Time-Space Tradeoffs for Branching Programs

Paul Beame*
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA
beame@cs.washington.edu

Michael Saks†
Dept. of Mathematics
Rutgers University
New Brunswick, NJ
saks@math.rutgers.edu

Jayram S. Thathachar*
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA
jayram@cs.washington.edu

19 May 1999

Abstract

We obtain the first non-trivial time-space tradeoff lower bound for functions $f : \{0,1\}^n \to \{0,1\}$ on general branching programs by exhibiting a Boolean function f that requires exponential size to be computed by any branching program of length $(1+\epsilon)n$, for some constant $\epsilon > 0$. We also give the first separation result between the syntactic and semantic read-k models [BRS93] for $k > 1$ by showing that polynomial-size semantic read-twice branching programs can compute functions that require exponential size on any syntactic read-k branching program. We also show a time-space tradeoff result on the more general R-way branching program model [BRS93]: for any k, we give a function that requires exponential size to be computed by length kn q-way branching programs, for some $q = q(k)$.

1 Introduction

One of the long-standing open questions of complexity theory is whether polynomial-time is the same as log-space. One approach to this problem has been to look at tradeoffs between time and space for natural problems in P. For example, does the addition of a restriction on the space allowed prevent one from solving problems in P within specific polynomial time

* Research supported by NSF grant CCR-9303017.
† Research supported by NSF grant CCR-9700239. This work was done while on sabbatical at University of Washington.

bounds? Despite significant progress given by Fortnow's recent time-space tradeoff lower bounds for SAT [For97], this question remains unsolved.

One natural model for studying this question is that of Boolean branching programs, which simultaneously capture time and space in a clean combinatorial manner. In this model, a program for computing a function $f(x_1, x_2, \ldots, x_n)$ is represented as a DAG with a unique start node. Each non-sink node is labeled by a variable, and the arcs out of a node correspond to the possible values of the variable. The sink nodes are each labeled by an output value. Executing the program on a given input corresponds to following a path from the start node using the values of the input variables to determine the arcs to follow. The maximum length of a path corresponds to time and the logarithm of the number of nodes corresponds to space. An algorithm running simultaneously in linear time and logarithmic space corresponds to a linear-length, polynomial-size branching program. Thus the question of finding explicit functions in P for which no such branching program exists has been of significant research interest.

This paper gives results on two distinct problems for branching programs, which we summarize in the next two subsections.

1.1 Lower bounds for single-output functions

There has been a great deal of success in proving time-space tradeoff lower bounds for *multi-output* functions in FP such as sorting, pattern matching, matrix-vector product, and hashing [BC82, Bea91, Abr90, Abr91, MNT93]. However, for single-output functions (those whose output is one bit), prior to the work in this paper, there were apparently no lower bounds known better than $n + C$ (for *constant* C) for any explicit n variable function. The existing techniques for multi-variate functions involve some sort of "progress measure" which quantifies how much of the output has been produced. These techniques do not seem to give any non-trivial bounds for functions with a single output bit. For example, it is not known how to relate the apparently very similar problems of sorting and element distinctness, although time-space tradeoffs for element distinctness on the structured *comparison branching program* have been shown [BFMadH$^+$87, Yao88].

The branching program model allows the domain of the variables to be any finite set. For variables taking values in a q element set, the nodes in the program have out-degree q, corresponding to the possible values. While the case of greatest interest is the case that variables are 2-valued, the general q-valued case is an interesting challenge, which can potentially provide insights into the 2-valued case. Our first result is to exhibit, for each odd prime power q, an explicit family of functions $\mathcal{F}_q = (f_n^{(q)} : n$ a positive power of 2) such that $f_n^{(q)}$ is a function from $GF(q)^n$ to $\{0, 1\}$, and for any k there is a $q(k)$ such that branching programs of depth kn for functions in family $\mathcal{F}_q$ require size 2^S where $S = \Omega(n \log q)$. In particular, if one chooses q to be a function of n, say $q = \Theta(n)$ then our bounds show that any branching program of length $o(n \log \log n)$ require size at least 2^S where $S = n \log^{1-\epsilon} n$ for any $\epsilon > 0$.

This result seems strong on its face, but is unsatisfying because of the dependence on q. For each k, the q required for the bound can be quite large. For q-valued variables, the number of bits that would be needed to represent the input is $n \log_2 q$ and the length bound is smaller than this quantity. What we really want is a lower bound that is super-linear in the number of input bits.

Our second lower bound pertains to the "real" model, single output functions on 2-valued variables, i.e., Boolean functions. Previously, the only size-depth tradeoffs for explicit functions were results in [KW88] that showed that for branching programs of depth exactly n, the size must be exponential. For this model, we obtain the first non-trivial length lower bound for polynomial size branching programs for functions whose output is a single bit: we exhibit an explicit family of functions in P and show that any sub-exponential size program for it must have length at least $1.0178n$. While this is only just barely non-trivial, it is the first such result in which the length divided by the number of variables is bounded away from one. [1]

The proofs introduce some techniques, some of which extend past techniques of [BRS93, Tha98]. First, we show that if a function f has a small size and length branching program, then it is possible to write f in the form $\bigvee_i \bigwedge_j T_j^i$ where each function T_j^i is a decision tree of "small" height, and where the number of terms in the expression is not too large. (This generalizes a similar construction for the special cases of *syntactic read-k* and *oblivious* branching programs, in [BRS93]). So proving a size lower bound trade-off for branching programs for f reduces to showing that no such representation exists.

We then analyze these decision programs and show that the sets of inputs they accept can be decomposed into *pseudo-rectangles*, which are objects related to, but more general than, the rectangles that are routinely analyzed in communication complexity. The analysis that such pseudo-rectangles do not work well at approximating the explicit functions we define are generalizations and improvements of arguments in [BRS93, Tha98].

The key to showing the result for the Boolean domain is to prove a tighter form of this decomposition. The proof of this tighter form employs an interesting entropy argument.

The explicit functions for which we prove our lower bounds in both the q-way and Boolean branching program models are based on quadratic forms $x^T M x$ where the $n \times n$ matrix M is a (possibly slightly modified) Sylvester or Generalized Fourier Transform matrix.

1.2 Semantic versus syntactic read-k branching programs

As a step towards proving super-polynomial size lower bounds for linear length branching programs, a natural restriction is to require that each input bit be read at most some fixed number of times. This led to the definition of read-k branching programs [Weg87] in which each input can be read at most k times. Many lower bounds have been shown for several functions on read-once branching programs (for example, see [Weg88, Raz91, SS93]).

(Another branching program restriction that has also been considered is that of *oblivious* branching programs which test the same variable at each time-step along any path. For oblivious branching programs, linear length and read-k for some constant k are essentially the same and several size-length tradeoff lower bounds for oblivious branching programs have been

[1] Since the publication of a preliminary version of our results, Ajtai [Ajt99a] (see also [Ajt99b], using related techniques, has exhibited an explicit family of boolean functions for which any linear size branching program must have exponential size.

shown using this connection [AM88, BNS92]. Oblivious read-once branching programs, known as OBDD's, have been very useful as representations of functions used in verification [Bry86, BCL$^+$94] and so have generated significant independent interest.)

Borodin, Razborov, and Smolensky [BRS93] observed that read-k branching programs come in two flavors, *syntactic* read-k in which all paths in the branching program must satisfy the read-k restriction and the more general *semantic* read-k in which only the paths consistent with some input must satisfy the restriction. They also proved strong size lower bounds for the syntactic read-k model. However, obtaining super-polynomial size lower bounds even for semantic read-twice branching programs has been an open question.[2] (It is easy to observe that there is no distinction between syntactic read-once and semantic read-once branching programs.)

Here, we show the first separation between the syntactic and semantic read-k models for $k > 1$ by showing that polynomial-size semantic read-twice branching programs can compute functions that require exponential size for any syntactic read-k branching programs. The functions we construct are based on a class of functions that were by introduced by Thathachar [Tha98] to separate the power of read-k and read-(k+1) in the syntactic model. These functions are exponentially hard for syntactic read-k, and seem to be hard for semantic read-k. We modify these functions so as to make them semantic read-twice while still retaining hardness for syntactic read-k.

2 Notation

2.1 Boolean functions

Throughout, X is a set of variables (usually $\{x_1, \ldots, x_n\}$) taking values from some finite set D. We say X is a D-valued variable set, or d-valued if $d = |D|$. An *input* σ is a point in D^X, the set of mappings from X to D, and we identify this set, in the usual way, with D^n. If $Y \subseteq X$, a point of D^Y is a *partial input* to X. If ρ is a partial input, we write $vars(\rho)$ for the set of variables that are assigned by ρ. The *size* of a partial input ρ is $|vars(\rho)|$.

