal{O}|$, then

$$\left| \Pr_{x \in \mathcal{A}, h \in \mathcal{G}} [h(x) \in \mathcal{B}] - p \right| \leq \sqrt{\frac{|\mathcal{H}|}{|\mathcal{G}|\,|\mathcal{A}|} p(1-p)(1 + 2\delta\,|\mathcal{I}|)}.$$

We now show that the family of hash functions corresponding to the inner-product-like functions considered above is an almost universal hash family for appropriately large sets. Its proof depends upon the following lemma that estimates the number of $\{-1, 1\}$ solutions to a type of modular linear equation:

Lemma 13: For any integer n, $u \in (GF(q)^*)^n$, $c \in GF(q)$,

$$\left| \Pr_{z \in \{-1,1\}^n} [u \cdot z \equiv c \pmod{q}] - \frac{1}{q} \right| \leq \left(1 - \frac{1}{q}\right) \cos^n(\pi/q).$$

Proof: Let $\omega = e^{2\pi i/q}$ denote the complex q^{th} root of unity. The number S of solutions $z \in \{-1, 1\}^n$ to the equation $u \cdot z \equiv c \pmod{q}$ is *exactly* given by the formula (e.g., see [Gou72])

$$\frac{1}{q} \sum_{s \in [0, q-1]} \omega^{-cs} \prod_{j \in [1,n]} (\omega^{-u_j s} + \omega^{u_j s}) = \frac{2^n}{q} + \frac{1}{q} \sum_{s \in [1, q-1]} \omega^{-cs} \prod_{j \in [1,n]} (\omega^{-u_j s} + \omega^{u_j s}).$$

Let $|\cdot|$ denote the magnitude of a complex number. Then,

$$\begin{aligned} \left| \frac{S}{2^n} - \frac{1}{q} \right| &= \frac{1}{q \cdot 2^n} \left| \sum_{s \in [1, q-1]} \omega^{-cs} \prod_{j \in [1,n]} (\omega^{-u_j s} + \omega^{u_j s}) \right| \\ &\leq \frac{1}{2^n} \left(1 - \frac{1}{q}\right) \max_{s \in [1, q-1]} \left| \omega^{-cs} \prod_{j \in [1,n]} (\omega^{-u_j s} + \omega^{u_j s}) \right| \end{aligned} \tag{2}$$

For any s,

$$\left| \omega^{-cs} \prod_{j \in [1,n]} (\omega^{-u_j s} + \omega^{u_j s}) \right| = |\omega^{-cs}| \prod_{j \in [1,n]} |\omega^{-u_j s} + \omega^{u_j s}| = \prod_{j \in [1,n]} 2\cos(2u_j s\pi/q) \leq (2\cos(\pi/q))^n.$$

The last inequality above holds because q is odd. The lemma follows by substituting this bound in Line (2). □

Lemma 14: Let m be an integer and q an odd prime. Fix any $u \in (GF(q)^*)^m$, $v, w \in GF(q)^m$ and $c \in GF(q)$. Define the family of hash functions $\mathcal{H}$ as follows: for each $h \in \mathcal{H}$, associate a unique vector in $\{-1, 1\}^m$ (also referred to as h) and define $h(x) = \sum_i u_i(h_i - v_i)(x_i - w_i) + c \bmod q$, for $x \in \{-1, 1\}^m$. Then for large enough m, there is a set $\Gamma \subseteq \{-1, 1\}^m$, of size at least $2^m(1 - e^{m/16})$ such that $\mathcal{H}$ is $\lambda^{2\epsilon m}$-almost universal on Γ, where $\lambda = \cos(\pi/q)$ and $\epsilon = 1/80$.

Proof: Let Γ denote the set of inputs $x \in \{-1,1\}^m$ such that $x_i \neq w_i$ for at least $m/4$ coordinates. Observe that by Chernoff's bound, $|\Gamma| \geq 2^m(1-e^{-m/16})$, as desired. (If some of the w_i's do not belong to $\{-1,1\}$, then the size could be much larger. For example, if none of the w_i's belong to $\{-1,1\}$ then in fact $\Gamma = \{-1,1\}^m$.)