A *Boolean function* f over X is a function mapping D^n to $\{0,1\}$. Given a partial input ρ, the *restriction* of f by ρ, $f\lceil_\rho$ is the function on $X' = X \setminus vars(\rho)$ such that for $\sigma \in D^{X'}$, $f\lceil_\rho(\sigma) = f(\sigma, \rho)$. For $b \in \{0,1\}$, a *b-certificate* for f is a partial input ρ such that $f\lceil_\rho$ is the constant function with value b.

Let f and g be boolean functions on D^n. We say that g is a *portion* of f if $g^{-1}(1) \subseteq f^{-1}(1)$.

If $A_1, A_2, \ldots, A_t$ are subsets of X we say that f is $(A_1, A_2, \ldots, A_t)$-*free* if f can be written as $\bigwedge_{i=1}^t g_i$, and each g_i does not depend on the variables in A_i. We say that f is an *ℓ-pseudo-rectangle* if there exist *disjoint* sets A_1, A_2 in X of size ℓ such that f is (A_1, A_2)-free, i.e. there are g_1 and g_2 such that $f = g_1 \wedge g_2$ and g_1 and g_2 are independent of disjoint sets of ℓ variables. We call ℓ the *order* of the pseudo-rectangle.

The name pseudo-rectangle is motivated by the following. A function f on variable set Y is a *rectangle* with respect to the partition Y_1, Y_2 of Y if $f^{-1}(1)$ can be written in the form

[2] The new results of [Ajt99a] mentioned earlier yield such bounds.

$\Gamma_1 \times \Gamma_2$ where Γ_i is a set of assignments to Y_i. A function f is an pseudo-rectangle with associated sets A_1, A_2, if and only if for each partial input ρ to $X - A_1 - A_2$, the restriction $f\lceil_\rho$ is a rectangle with respect to the partition A_1, A_2 of the remaining variables.

2.2 Branching programs, decision trees, and decision programs

Since we are only interested in the computation of Boolean (single output) functions here, we present our definitions of branching programs only for this case. A *non-deterministic branching program* B on a D-valued variable set X is an acyclic directed graph with the following properties:

- There is a unique source node, denoted $start_B$.
- Every sink node v has a label $output(v)$, which is 0 or 1.
- Each non-sink node v is labeled by a variable $x(v) \in X$
- Each arc a is labeled by an element $value(a)$ of D.

We say that an input σ is *consistent with an arc a*, where $a = (u, v)$, if the value given by σ to $x(u)$ is $value(a)$. We extend this definition to any path P: σ is consistent with P if it is consistent with each arc in P. A path P is *maximal* if it starts at $start_B$ and ends at a sink. A path that ends at a sink with output value 1, resp. 0, is a 1-path, resp. 0-path.

We say that B *accepts* the input σ if there is a maximal 1-path that is consistent with σ. We view B as a boolean function from D^n by defining $B(\sigma) = 1$ if and only if B accepts σ.

Two measures associated with B are *size* which equals the number of nodes, and *length* which is the length of the longest path.

(In [BRS93], a non-deterministic branching program also contains free arcs, namely arcs which are consistent with any input. Such branching programs can be modified to conform to our definition with at most a quadratic blow-up in size.)

A branching program is *deterministic* if there are exactly $|D|$ arcs out of each non-sink node, each with a different value. For a deterministic program, each input is consistent with exactly one maximal path. Intuitively, a deterministic program is "executed" on input σ by starting at $start_B$, reading the variable $x(start_B)$ and following the unique arc labeled by $\sigma(x(start_B))$. This process is continued until a sink is reached and the output of the computation is the output value of the sink. The non-deterministic version can be similarly viewed as a process, where from each node, v, one can choose from among the arcs labeled $\sigma(x(v))$ (if any); the output of the function is 1 if and only if some sequence of allowable choices leads to a sink node with output value 1.

A branching program of length d is *leveled* if the nodes can be partitioned into d sets $V_0, V_1, \ldots, V_d$ where V_0 is the source, V_d is the set of sink nodes and every arc out of V_i goes to V_{i+1}, for $0 \le i < d$. It is well known[Pip79] that every branching program P of size s and length d, can be converted into a leveled branching program P' of length d that has at most s nodes in each of its levels and computes the same function as P (and is deterministic if P is).

A branching program is *oblivious* if it is leveled and for each level, all of the variables on the level are labeled by the same variable.

A *decision tree* is a branching program B whose underlying graph is a tree rooted at $start_B$. In particular, a decision tree is leveled. Every function on n variables is computable by a deterministic decision tree of length n. Following common practice, the length of a decision tree is referred to as its *height*.

A *decision program* is an $\bigwedge$ of decision trees. More precisely, a (r, ϵ) decision program P over D is a collection $T_1, \ldots, T_r$ of decision trees on D^n such that each tree has height at most $\lceil \epsilon n \rceil$. The function computed by P is $\wedge_{i=1}^{r} T_i$. A decision program is deterministic if each of its component trees is, and is oblivious if each of its component trees is oblivious.

3 Main Decomposition Theorems

The general approach that we take towards proving size-length tradeoff lower bounds is to show that if f can be computed by a branching program of small length and size then f (or a large portion of f) can be expressed as an $\bigvee$ of not too many "simple" functions f_i. In our main theorems, this notion of "simple" will mean that each f_i is a βn-pseudo-rectangle for some appropriate parameter β. We obtain branching program lower bounds for a particular function f by showing that no large portion of f can be so expressed. This implies that there must be many functions f_i and thus the lower bounds on length and size will follow.

Theorem 1. *Let $k, n, s \in \mathbf{N}$ with $n \geq k(k+1)^2 2^{2(k+4)}$. Let f be a boolean function on D^n for some finite set D. If f can be computed by a (non-deterministic) branching program of length kn and size s then f may be written as:*

$$f = \bigvee_{i=1}^{m} f_i,$$

where each f_i is an βn-pseudo-rectangle and $m \leq 2^{4(k+1)\beta n}(2s)^{k(k+1)2^{k+4}}$ for $\beta = 1/2^{k+2}$.

This theorem is quite general but, as we will see, the bound on m is fairly large, and so this will only yield a non-trivial size-length tradeoff when the size of the domain D is sufficiently large when compared to k. To obtain our lower bound for functions on $\{0,1\}^n$ we prove a more specialized decomposition theorem that applies only to deterministic branching programs of length at most $(1+\epsilon)n$. In order to state this decomposition theorem it will be convenient to make the following technical definition. For $\epsilon, \delta \in (0, 1/4)$ define:

$$b(\epsilon, \delta) = \epsilon + \delta + (1+\epsilon)H\left(\frac{2(\epsilon+\delta)}{1+\epsilon}\right) + 2H(\delta),$$

where $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$ is the binary entropy function. The function $b(\epsilon, \delta)$ will arise as the result of a calculation in the proof of the next theorem. The key properties for our purposes are that if $\epsilon, \delta > 0$ with $\epsilon + \delta \leq 1/4$ then $b(\epsilon, \delta)$ increases with either ϵ or δ and tends to 0 if both ϵ and δ tend to 0.

Theorem 2. *Let $\epsilon > 0$, and suppose that f is a boolean function on $\{0,1\}^n$, for sufficiently large n, that can be computed by a branching program of length $(1+\epsilon)n$ and size s. Let the size of the smallest 1-certificate of f be at least $(1-\delta)n$. If $\delta+\epsilon \leq 1/4$ then, for any $\gamma \in (0,1)$, we can express f as:*

$$f = f_0 \vee (\bigvee_{i=1}^{m} f_i),$$

where $m \leq s\,2^{\gamma n}$, $|f_0| \leq s\,2^{(1+b(\epsilon,\delta)/2-\gamma/2)n}$, and, for $i \geq 1$, each f_i is a $(1-\epsilon-\delta)n/2$-pseudo-rectangle.

In the remainder of this section we outline the overall strategy for the proofs of these theorems in the deterministic case.

In order to obtain these decompositions of (portions of) f into pseudo-rectangles of large order, we first express f as an $\bigvee$ of not too many functions P_j, each of which is "a little bit simple", in that it can be computed by an (r,ϵ) decision program for appropriate parameters r and $\epsilon < 1$. This decomposition, which we prove in section 5, has the feature that the only place where the size bound enters is in the number of functions P_j.

The more interesting parts of our arguments are those showing that each P_j (or a large portion thereof) can be written as an $\bigvee$ of not too many pseudo-rectangles. In the proof of Theorem 1 this is shown as follows. We choose the value of r so that on any input x, each constituent decision tree of the (r,ϵ) program reads only a small fraction of the bits of x. We find a pair of large sets of variables, $A_1(x)$ and $A_2(x)$, so that every constituent decision tree does not read any variables in at least one of $A_1(x)$ or $A_2(x)$ on input x. Theorem 1 then follows by counting the number of possible choices of the $A_i(x)$ and the way they partition the set of trees.

In the proof of Theorem 2, we choose $r=2$, so each decision tree T_1, T_2 examines only a little more than half the variables on any input. We thus obtain, more directly, a pair, $S_1(x)$ and $S_2(x)$, of sets of variables that are not read by the corresponding decision trees on input x. Using this, and the lack of coordination possible between the two trees when the function only has large 1-certificates, we show that all but a small fraction of inputs x are associated with a small number of pairs of sets.

In the next section, we show how these decompositions may be combined with various properties of functions to derive our size-length lower bound tradeoffs and give bounds for certain explicit functions. In the following four sections we give all the details necessary for the obtaining these lower bounds. We prove the decomposition into decision programs in section 5; give the detailed proofs of Theorem 1 and 2 in sections 6 and 7. In section 8, we give the detailed proofs that the explicit functions we define have the desired properties.

4 Size-Length Tradeoff Lower Bounds

4.1 General Bounds

We now show how to convert the decomposition theorems of the previous section into lower bound tradeoffs for branching program computation of specific functions. Since our decompositions represent portions of f as βn-pseudo-rectangles, we make the following definitions:

Given a Boolean function f on domain D^n and $\beta \in [0,1]$,
define

- $\eta(f)$ be so that $\left|f^{-1}(1)\right| = |D|^{(1-\eta(f))n}$
- $\Psi_f(\beta)$ be the largest γ such that every βn-pseudo-rectangle g that is a portion of f has $\left|g^{-1}(1)\right| \leq |D|^{(1-\gamma)n}$.

If m is the parameter given by Theorem 1, any f must have a portion of size at least $\left|f^{-1}(1)\right|/m$ that is an $n/2^{k+2}$-pseudo-rectangle, which implies that $\left|f^{-1}(1)\right| |D|^{1-\Psi_g(1/2^{k+2})} \leq m$. Combining this with the upper bound on m given by Theorem 1, we obtain the following lower bound on the size s of any depth kn branching program to compute f:

Corollary 3. *Let $n, k \in \mathbf{N}$ with $n \geq (k+1)2^{k+4}$. Let f be a boolean function on D^n for some finite set D. Then any (nondeterministic) length kn branching program that computes f has size at least:*

$$\frac{1}{2}\left(\frac{|D|^{\Psi_f(\beta)-\eta(f)}}{2^{4(k+1)\beta}}\right)^{n/(k(k+1)2^{k+4})},$$

where $\beta = 1/2^{k+2}$.

We will apply this result for functions f for which the expression $\Psi_f(\beta) - \eta(f)$ is bounded below by some $\epsilon > 0$. In such a case, if $|D|$ is large enough (depending only on k) then $|D|^\epsilon$ is at least the square of the denominator $2^{4(k+1)\beta}$, and the above expression is at least $|D|^{\epsilon' n}$ for some $\epsilon' > 0$ depending only on k.

Remark: Note that if all 0-certificates of f are of size at least $t = (1-\delta)n$, then $\eta(f) \leq \delta + 1/n$, since every assignment to the first $t-1$ variables can be extended to an element of $f^{-1}(1)$. Thus we can replace the exponent $\Psi_f(\beta) - \eta(f)$ by $\Psi_f(\beta) - \delta - 1/n$ in the above corollary.

Theorem 1 requires $|D|$ to be sufficiently large (as a function of k) to yield a non-trivial lower bound. This is because one can construct a βn-pseudo-rectangle that is a portion of f and is 1 on at least a $|D|^{-\beta n}$ fraction of $f^{-1}(1)$: Among the elements of $f^{-1}(1)$, choose the most popular assignment ρ to the first βn variables $Y \subset X$. The pseudo-rectangle is $g_1 \wedge g_2$ where g_1 accepts input σ iff σ agrees with ρ on Y and g_2 accepts input σ iff the portion of σ on $X \setminus Y$, when extended by ρ, is in $f^{-1}(1)$. Thus $\Psi_f(\beta) \leq \beta + \eta(f)$ and so Theorem 1 requires $|D| > 2^{4(k+1)}$ in order to yield a non-trivial lower bound.

Now we consider the conditions needed for the Boolean case. Suppose we have a function $f : \{0,1\}^n \to \{0,1\}$ with all 1-certificates of size at least $(1-\delta)n$ computed by a branching program of depth $(1+\epsilon)n$. For any arbitrary γ, we can find $f_0, \dots, f_m$ as in Theorem 2. For each $i \geq 1$, f_i is a βn-pseudo-rectangle for $\beta = (1-\epsilon-\delta)/2$. By the definition of $\Psi_f(\beta)$, for each $i \geq 1$, $\left|f_i^{-1}(1)\right| \leq 2^{(1-\Psi_f(\beta))n}$, so, using the bounds of Theorem 2 on m and $\left|f^{-1}(1)\right|$, we have

$$\left|f^{-1}(1)\right| \leq s2^{(\gamma+1-\Psi_f(\beta))n} + s2^{(1+b(\epsilon,\delta)/2-\gamma/2)n}$$

Choosing γ to make the two summands equal, and rewriting the inequality to lower bound s yields:

Corollary 4. *Let $\epsilon > 0$, and suppose that f is a boolean function on $\{0,1\}^n$, for sufficiently large n, that can be computed by a branching program of length $(1+\epsilon)n$ and size s. Let the smallest 1-certificate of f be of size at least $(1-\delta)n$. If $\delta+\epsilon \leq 1/4$ then*

$$s \geq \frac{1}{2}2^{\alpha n},$$

where $\alpha = \frac{1}{3}\Psi_f((1-\epsilon-\delta)/2)) - \frac{1}{3}b(\epsilon,\delta) - \eta(f)$.

If the value α is a positive constant independent of n we get a strong lower bound. This will happen, for example, if f is chosen to be a function for which $\Psi_f(\frac{1}{4})$ can be bounded away from 0 (independent of n) and for which both the smallest 1-certificate and $\left|f^{-1}(1)\right|$ are large and if ϵ is chosen to be small enough. By the remark after the previous corollary it suffices to have both the smallest 0-certificates and 1-certificates large.

4.2 Tradeoffs for explicit functions

We now describe families of explicit functions for which the lower bounds of Corollaries 3 and 4 yield exponential size lower bounds on branching programs with certain length upper bounds. The functions are based on quadratic forms. (Similar functions based on bilinear forms were considered in [BRS93].) Let $M = M_n$ be an $n \times n$ matrix over $GF(q)$. Define the function $QF_M : GF(q)^n \rightarrow \{0,1\}$ to be true if and only if for any input σ (viewed as a vector of length n), $\sigma^T M \sigma = 0 \pmod{q}$. We define the function BQF_M to be the restriction of QF_M to the domain $\{0,1\}^n$. We consider these functions for the class of *Sylvester matrices*. For any odd prime power q and any $n = 2^k$, the $n \times n$ Sylvester matrix N is defined over $GF(q)$ and has rows and columns indexed by binary vectors of length k. The (i,j)-th entry of N is $(-1)^{\langle i,j \rangle}$, where $\langle i,j \rangle$ denotes the inner product of i and j. We also consider the *modified Sylvester matrix*, $N^{[0]}$, obtained by setting the diagonal entries of a Sylvester matrix N to 0.

The following lemma, which we prove in Section 8, provides bounds on the quantities needed to apply Corollaries 3 and 4.