To prove Property (A) of the definition of an almost universal family, fix an $x \in \Gamma$ and let $x' \in GF(q)^m$ be defined by $x'_i = u_i(x_i - w_i)$ for all i. Now $h(x)$ can be expressed as $h \cdot x' + c' \bmod q$, where $c' \in GF(q)$ is independent of h. Moreover, at least $m/4$ of the x'_i's are non-zero. Therefore, by Lemma 13, for any $a \in GF(q)$,

$$\left| \Pr_{h \in \mathcal{H}}[h(x) = a] - \frac{1}{q} \right| \leq (1-1/q)\lambda^{m/4} \leq \lambda^{2\epsilon m}/3,$$

proving Property (A) for $\mathcal{H}$.

To show Property (B), fix any $x \in \Gamma$ and a set of $m/4$ coordinates K where $x_i \neq w_i$, for $i \in K$. Now consider the set $\Gamma_x \subseteq \Gamma$ such that for each $y \in \Gamma_x$, amongst the coordinates in K, $y_i = x_i$ in at least $\gamma m/4$ places and $y_i = -x_i$ in at least $\gamma m/4$ places, where $\gamma = 1/9$. We will show shortly that (x,y) is a good pair for each $y \in \Gamma_x$. Assuming this to be true, the number of bad pairs can be estimated using Chernoff's bound. We have

$$\Pr_{y \in \Gamma}[y \notin \Gamma_x \mid x \neq y] \leq \Pr_{y \in \{-1,1\}^m}[y \notin \Gamma_x \mid x \neq y] \leq 2e^{-(1-2\gamma)^2 m/16} \leq \lambda^{2\epsilon m}/2,$$

since $\lambda \geq 1/2$, when $q \geq 3$.

Fix any $y \in \Gamma_x$. Our goal is to show that (x,y) is a good pair. Let $a, b \in GF(q)$ and consider the h's such that $h(x) = a$ and $h(y) = b$. Let I (respectively, J) be a set of $\gamma m/4$ coordinates in K where x and y agree (respectively, disagree). Let x' be defined as before. Similarly, let $y' \in GF(q)^m$ be such that $y'_i = u_i(y_i - w_i)$ for all i. Again, $h(x) = h \cdot x' + c' \bmod q$ and $h(y) = h \cdot y' + d' \bmod q$, where c' and d' are independent of h. Fixing some choice of h_k's, where $k \notin I \cup J$, we have

$$\sum_{i \in I} h_i x'_i + \sum_{j \in j} h_j x'_j \equiv a - (c' + \sum_{k \notin I \cup J} h_k x'_k) \pmod q, \tag{3}$$

$$\sum_{i \in I} h_i y'_i + \sum_{j \in j} h_j y'_j \equiv a - (d' + \sum_{k \notin I \cup J} h_k y'_k) \pmod q, \tag{4}$$

Set a' and b' to be the expressions in the right hand sides of Equation (3) and Equation (4) respectively. Subtract Equation (4) from Equation (3). Since $y_i = x_i$ for $i \in I$, and $y_j = -x_j$ for $j \in J$, we obtain

$$\sum_{j \in J} 2h_j x_j \equiv a' - b' \pmod q \tag{5}$$

Applying Lemma 13 with $n = \gamma m/4$, the number of solutions $(h_j)_{j \in J}$ to this equation is bounded by $(1/q) \cdot 2^{\gamma m/4}(1 + (q-1)\lambda^{\gamma m/4})$. For each such solution, we substitute in Equation (3) to get