Lemma 5. *Let M be a GFT matrix over $GF(q)$, where q is an odd prime power.*

1. *Let D be a subset of $GF(q)$ of size at least 2. If f is the restriction of either QF_M and $QF_{M^{[0]}}$ to the domain D^n, then $\Psi_f(\delta) \geq \delta^2$.*

2. *If $f = QF_{M^{[0]}}$ then every 0-certificate and every 1-certificate of f has size at least $n-1$.*

3. *If M is the Sylvester matrix over $GF(3)$, and $f = BQF_M$ then every 0-certificate and every 1-certificate of f has size greater than $n - 24\sqrt{n}\log n$.*

Using Corollary 4 we obtain:

Theorem 6. *Any deterministic branching program of length $1.0178n$ computing BQF_N, where N is the $n \times n$ Sylvester matrix over $GF(3)$, requires size $2^{\Omega(n)}$.*

Proof. Fix n large enough and let N be the $n \times n$ Sylvester matrix over $GF(3)$, and let $f = BQF_N$. Applying Corollary 4, we want to show that for the given $\epsilon = .0178n$, that $\alpha > 0$. Every 0-certificate and 1-certificate of f has size at least $(1-\delta)n$ where δ is $o(1)$ in n and, using the remark following Corollary 3, $\eta(f) \leq \frac{1}{n} + \delta$ which is also $o(1)$ in n. Therefore by the lower bound $\Psi_f(\beta) \geq \beta^2$, we get: $\alpha = \frac{1}{3}\left(\frac{(1-\epsilon)^2}{4} - \epsilon - (1+\epsilon)H(\frac{2\epsilon}{1+\epsilon})\right) + o(1)$. It can be checked that for $\epsilon \leq 0.0178$, the expression upper bounding α is strictly positive. Hence, any branching program of length at most $1.0178n$ that computes BQF_N must have exponential size. □

In the case of large D, we use Corollary 3 to obtain:

Theorem 7. *There are constants $c, \gamma > 0$ such that for every positive integer k, and every prime power q with $\log\log q \geq ck$, and all sufficiently large n the following holds: Let $N^{[0]}$ be an $n \times n$ modified Sylvester matrix over $GF(q)$. Then any length kn non-deterministic branching program for $QF_{N^{[0]}}$ requires size at least $q^{\gamma n}$.*

Proof. Let $f = QF_{N^0}$ for some q and n. Then f is a function on D^n where $|D| = q$. Let $\beta = 1/2^{k+2}$. By Lemma 5 and the remark following Corollary 3, $\Psi_f(\beta) - \eta(f) \geq \beta^2 - \frac{2}{n}$. Choosing q so that $\log\log q \geq ck$ for some sufficiently large constant c, we have that $q^{\beta/2} > 2^{8(k+1)}$ and so $q^{\Psi_f(\beta)-\eta(f)}/2^{4(k+1)\beta} \geq q^{\beta^2/2}$. Applying Corollary 3 now finishes the proof. □

Also, we note that in Corollary 3 the parameters k, n, and $q = |D|$ may all be set independently. In particular, this implies that we can set k and q in ways that depend on n. Choosing q to be the smallest prime $\geq n$ and $k = \frac{1}{c}\log_2\log_2 n$, by the same reasoning as above, we derive the following theorem.

Theorem 8. *Let N be an $n \times n$ Sylvester matrix and consider the associated function $QR_{N^{[0]}}$ defined on a domain D of size q, $n \leq q < 2n$. For any $\epsilon > 0$ there is a constant $c' > 0$ such that any (nondeterministic) branching program of length at most $c'n\log\log n$ requires size at least 2^S to compute $QR_{N^{[0]}}$, where $S = \Omega(n\log^{1-\epsilon} n)$.*

The conclusions of Theorems 7 and 8 hold, more generally, whenever N is a Generalized Fourier Transform (GFT) matrix (see [BRS93]). For any finite Abelian group G, let G^* be the set of multiplicative characters of G mapping elements of G to $GF(q)^*$, that is, $\chi(g_1g_2) = \chi(g_1)\chi(g_2)$ for any $g_1, g_2 \in G$ and $\chi \in G^*$. Provided that q is relatively prime to $|G|$ (see, e.g., [BRS93]), it is known that there are $|G|$ distinct characters and that they are linearly independent when viewed as a vector space over $GF(q)$. Let $N = N_{G,G^*}$ be the matrix in which the $(g,\chi)^{\text{th}}$ element equals $\chi(g)$, for all $g \in G$ and $\chi \in G^*$. Sylvester matrices of dimension $n \times n$, where $n = 2^k$ for some k, can be shown to be special cases of GFT matrices corresponding to the additive group of $GF(2)^k$.

5 Decomposition into Decision Programs

Lemma 9. *Let $k \in \mathbf{R}$ and $n, s \in \mathbf{N}$. Let f be a boolean function on D^n for some finite set D. If f can be computed by a branching program of length kn and size s then for any integer $r \geq k$, f can be expressed as:*

$$f = \bigvee_{i=1}^{u} P_i,$$

where $u \leq s^{r-1}$ and each P_i is a $(r, \frac{k}{r})$-decision program.

Furthermore, (i) if the branching program is deterministic then each of the decision programs P_i is deterministic and the sets $P_i^{-1}(1)$ are pairwise disjoint, and (ii) if the branching program is oblivious then each of the decision programs is oblivious.

Proof. Let B be any branching program of size s computing f of length $d \leq kn$. Let B' be an equivalent leveled program of length d with at most s nodes per level. For distinct nodes v and w of the branching program, let $f_{v,w}$ denote the function on D^n which is 1 on input σ if, starting from v, there is a path consistent with σ that leads to w. It is easy to see that if v is at level i and w is level $j > i$, then $f_{v,w}$ can be computed by a decision tree of height $j - i$. Furthermore each such decision tree is deterministic and/or oblivious if B is. For $1 \leq i \leq r-1$, define $l_i = \lceil \frac{id}{r} \rceil$. Note that $l_1 < \cdots < l_{r-1} < d$ divides the interval $[0, d]$ into r intervals each of size at most $\lceil \frac{d}{r} \rceil \leq \lceil \frac{kn}{r} \rceil$. An input is accepted by P if and only there is a sequence of nodes $v_0, v_1, v_2, \ldots, v_{r-1}, v_r$, where v_0 is the start node, v_r is the accepting node and for $i \in [r-1]$, v_i is at level l_i, such that $f_{v_{i-1},v_i}(\sigma) = 1$ for each $i \in [r]$. Therefore

$$f = \bigvee_{v_1, \ldots v_{r-1}} \bigwedge_{i=0}^{r-1} f_{v_i, v_{i+1}}.$$

There are at most s^{r-1} terms in the $\bigvee$, and each term is a $(r, \frac{k}{r})$ decision program.

Finally, note that, in the deterministic case, each input follows a unique path, and so is accepted by at most one of the decision programs.

□

The oblivious version of this lemma was implicit in [BRS93] which gave a similar oblivious decomposition in the case of syntactic read-k branching programs. (For the definition of syntactic read-k, see section 9.)

6 Tradeoffs over Large Domains and for Oblivious Branching Programs

In this section we prove the decomposition given by Theorem 1. The decomposition of f into (r, ϵ)-decision programs of the previous section can chosen so that each of the r component decision trees is only able to examine a small portion of the variables on any input. To prove Theorem 1, we will need to decompose each decision program into collections of pseudo-rectangles of large order which, by definition, is of the form $g_1 \wedge g_2$ where g_1 and g_2 do not depend on fairly large disjoint sets of variables A_1 and A_2 respectively. To do this we will use the following combinatorial lemma which allows us to find sufficiently large sets A_1 and A_2 so that each decision tree in the decision program does not examine either all of A_1 or all of A_2. This lemma is a generalization of a lemma from [Tha98].

6.1 A combinatorial lemma

Lemma 10. *Let k, p, n be positive integers with $p = (k+1)2^{k+4}$ and let $n \geq kp^2$. Let $\beta = 1/2^{k+2}$ and $r = kp$. Suppose that $S_1, \ldots, S_r$ are subsets of $[n]$, each of size at least $\lfloor n(1-\frac{1}{p})\rfloor$. Then there is a partition (I_1, I_2) of $[r]$ and two sets A_1, A_2, each of size at least βn such that $A_1 \subseteq \bigcap_{i \in I_1} S_i$ and $A_2 \subseteq \bigcap_{i \in I_2} S_i$.*

We derive this lemma as a corollary of the following:

Lemma 11. *Let $\mathcal{F}$ denote a family of non-empty sets each containing at most m elements. Let $X = \bigcup_{Y \in \mathcal{F}} Y$ and suppose $|X| = n$. Let $\lambda = \sum_{Y \in \mathcal{F}} |Y|/n$. Then, X has disjoint sets S and T, each of size at least $(1-\delta)2^{-\lambda}n$, such that each $Y \in \mathcal{F}$ is disjoint either from S or from T, where*

$$\delta = \sqrt{\frac{\lambda m 2^{1+\lambda}}{n}}.$$

The quantity λ can be interpreted naturally as the average number of times each element in X is covered. In [Tha98], a weaker lemma was proved that required each x to appear in at most $m|\mathcal{F}|/n$ sets.