$$\sum_{i \in I} h_i x'_i \equiv a' - (\sum_{j \in J} h_j x'_j) \pmod q.$$

Since $x'_i \neq 0$ for $i \in I$, by Lemma 13, the above equation has at most $(1/q) \cdot 2^{\gamma m/4}(1 + (q-1)\lambda^{\gamma m/4})$ solutions $(h_i)_{i \in I}$. Thus,

$$\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b] \leq (1/q)^2 \cdot (1 + (q-1)\lambda^{\gamma m/4})^2 \leq (1/q)^2 \cdot (1 + \lambda^{2\epsilon m}/3),$$

proving that (x, y) is a good pair. $\square$

Using Lemma 12 and Lemma 14, we argue below that each sufficiently large rectangle is balanced.

Lemma 15: Let m be an integer and q an odd prime. Fix any $u \in (GF(q)^*)^m$, $v, w \in GF(q)^m$ and $c \in GF(q)$. Let $\Sigma, \Pi \subseteq \{-1, 1\}^m$ and let R denote $\Sigma \times \Pi$. If S denotes the set of assignments $(\sigma, \pi) \in R$ such that $\sum_i u_i(\sigma_j - v_i)(\pi_i - w_i) \ + \ c \equiv 0 \pmod{q}$, then for large enough m,

$$|R \cap S| \leq (1/q)\,|R| + 2\lambda^{\epsilon m} 2^{2m}, \tag{6}$$

where $\lambda = \cos(\pi/q)$ and $\epsilon = 1/80$.

Proof: If $|R| \leq 2\lambda^{\epsilon m} 2^{2m}$, the lemma holds trivially. Therefore, we can assume that $|R| \geq 2\lambda^{\epsilon m} 2^{2m}$, implying that $|\Sigma| \geq 2\lambda^{\epsilon m} 2^m$ and $|\Pi| \geq 2\lambda^{\epsilon m} 2^m$. Associate Σ with a family of hash functions $\mathcal{H}$ as described in the statement of Lemma 14. By Lemma 14, $\mathcal{H}$ is $\lambda^{2\epsilon m}$-almost universal on some set Γ of size at least $2^m(1 - e^{m/16})$. Let Π' denote the set $\Pi \cap \Gamma$ so that $|\Pi'| \geq (2\lambda^{\epsilon m} - e^{-m/16})2^m \geq \lambda^{\epsilon m} 2^m$, since $\lambda \geq 1/2$.

Applying Lemma 12, with $\mathcal{H} = \{-1, 1\}^m$, $\mathcal{G} = \Sigma$, $\mathcal{I} = \mathcal{A} = \Pi'$ and $\mathcal{O} = GF(q)$, we obtain

$$\begin{aligned}
\left| \Pr_{\sigma \in \Sigma, \pi \in \Pi'}[(\sigma, \pi) \in S] - 1/q \right| &\leq \sqrt{\frac{2^m}{|\Sigma|\,|\Pi'|}(1/q)(1 - 1/q)(1 + \lambda^{2\epsilon m}\,|\Pi'|)} \\
&\leq \sqrt{\frac{2^m}{|\Sigma|\,|\Pi'|}(1/4)(4\lambda^{2\epsilon m} 2^m)} \\
&= \lambda^{\epsilon m} 2^m \sqrt{\frac{1}{|\Sigma|\,|\Pi'|}}
\end{aligned}$$

Therefore,

$$\begin{aligned}
|R \cap S| &\leq |\Sigma|\,|\Pi \backslash \Pi'| + |\Sigma|\,|\Pi'| \left((1/q) + \lambda^{\epsilon m} 2^m \sqrt{\frac{1}{|\Sigma|\,|\Pi'|}} \right) \\
&\leq e^{-m/16} 2^{2m} + (1/q)\,|R| + \lambda^{\epsilon m} 2^m \sqrt{|\Sigma|\,|\Pi'|} \\
&\leq (1/q)\,|R| + (e^{-m/16} + \lambda^{\epsilon m}) 2^{2m} \\
&\leq (1/q)\,|R| + 2\lambda^{\epsilon m} 2^{2m},
\end{aligned}$$

which proves Equation (6) and hence the lemma. $\square$

Applying Proposition 3 part 2, Lemma 10, Lemma 15, and using arguments similar to Theorem 1, we obtain