Using Lemma 11, we prove Lemma 10 as follows. If $|\bigcap_i S_i| \geq 2\lceil \beta n \rceil$, we can choose A_1 and A_2 to be any even (or nearly even) partition of $\bigcap_i S_i$ and I_1 and I_2 be be any partition of $[r]$. Otherwise, let $S_i' = X \backslash S_i$, and define the family $\mathcal{F}$ to consist of S_i' for $1 \leq i \leq kp$, and singleton sets $\{j\}$ for each $j \notin \bigcup_i S_i'$ so that the total union of sets in $\mathcal{F}$ is $[1, n]$. The number of singleton sets is at most $2\lceil \beta n \rceil - 1$. Therefore, the sum of the sizes of the sets in $\mathcal{F}$ is at most $2\lceil \beta n \rceil - 1 + (kp-1)\lceil n/p \rceil \leq (k+2\beta)n + kp - n/p \leq (k+2\beta)n$ by our bound on n. In the terminology of Lemma 11, we have $\lambda \leq (k+2\beta)n/n = k+2\beta \leq k+1$. Moreover, if we let $m = \lceil n/p \rceil$, it follows that $\lambda m \leq (k+2\beta)(n/p+1) \leq (k+1)n/p$. Applying Lemma 11 with λ and m, there exists a pair of disjoint sets A_1 and A_2 each of size at least $(1-\delta)n/2^{k+1}$, where

$$\delta = \sqrt{\frac{\lambda m 2^{1+\lambda}}{n}} \leq \sqrt{\frac{(k+1)n2^{k+2}}{np}} = \sqrt{\frac{(k+1)2^{k+2}}{(k+1)2^{k+4}}} = \frac{1}{2},$$

such that each S_i' is disjoint from either A_1 or A_2. In other words, there exist sets A_1 and A_2 each of size at least βn, and a partition of $[1, kp]$ into I_1 and I_2 such that S_i contains A_1 if $i \in I_1$, and S_i contains A_2 if $i \in I_2$. This proves Lemma 10.

To prove Lemma 11, we use two forms of Chebyschev's inequality – the usual well-known form, and a second, less standard, but equivalent, form that is also known as Chebyschev's inequality (see for example [HLP52, Theorem 43, page 43]).

Proposition 12. *Let* $\mathrm{E}[\mathbf{Z}_x]$ *denote the mean and* $\mathrm{var}[\mathbf{Z}_x]$ *denote the variance of a random variable* $\mathbf{Z}_x$*. Then, for any* $\delta > 0$*,*

$$\Pr[|\mathbf{Z} - E[\mathbf{Z}]| > \delta \cdot E[\mathbf{Z}]] < \frac{\mathrm{var}[\mathbf{Z}]}{\delta^2 \cdot \mathrm{E}[\mathbf{Z}]^2}$$

Proposition 13. *Let* $a_1, \ldots a_N$ *is a non-decreasing sequence and* $b_1, \ldots b_N$ *be a non-increasing sequence of non-negative numbers. Then,* $\sum_i a_i b_i \leq (\sum_i a_i)(\sum_i b_i)/N$*.*

Proof of Lemma 11. The proof uses some of the ideas in [Tha98]; we give the complete proof to make it self-contained. Color each set $Y \in \mathcal{F}$ red or blue uniformly and independently. Define S (respectively, T) to be the set of elements x such that every set that contains x is colored red (respectively, blue). Since every element of X occurs in at least one set, it follows that S and T are disjoint. Moreover, for each $Y \in \mathcal{F}$, either $Y \cap S$ or $Y \cap T$ is empty. To complete the proof, we show that with positive probability both S and T have at least $(1-\delta)2^{-\lambda} n$ elements.

For $x \in X$, let d_x denote the number of sets that contain x. We have the elementary equality $\sum_{x \in X} d_x = \sum_{Y \in \mathcal{F}} |Y| = \lambda n$. Let $\mathbf{Z}_x$ be the 0-1 indicator random variable for the event "$x \in S$". By the definition of S, this event occurs with probability 2^{-d_x}, implying that $\mathrm{E}[\mathbf{Z}_x] = \Pr[\mathbf{Z}_x = 1] = 2^{-d_x}$. Let $\mathbf{Z} = \sum_x \mathbf{Z}_x$ and observe that $\mathbf{Z} = |S|$. Using the arithmetic-geometric mean inequality, we obtain

$$\mathrm{E}[\mathbf{Z}] = \sum_x \mathrm{E}[\mathbf{Z}_x] = \sum_x 2^{-d_x} \geq 2^{-\sum_x d_x/n} n = 2^{-\lambda} n. \tag{1}$$

We show below that $\mathbf{Z}$ is close to its expected value with high probability.

The variance of $\mathbf{Z}$ is given by

$$\mathrm{var}[\mathbf{Z}] = \sum_x \mathrm{var}[\mathbf{Z}_x] + \sum_{x \neq y} \mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y), \tag{2}$$

where $\mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y) = \mathrm{E}[\mathbf{Z}_x \mathbf{Z}_y] - \mathrm{E}[\mathbf{Z}_x]\mathrm{E}[\mathbf{Z}_y]$ denotes the covariance of $\mathbf{Z}_x$ and $\mathbf{Z}_y$ (see [AS92] for an elementary proof of this equality). Consider the first term in the right hand side of Equation 2. For any x, $\mathbf{Z}_x$ is a Bernoulli random variable, so $\mathrm{var}[\mathbf{Z}_x] = \mathrm{E}[\mathbf{Z}_x](1 - \mathrm{E}[\mathbf{Z}_x]) \leq \mathrm{E}[\mathbf{Z}_x]$, implying that

$$\sum_x \mathrm{var}[\mathbf{Z}_x] \leq \mathrm{E}[\mathbf{Z}] \tag{3}$$

To bound the second term in the right hand side of Equation 2, observe that if no set Y contains both x and y, then events $\mathbf{Z}_x$ and $\mathbf{Z}_y$ are independent implying that $\mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y) = 0$. Thus, we are only interested in those pairs (x, y) such that some set contains both x and y. For any fixed x, the number of such pairs (x, y) is at most $(m-1)d_x$. For each such pair, $\mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y) \leq \mathrm{E}[\mathbf{Z}_x \mathbf{Z}_y] \leq \mathrm{E}[\mathbf{Z}_x] = 2^{-d_x}$. Therefore,

$$\sum_{x \neq y} \mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y) \leq (m-1) \sum_x d_x 2^{-d_x}.$$

The last term above can be bounded as follows: Order the x's so that the sequence $\{d_x\}$ is non-decreasing. Now the second Chebyschev's inequality (Proposition 13) can be applied to the sequences $\{d_x\}$ and $\{2^{-d_x}\}$. We obtain

$$\sum_{x \neq y} \mathrm{cov}(\mathbf{Z}_x, \mathbf{Z}_y) \leq \frac{(m-1)\left(\sum_x 2^{-d_x}\right)\left(\sum_x d_x\right)}{n} = \lambda(m-1)\mathrm{E}[\mathbf{Z}], \tag{4}$$

because $(\sum_x d_x)/n = \lambda$, and $\sum_x 2^{-d_x} = \mathrm{E}[\mathbf{Z}]$. Substitute the bounds in Equation 3 and Equation 4 into Equation 2. We obtain

$$\mathrm{var}[\mathbf{Z}] \leq (\lambda(m-1)+1)\mathrm{E}[\mathbf{Z}] \leq \lambda m \mathrm{E}[\mathbf{Z}],$$

where the last inequality holds because each $x \in X$ occurs in at least one set, implying that $\lambda = \sum_x d_x/n \geq 1$. Using the first Chebyschev's inequality (Proposition 12), we have

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

Substituting for δ as given by the statement of the lemma, and using Equation 1, we obtain

$$\Pr[\mathbf{Z} < (1-\delta) \cdot E[\mathbf{Z}]] < \frac{\lambda m n}{\lambda m 2^{1+\lambda} E[\mathbf{Z}]} = \frac{n}{2^{1+\lambda} E[\mathbf{Z}]} \leq \frac{n}{2^{1+\lambda} 2^{-\lambda} n} = \frac{1}{2}.$$

In a similar fashion, we obtain $\Pr[|T| < (1-\delta) \cdot E[\mathbf{Z}]] < 1/2$. Thus, with positive probability, both S and T have size at least $(1-\delta) \cdot E[\mathbf{Z}] \geq (1-\delta)2^{-\lambda} n$. We conclude that there is a coloring of the Y's such that the induced S and T satisfy the lemma. □

6.2 Oblivious branching programs

As a warm-up for the proof of Theorem 1, we show how Lemma 10 yields a time-space tradeoff for oblivious branching programs. Observe that, by definition, any *oblivious* (r, ϵ) decision program is the $\bigwedge$ of r functions T_i, each of which is associated to a set S_i of variables of size $(1-\epsilon)n$ on which it does not depend. Therefore, by Lemma 10, for $p = (k+1)2^{k+4}$ and $\beta = 1/2^{k+2}$, there are sets A_1, A_2 of size βn and a partition of $[r]$ into I_1, I_2 such that for $i = 1, 2$, $g_j = \bigwedge_{i \in I_j} T_i$ does not depend on A_j. Thus $g_1 \wedge g_2$ represents a $(kp, 1/p)$ decision program as a βn-pseudo-rectangle. So, by Lemma 9, we have:

Theorem 14. *Let $k, n, s \in \mathbf{N}$ with $n \geq k(k+1)^2 2^{2(k+4)}$. Let f be a boolean function on D^n on a finite set D. If f can be computed by an oblivious branching program of length kn and size s then f can be expressed as:*

$$f = \bigvee_{i=1}^{m} f_i,$$

where $m \leq s^{k(k+1)2^{k+4}-1}$ and each h_i is a βn-pseudo-rectangle for $\beta = 1/2^{k+2}$.

In the same way that Theorem 1 was used to deduce Corollary 3, we can use Theorem 14 to deduce the following lower bound result for oblivious branching programs. (Again, something similar is implicit in [BRS93]).

Corollary 15. *Let $n, k \in \mathbf{N}$ with $n \geq k(k+1)^2 2^{2(k+4)}$. Let f be a boolean function on D^n for some finite set D. Then any (nondeterministic) oblivious length kn branching program that computes f has size at least:*

$$|D|^{\frac{\Psi_f(\beta)-\eta(f)}{k(k+1)2^{k+4}} n},$$

where $\beta = 1/2^{k+2}$.

Note that, in contrast with the bound of Corollary 3, this gives exponential size lower bounds even for $|D| = 2$ in the case of oblivious branching programs.

6.3 Large domain decomposition

We now proceed to the proof of Theorem 1. Let k, n, s, D and f be as hypothesized in the theorem. As in the oblivious case, setting $p = (k+1)2^{k+4}$ and using Lemma 9, we can write $f = \bigvee_{i=1}^{m} P_i$ where $m \leq s^{kp-1}$ and each of the P_i is a $(kp, \frac{1}{p})$ decision program. When these are oblivious then they are automatically also pseudo-rectangles of large order. For arbitrary decision programs this need not be true; instead we show that each decision program can be written as an $\bigvee$ of not too many βn-pseudo-rectangles.

Lemma 16. *Let k, p, n be positive integers with $p = (k+1)2^{k+4}$, and $n \geq kp^2$. Let $\beta = 1/2^{k+2}$. Let $P = (T_1, \ldots, T_{kp})$ be a $(kp, \frac{1}{p})$ decision program. Then P can be written as:*

$$P = \bigvee_{i=1}^{t} h_i,$$

where $t \leq 2^{4\beta(k+1)n+kp}$, and each h_i is a βn-pseudo-rectangle.

Applying this lemma to each P_i in the representation of f, we can write f as a $\bigvee$ of at most $s^{kp-1} 2^{4\beta(k+1)} 2^{kp}$ function that are each βn-pseudo-rectangles, as required to prove the theorem. So it remains to prove the lemma.

Proof. Let $T_1, \dots T_{kp}$ be the component decision trees in P. For each input σ, let $S_i(\sigma)$ be the set of variables that are not read by T_i on input σ. Note that each set $S_i(\sigma)$ has size at least $n(1-\frac{1}{p})-1$. Consider all quadruples (A_1, A_2, I_1, I_2) where $|A_1| = |A_2| = \beta n$ are disjoint sets of variables and I_1, I_2 are complementary subsets of $\{1, 2, \dots, kp\}$. Call such a quadruple *eligible*. Note that the number of eligible quadruples is bounded above by $2^{kp}\binom{n}{\beta n}^2 \le 2^{kp}2^{2H(\beta)n}$, using the standard inequality for binomial coefficients given below in Proposition 24. Since $\beta \le 1/2$, $-\beta \log_2 \beta \ge -(1-\beta)\log_2(1-\beta)$ and so $H(\beta) \le -2\beta \log_2 \beta$. Thus the number of eligible quadruples is at most $2^{kp+4(k+1)\beta n}$.

We say that (A_1, A_2, I_1, I_2) *covers input* σ if for each $i \in I_1$, $A_1 \subseteq S_i(\sigma)$ and for each $i \in I_2$, $A_2 \subseteq S_i(\sigma)$. Observe that Lemma 10 implies that each σ is covered by some eligible quadruple (A_1, A_2, I_1, I_2).

For each pair (A, I) where A is a variable subset and $I \subseteq \{1, 2 \dots, kp\}$, define the function $g_{A,I}$ that accepts σ if and only if, for every $i \in I$, $A \subseteq S_i(\sigma)$ and T_i accepts σ. It is easy to see that $g_{A,I}$ does not depend on the variables in A. Also for any eligible quadruple (A_1, A_2, I_1, I_2), the βn-pseudo-rectangle $g_{A_1,A_2,I_1,I_2} = g_{A_1,I_1} \wedge g_{A_2,I_2}$ accepts exactly the set of 1's of f that are covered by the quadruple. Thus $f = \bigvee_{(A_1,A_2,I_1,I_2)} g_{A_1,A_2,I_1,I_2}$, and this representation satisfies the conclusion of the lemma. $\square$

7 Boolean Branching Program Decomposition

In this section, we prove Theorem 2. Let f be a function and B be a branching program for f of size at most s and length at most $(1+\epsilon)n$. We first apply Lemma 9 with $r = 2$ and $k = 1+\epsilon$. Thus we can write $f = \bigvee_{j=1}^{s} P_j$ where each P_j is a $(2, (1+\epsilon)/2)$-decision program, and we may assume that P_j is deterministic since we are only considering deterministic branching programs. We will prove the following:

Lemma 17. *Let $P = (T_1, T_2)$ be a deterministic $(2, \frac{1+\epsilon}{2}n)$-decision program with $\epsilon \in (0,1)$ and suppose that the smallest 1-certificate of P is of size at least $(1-\delta)n$. If $\delta + \epsilon \le 1/4$ then, for any $\gamma \in (0,1)$, we can write P as:*

$$P = h_0 \vee \bigvee_{i=1}^{t} h_i$$

where $t \le 2^{\gamma n}$, each h_i is a $(1-\epsilon-\delta)n/2$-pseudo-rectangle and $\left|h_0^{-1}(1)\right| \le 2^{(1+b(\epsilon,\delta)/2-\gamma/2)n}$.

Using the lemma, we easily complete the proof of Theorem 2. Let $P_1, \dots, P_u$ be the decision programs in the representation of f in Lemma 9. Observing that any 1-certificate of a P_j must also be a 1-certificate of f, we apply the lemma to each P_j. For $0 \le i \le 2^{\gamma n}$, let h_i^j be the functions appearing in the representation of P_j. Let $f_0 = \bigvee_{j=1}^{s} h_0^j$. Then

$$f = f_0 \vee \bigvee_{i,j} h_i^j$$

where $1 \le j \le s$ and $1 \le i \le 2^{\gamma n}$, as required to prove the Theorem.

So it remains to prove the lemma. Let $P = (T_1, T_2)$ be a decision program satisfying the hypotheses of the lemma. We may assume that every path in T_1 and T_2 has size exactly $\frac{1+\epsilon}{2}n$. As in the previous proof, we define $S_i(\sigma)$, for input σ and $i = 1, 2$, to be set of variables that are not read by T_i on input σ. We say that $S_i(\sigma)$ is the set of variables *missed* by σ in T_i. Note that $|S_i(\sigma)|$ is $\frac{1-\epsilon}{2}n$. For any subset S of variables, and for $i = 1, 2$, let $\mathrm{Miss}_i(S)$ be the set of inputs σ such that $S \subseteq S_i(\sigma)$, i.e., those inputs for which $T_i(\sigma)$ avoids σ. Define $\mathrm{Accept}_i(S)$ to be the set of inputs accepted by T_i that are in $\mathrm{Miss}_i(S)$. Define the function g^i_S which is 1 on the inputs in $\mathrm{Accept}_i(S)$. Observe that the function g^i_S does not depend on the variables of S.