Theorem 16: Any randomized read-k-times branching program for $\mathrm{HSP}_q^{k+1}(X)$ with 2-sided error $(1/3) \cdot 2^{-(2q-1)}$ must have size at least $\exp\left\{\Omega\left(n^{1/k+1} 2^{-2k} k^{-4}\right)\right\}$ and for $\mathrm{HSP}_q^{k+1}(X)$ with 2-sided error $(1/3) \cdot 2^{-(2q-1)^k}$ must have size at least $\exp\left\{\Omega\left(n^{1/k+1} 2^{-2k} k^{-3}\right)\right\}$.

5. Almost Universal Family of Hash Functions

This section is devoted to the proof of Lemma 12.

Proof (of Lemma 12): Assume that $p \leq 1/2$, because otherwise we can prove the lemma with $\overline{\mathcal{B}}$ in place of $\mathcal{B}$. Consider the matrix M whose columns are indexed by hash functions from $\mathcal{H}$ and rows by elements from $\mathcal{I}$. Define

$$M_{x,h} = \begin{cases} 1 & \text{if } h(x) \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

Using the Cauchy-Schwartz inequality,

$$\begin{aligned} \left| \Pr_{x \in \mathcal{A}, h \in \mathcal{G}}[h(x) \in \mathcal{B}] - p \right| &= |\mathrm{E}_{h \in \mathcal{G}} \mathrm{E}_{x \in \mathcal{A}}[M_{x,h} - p]| \\ &\leq \sqrt{\mathrm{E}_{h \in \mathcal{G}} \{\mathrm{E}_{x \in \mathcal{A}}[M_{x,h} - p]\}^2} \\ &\leq \sqrt{\frac{|\mathcal{H}|}{|\mathcal{G}|} \mathrm{E}_{h \in \mathcal{H}} \{\mathrm{E}_{x \in \mathcal{A}}[M_{x,h} - p]\}^2} \end{aligned} \tag{7}$$

We estimate the expression in the right hand side as follows.

$$\begin{aligned} & \mathrm{E}_{h \in \mathcal{H}} \{\mathrm{E}_{x \in \mathcal{A}}[M_{x,h} - p]\}^2 \\ &= \mathrm{E}_{h \in \mathcal{H}} \{(\mathrm{E}_{x \in \mathcal{A}}[M_{x,h} - p])(\mathrm{E}_{y \in \mathcal{A}}[M_{y,h} - p])\} \\ &= \mathrm{E}_{x \in \mathcal{A}, y \in \mathcal{A}} \mathrm{E}_{h \in \mathcal{H}}[M_{x,h} M_{y,h} - p(M_{x,h} + M_{y,h}) + p^2] \end{aligned} \tag{8}$$

Let us bound each term in the above expression. First, by Property (A) in the definition of $\mathcal{H}$, we have $\mathrm{E}_{h \in \mathcal{H}}[M_{z,h}] \geq p(1 - \delta/3)$, for $z = x, y$. Therefore

$$\mathrm{E}_{x \in \mathcal{A}, y \in \mathcal{A}} \mathrm{E}_{h \in \mathcal{H}}[M_{x,h} + M_{y,h}] \geq 2p(1 - \delta/3).$$

Next, for $x, y \in \mathcal{A}$, we bound $\mathrm{E}_{h \in \mathcal{H}}[M_{x,h} M_{y,h}]$ by considering the two cases:

Case 1: Either $x = y$ or (x, y) is a bad pair. For each such pair, by Property (A), we have

$$\mathrm{E}_{h \in \mathcal{H}}[M_{x,h} M_{y,h}] \leq \mathrm{E}_{h \in \mathcal{H}}[M_{x,h}] \leq p(1 + \delta/3).$$

Case 2: (x, y) is a good pair. In this case, by definition

$$\mathrm{E}_{h \in \mathcal{H}}[M_{x,h} M_{y,h}] \leq p^2(1 + \delta/3).$$

Let U denote the number of pairs considered in Case 1 above. By Property (B), $U \leq |\mathcal{A}| + \delta |\mathcal{I}| |\mathcal{A}| /2$. Therefore, we can combine the two cases to obtain

$$\begin{aligned} \mathrm{E}_{x \in \mathcal{A}, y \in \mathcal{A}} \mathrm{E}_{h \in \mathcal{H}}[M_{x,h} M_{y,h}] &\leq p(1 + \delta/3) \cdot U / |\mathcal{A}|^2 + p^2(1 + \delta/3) \cdot (1 - U / |\mathcal{A}|^2) \\ &= p(1 - p)(1 + \delta/3) \cdot U / |\mathcal{A}|^2 + p^2(1 + \delta/3) \\ &\leq p(1 - p)(1 + \delta |\mathcal{I}|) / |\mathcal{A}|) + p^2(1 + \delta/3) \end{aligned}$$

The last inequality is implied by the following:

$$(1+\delta/3)\cdot U/|\mathcal{A}| \leq (1+\delta/3)(1+\delta|\mathcal{I}|/2) = 1+\delta(1/3+|\mathcal{I}|/2+\delta|\mathcal{I}|/6) \leq 1+\delta|\mathcal{I}|.$$

Substituting these bounds in Line 8, we obtain

$$\begin{aligned}
\mathrm{E}_{h\in\mathcal{H}}\{\mathrm{E}_{x\in\mathcal{A}}[M_{x,h}-p]\}^2 &\leq p(1-p)(1+\delta|\mathcal{I}|)/|\mathcal{A}| + p^2(1+\delta/3) - 2p^2(1-\delta/3) + p^2 \\
&= p(1-p)(1+\delta|\mathcal{I}|)/|\mathcal{A}| + p^2\delta \\
&\leq p(1-p)(1+\delta|\mathcal{I}|)/|\mathcal{A}| + p(1-p)\delta|\mathcal{I}|/|\mathcal{A}| \\
&\qquad\qquad \text{(Because } p \leq 1/2 \text{ and } |\mathcal{A}| \leq |\mathcal{I}|\text{)} \\
&= p(1-p)(1+2\delta|\mathcal{I}|)/|\mathcal{A}|
\end{aligned}$$

Finally, from Equation (7), we conclude

$$\left|\Pr_{x\in\mathcal{A},h\in\mathcal{G}}[h(x)\in\mathcal{B}] - p\right| \leq \sqrt{\frac{|\mathcal{H}|}{|\mathcal{G}|\,|\mathcal{A}||}p(1-p)(1+2\delta|\mathcal{I}|)}.$$

□

6. Discussion

Let BP_k (respectively, NBP_k, $BPBP_k^\varepsilon$) denote the class of functions computable by polynomial sized read-k-times (respectively, non-deterministic, randomized read-k-times with 2-sided error ε) branching programs. We have shown that for all $k > 0$, $BP_{k+1}\backslash NBP_k \neq \phi$ via the predicates CHSP_q^k and HSP_q^k. We have also shown via the predicate HSP_q^k that for some fixed constant ε, $BP_{k+1}\backslash BPBP_k^\varepsilon \neq \phi$. Observe that $\neg\mathrm{CHSP}_q^{k+1}$ can be easily computed by a non-deterministic read-once branching program that "guesses" a d and verifies that the polynomial H_d does not evaluate to 0 modulo q. Thus, we also obtain the relation $NBP_k\backslash BPBP_k^\varepsilon \neq \phi$.