For a pair of sets (S_1, S_2) define $\mathrm{Miss}(S_1, S_2) = \mathrm{Miss}_1(S_1) \cap \mathrm{Miss}_2(S_2)$ and $\mathrm{Accept}(S_1, S_2) = \mathrm{Accept}_1(S_1) \cap \mathrm{Accept}_2(S_2)$. Also, let $g_{S_1,S_2} = g^1_{S_1} \wedge g^2_{S_2}$. Let Q be the set of pairs of sets (S_1, S_2) of variables each of size $\frac{1-\epsilon}{2}n$. Observe that every input accepted by P is accepted by exactly one of the functions g_{S_1,S_2} for $(S_1, S_2) \in Q$, namely
$g_{S_1(\sigma),S_2(\sigma)}$. Thus $P = \bigvee_{(S_1,S_2)\in Q} g_{S_1,S_2}$. Now, in general, the sets S_1 and S_2 may not be disjoint. By allocating their overlap evenly, we can find two disjoint sets U_1 and U_2 of size $\lfloor \frac{(1-\epsilon)n-|S_1\cap S_2|}{2} \rfloor \leq \frac{(1-\epsilon)n-|S_1\cap S_2|}{2}$, such that $U_1 \subseteq S_1$ and $U_2 \subseteq S_2$. Therefore g_{S_1,S_2} is a $\frac{(1-\epsilon)n-|S_1\cap S_2|}{2}$-pseudo-rectangle.

Next we show that the functions g_{S_1,S_2} with $|S_1 \cap S_2| > \delta n$ are identically 0, and hence can be omitted from the $\bigvee$. For such a term, g_{S_1,S_2} does not depend on $S_1 \cap S_2$. If σ is accepted by g_{S_1,S_2} then any input that agrees with σ outside of $S_1 \cap S_2$ must also be accepted, which means that the projection of σ on $X \setminus (S_1 \cap S_2)$ is a 1-certificate of P. But this contradicts the definition of δ. Hence, letting R denote the set of pairs (S_1, S_2) such that $|S_1 \cap S_2| \leq \delta n$, we have $P = \bigvee_{(S_1,S_2)\in P} g_{S_1,S_2}$. As noted, each of these terms is a $\frac{(1-\epsilon-\delta)n}{2}$-pseudo-rectangle. Thus we have expressed P as an $\bigvee$ of functions of the required form. However, the number of terms is too large (close to 2^n, while we want at most $2^{\nu n}$ terms for some small ν).

Next we divide the terms of the $\bigvee$ into two parts depending on the size of $\mathrm{Miss}(S_1, S_2)$. Let $0 \leq \gamma \leq 1$. Call a pair $(S_1, S_2) \in R$ *common* if $\mathrm{Miss}(S_1, S_2) \geq 2^{(1-\gamma)n}$, and *rare* otherwise, and denote the sets of common and rare pairs by R_{common} and R_{rare}. Define the two functions:

$$g_{\mathrm{rare}} = \bigvee_{(S_1,S_2)\in R_{\mathrm{rare}}} g_{S_1,S_2},$$

and

$$g_{\mathrm{common}} = \bigvee_{(S_1,S_2)\in R_{\mathrm{common}}} g_{S_1,S_2}.$$

Thus $P = g_{\mathrm{common}} \vee g_{\mathrm{rare}}$. Now, by definition of R_{common}, and the fact that the sets $\mathrm{Miss}(S_1, S_2)$ are disjoint, the number of pairs $(S_1, S_2) \in R_{\mathrm{common}}$ is at most $2^{\gamma n}$. Thus, writing $P = g_{\mathrm{rare}} \vee \bigvee_{(S_1,S_2)\in R_{\mathrm{common}}} g_{S_1,S_2}$, we have P in the exact form required for the lemma, provided that we can upper-bound $\left|g^{-1}_{\mathrm{rare}}(1)\right|$ appropriately.

The set of inputs accepted by g_{rare} is the union of $\mathrm{Accept}(S_1, S_2)$ over all rare pairs; this is clearly contained in the union of $\mathrm{Miss}(S_1, S_2)$ over all rare pairs. Call this latter union A. Thus it suffices to prove:

Lemma 18. *Let $P = (T_1, T_2)$ be a $(2, \frac{1+\epsilon}{2}n)$-decision program with $\epsilon \in (0,1)$. Let the size of the smallest 1-certificate of P be at least $(1-\delta)n$ and suppose that $\delta + \epsilon \leq 1/4$. Let $\gamma \in (0,1)$ and let $A = \cup_{(S_1,S_2)\in R_{\mathrm{rare}}} \mathrm{Miss}(S_1, S_2)$. Then*

$$\log_2 |A| \leq (1 - \frac{\gamma - b(\epsilon,\delta)}{2})n \tag{5}$$

This lemma is the crux of the argument. Its proof uses elementary information theory. We review some basic definitions and results. Let X be an arbitrary probability space. For any event A, we write $\mathbf{Prob}[A]$ for the probability of A. If $\mathbf{C}$ is a random variable taking values from a finite set S, the binary entropy $H(\mathbf{C})$ is defined to be $-\sum_{s\in S} \mathbf{Prob}[\mathbf{C} = s] \log_2 \mathbf{Prob}[\mathbf{C} = s]$.

Proposition 19. *If $\mathbf{C}$ is a random variable taking on values from some set S then $H(\mathbf{C}) \leq \log_2 |S|$.*

If A is an arbitrary event (measurable subset) of the probability space then the conditional entropy of $\mathbf{C}$ given A is $H(\mathbf{C}|A) = -\sum_{s\in S} \mathbf{Prob}[\mathbf{C} = s|A] \log_2 \mathbf{Prob}[\mathbf{C} = s|A]$.

Proposition 20. *Let $\mathbf{C}$ be some random variable taking values on a finite set S and let A be an event. Let $S_{\mathbf{C}}(A)$ denote the set of $s \in S$ such that $\mathbf{Prob}[\mathbf{C} = s|A] > 0$. Then $H(\mathbf{C}|A) \geq \log_2 \mathbf{Prob}[A] - \max_{s\in S_{\mathbf{C}}(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s]$.*

Proof.

$$\begin{aligned}
H(\mathbf{C}|A) &= -\sum_{s\in S} \mathbf{Prob}[\mathbf{C} = s|A] \log_2 \mathbf{Prob}[\mathbf{C} = s|A] \\
&\geq -\max_{s\in S_C(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s|A] \\
&\geq -\max_{s\in S_C(A)} \log_2 \frac{\mathbf{Prob}[\mathbf{C} = s]}{\mathbf{Prob}[A]} \\
&\geq \log \mathbf{Prob}[A] - \max_{s\in S_C(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s]
\end{aligned}$$

□

If $\mathbf{B}_1$ and $\mathbf{B}_2$ are random variables on the same probability space taking values in S_1 and S_2, respectively, $H(\mathbf{B}_1, \mathbf{B}_2)$ is the entropy of the random variable consisting of the pair $(\mathbf{B}_1, \mathbf{B}_2)$. The conditional entropy of $\mathbf{B}_1$ given $\mathbf{B}_2$ is defined by $H(\mathbf{B}_1|\mathbf{B}_2) = H(\mathbf{B}_1, \mathbf{B}_2) - H(\mathbf{B}_2) = \sum_{s\in S_2} H(\mathbf{B}_1|\mathbf{B}_2 = s) \cdot \mathbf{Prob}[\mathbf{B}_2 = s]$. The mutual information of $\mathbf{B}_1$ and $\mathbf{B}_2$ is defined to be $I(\mathbf{B}_1, \mathbf{B}_2) = H(\mathbf{B}_1) + H(\mathbf{B}_2) - H(\mathbf{B}_1, \mathbf{B}_2)$.

Proposition 21. *Let $\mathbf{B}_1, \mathbf{B}_2$ be random variables taking values on finite sets S_1 and S_2. Then*

1. $H(\mathbf{B}_1, \mathbf{B}_2) = H(\mathbf{B}_1|\mathbf{B}_2) + H(\mathbf{B}_2|\mathbf{B}_1) + I(\mathbf{B}_1, \mathbf{B}_2)$.

2. $H(\mathbf{B}_1, \mathbf{B}_2) \leq H(\mathbf{B}_1) + H(\mathbf{B}_2)$.

3. $H(\mathbf{B}_1|\mathbf{B}_2) \leq \max_{s \in S_2} H(\mathbf{B}_1|\mathbf{B}_2 = s)$.

Given two random variables $\mathbf{B}, \mathbf{C}$ we say that $\mathbf{B}$ *determines* $\mathbf{C}$ if $\mathbf{C}$ is a function of $\mathbf{B}$. We have:

Proposition 22. *1. If $\mathbf{B}$ and $\mathbf{C}$ are random variables such that $\mathbf{B}$ determines $\mathbf{C}$ then (a) $H(\mathbf{C}) \leq H(\mathbf{B})$ and (b) $H(\mathbf{C}|\mathbf{B}) = 0$.*

2. If $\mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$ are random variables such that $\mathbf{B}_1$ determines $\mathbf{C}_1$ and $\mathbf{B}_2$ determines $\mathbf{C}_2$ then $I(\mathbf{C}_1, \mathbf{C}_2) \leq I(\mathbf{B}_1, \mathbf{B}_2)$.