It would be interesting to improve the error bounds of the randomized read-k-times branching programs computing HSP_q^{k+1} or CHSP_q^{k+1}. This is not trivial because many of the probability amplification techniques standardly used for many other computing models do not translate for randomized read-k-times branching programs. The value of ε_k that we have obtained for each predicate is limited by the density of the satisfying assignments because we are essentially using Yao's technique to bound the size of a deterministic branching program that is allowed to err on ε_k fraction of the inputs. Possible approaches for improving the value of ε_k include improving the bound on the density of the satisfying assignments or considering non-uniform probability distributions that are biased towards the satisfying assignments.

References

[BCL+94] J.R. Burch, E.M. Clarke, D.E. Long, K.L. MacMillan, and D.L. Dill. Symbolic model checking for sequential circuit verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(4):401–424, April 1994.

[Bor93] Borodin. Time-space tradeoffs (getting closer to the barrier?). In *ISAAC: 4th International Symposium on Algorithms and Computation*, pages 209–220, 1993.

[BRS93] Borodin, Razborov, and Smolensky. On lower bounds for read-k-times branching programs. *Computational Complexity*, 3:1–18, 1993.

[Bry86] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[BS90] Boppana and Sipser. The complexity of finite functions. In *Handbook of Theoretical Computer Science, Ed. Jan van Leeuwen (Volume A (= "1")*. Elsevier and MIT Press, 1990.

[BSSW93] B. Bollig, M. Sauerhoff, D. Sieling, and I. Wegener. Read k times ordered binary decision diagrams – efficient algorithms in the presence of null chains. Technical Report Forschungsbericht Nr. 474, Universität Dortmund, 1993.

[GM94] J. Gergov and C. Meinel. Efficient analysis and manipulation of OBDDs can be extended to FBDDs. *IEEE Transactions on Computers*, 43:1197–1209, 1994. Techreport 92-10.

[Gou72] H. W. Gould. *Combinatorial identities; A Standardized Set of Tables Listing 500 Binomial Coefficient Summations.* Morgantown, W. Va., 1972.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, Cambridge [England] ; New York, 1997.

[MNT93] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107:121–133, 1993.

[Neč66] E. Nečiporuk. On a Boolean function. *Soviet Math. Doklady*, 7:999–1000, 1966.

[Oko93] E. Okol'nishnikova. On lower bounds for branching programs. *Siberian Advances in Mathematics*, 3(1):152–166, 1993.

[Oko97] E. Okol'nishnikova. On the hierarchy of nondeterministic branching k-programs. In *Fundamentals of computation theory : 11th International Symposium*, volume 1102 of *Lecture Notes in Computer Science*, pages 376–387, Krakow, Poland, 1997. Springer Verlag.

[Pon95] Stephen Ponzio. *Restricted Branching Programs and Hardware Verification.* PhD thesis, Massachusetts Institute of Technology, 1995.

[Pon97] Stephen Ponzio. Towards a new lower bound for read-twice branching programs. Private Communication, 1997.

[Raz91] A. A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. *Lecture Notes in Computer Science*, 529:47–61, 1991.

[Sau97a] M. Sauerhoff. A lower bound for randomized read-k-times branching programs. Technical Report TR-97-019, Electronic Colloquium on Computational Complexity, 1997.

[Sau97b] M. Sauerhoff. On nondeterminism versus randomness for read-once branching programs. Technical Report TR-97-030, Electronic Colloquium on Computational Complexity, 1997.

[SS93] Janos Simon and Mario Szegedy. A new lower bound theorem for read only once branching programs and its applications. In *Advances in COmputational Complexity (J. Cai, editor)*, volume 13 of *DIMACS Series in Discrete Mathematics*, pages 183–193. AMS, 1993.

[SW95] Detlef Sieling and Ingo Wegener. Graph driven BDDs—a new data structure for Boolean functions. *Theoretical Computer Science*, 141(1–2):283–310, 17 April 1995.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions.* B.G. Teubner, Stuttgart, 1 edition, 1987.

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'