Proposition 23. *Let $\mathbf{B}_1, \mathbf{B}_2$ and $\mathbf{C}_1, \mathbf{C}_2$ be pairs of random variables such that $\mathbf{B}_i$ determines $\mathbf{C}_i$ for $i = 1, 2$. Then*

$$H(\mathbf{B}_1, \mathbf{B}_2) + H(\mathbf{C}_1, \mathbf{C}_2) \leq H(\mathbf{B}_1) + H(\mathbf{B}_2) + H(\mathbf{C}_1|\mathbf{C}_2) + H(\mathbf{C}_2|\mathbf{C}_1).$$

Proof. By Proposition 21(1) and Proposition 22(2):

$$\begin{aligned} H(\mathbf{C}_1, C_2) &= H(\mathbf{C}_1|\mathbf{C}_2) + H(\mathbf{C}_2|\mathbf{C}_1) + I(\mathbf{C}_2, \mathbf{C}_1) \\ &\leq H(\mathbf{C}_1|\mathbf{C}_2) + H(\mathbf{C}_2|\mathbf{C}_1) + I(\mathbf{B}_2, \mathbf{B}_1) \\ &= H(\mathbf{C}_1|\mathbf{C}_2) + H(\mathbf{C}_2|\mathbf{C}_1) + H(\mathbf{B}_1) + H(\mathbf{B}_2) - H(\mathbf{B}_1, \mathbf{B}_2). \end{aligned}$$

□

We will need a well known technical fact concerning sums of binomial coefficients (whose proof we give since we don't know a reference).

Proposition 24. *If $k \leq n/2$, $\log_2(\sum_{i \leq k} \binom{n}{i}) \leq nH(k/n)$.*

Proof. By the binomial theorem, for any $p \leq 1/2$, we have $1 \geq \sum_{i \leq k} p^i(1-p)^{n-i}\binom{n}{i} \geq p^k(1-p)^{n-k} \sum_{i \leq k} \binom{n}{i}$. Setting $p = k/n$ and taking logarithms yields the desired inequality. □

Proof of Lemma 18. Consider the probability space on $\{0,1\}^X$ (which we identify with $\{0,1\}^n$), with the uniform distribution. Events in this probability space correspond to subsets C of $\{0,1\}^n$. The probability of C is $\mathbf{Prob}[C] = |C|/2^n$. For a subset (event) C, let $\mu[C] = \log_2 |C| = n + \log_2 \mathbf{Prob}[C]$.

We define the following random variables. For $i = 1, 2$, let $\mathbf{P}_i$ be the path taken in T_i and $\mathbf{S}_i$ be the set of variables missed in T_i by a random input. Observe that $\mathbf{S}_i$ is a function of $\mathbf{P}_i$.

The basic intuition for the proof is this. We are trying to upper bound the size of A by something like $2^{(1-\nu)n}$ where $\nu > 0$, i.e. that $\mu[A] \leq (1-\nu)n$. Note that A is defined to be the event that $(\mathbf{S}_1, \mathbf{S}_2) \in R_{\text{rare}}$, and so by Proposition 20, $\mu[A] \leq H(\mathbf{S}_1, \mathbf{S}_2|A) + \max_{(S_1,S_2) \in R_{\text{rare}}} \mu[(\mathbf{S}_1, \mathbf{S}_2) = (S_1, S_2)]$. The second term is at most $(1-\gamma)n$ by the definition of R_{rare}. Thus we want to upper bound $H(\mathbf{S}_1, \mathbf{S}_2|A) = H(\mathbf{S}_1|\mathbf{S}_2, A) + H(\mathbf{S}_2|\mathbf{S}_1, A) + I(\mathbf{S}_1, \mathbf{S}_2|A)$

by $\gamma' n$, where γ' is a constant less than γ. Now observe that given A, $|\mathbf{S}_1 \cap \mathbf{S}_2|$ is small, and therefore $\mathbf{S}_1$ and $\mathbf{S}_2$ are "approximately" complementary subsets of variables, and so one of them approximately determines the other. This allows us to conclude that the first two terms in the sum are small. We use Proposition 22 to upper bound the third term by $I(\mathbf{P}_1, \mathbf{P}_2|A)$. Intuitively, this represents the amount of information $\mathbf{P}_1$ reveals about $\mathbf{P}_2$ (and vice versa) given that A holds. Now A implies that $\mathbf{P}_1$ and $\mathbf{P}_2$ read very few variables in common, and this can be used to show that the mutual information is small. Although the intuition is based on mutual information, in the formal argument we use Proposition 23 and do not explicitly mention mutual information.

We now proceed with the proof by considering $H(\mathbf{P}_1, \mathbf{P}_2|A) + H(\mathbf{S}_1, \mathbf{S}_2|A)$. As noted, Proposition 20 implies $H(\mathbf{S}_1, \mathbf{S}_2|A) \geq \mu[A] - (1-\gamma)n$. To apply the same proposition to $H(\mathbf{P}_1, \mathbf{P}_2|A)$ we note that any pair of paths (P_1, P_2) corresponding to a point in $\sigma \in A$ together specify at least $(1-\delta)n$ variables and so $\mu[(\mathbf{P}_1, \mathbf{P}_2) = (P_1, P_2)] \leq \delta n$. Consequently $H(\mathbf{P}_1, \mathbf{P}_2|A) \geq \mu[A] - \delta n$ and so

$$H(\mathbf{P}_1, \mathbf{P}_2|A) + H(\mathbf{S}_1, \mathbf{S}_2|A) \geq 2\mu[A] - (1-\gamma+\delta)n.$$

On the other hand, by Proposition 23,

$$\begin{aligned} H(\mathbf{P}_1, \mathbf{P}_2|A) + H(\mathbf{S}_1, \mathbf{S}_2|A) &\leq H(\mathbf{P}_1|A) + H(\mathbf{P}_2|A) + H(\mathbf{S}_1|\mathbf{S}_2, A) + H(\mathbf{S}_2|\mathbf{S}_1, A) \\ &\leq (1+\epsilon)n + H(\mathbf{S}_1|\mathbf{S}_2, A) + H(\mathbf{S}_2|\mathbf{S}_1, A) \end{aligned}$$

where the last inequality comes from Proposition 19 and the fact that the number of paths in each tree is $2^{(1+\epsilon)n/2}$.

Since the two remaining terms are symmetric it remains to upper bound $H(\mathbf{S}_1|\mathbf{S}_2, A)$. Define the random variables $\mathbf{I} = \mathbf{S}_1 \cap \mathbf{S}_2$ and $\mathbf{J} = X - (\mathbf{S}_1 \cup \mathbf{S}_2)$ and observe that the triple $(\mathbf{S}_2, \mathbf{I}, \mathbf{J})$ determines $\mathbf{S}_1$. Hence:

$$\begin{aligned} H(\mathbf{S}_1|\mathbf{S}_2, A) &\leq H(\mathbf{S}_2, \mathbf{I}, \mathbf{J}|\mathbf{S}_2, A) && \text{by Proposition 22(1)} \\ &\leq H(\mathbf{S}_2|\mathbf{S}_2, A) + H(\mathbf{I}|\mathbf{S}_2, A) + H(\mathbf{J}|\mathbf{S}_2, A) && \text{by Proposition 21(2)} \\ &= H(\mathbf{I}|\mathbf{S}_2, A) + H(\mathbf{J}|\mathbf{S}_2, A) && \text{by Proposition 22(1)} \end{aligned}$$

Since A implies $|\mathbf{I}| \leq \delta n$, Proposition 19 and Proposition 24 imply that $H(\mathbf{I}|\mathbf{S}_2, A) \leq \log_2 \sum_{i \leq \delta n} \binom{n}{i} \leq nH(\delta)$. Now, by Proposition 21(3), $H(\mathbf{J}|\mathbf{S}_2, A) \leq \max_{S_2} H(\mathbf{J}|\mathbf{S}_2 = S_2, A)$. Given A, $|\mathbf{J}| \leq (\delta+\epsilon)n$ and given $\mathbf{S}_2 = S_2$, $\mathbf{J}$ is contained in $\overline{S_2}$ which is a set of size $(1+\epsilon)n/2$. Again, using Proposition 19 and Proposition 24 we conclude $H(\mathbf{J}|\mathbf{S}_2, A) \leq n\frac{1+\epsilon}{2}H\left(\frac{2(\epsilon+\delta)}{1+\epsilon}\right)$ (here we use the hypothesis that $\epsilon + \delta \leq 1/4$). Thus:

$$H(\mathbf{S}_1|\mathbf{S}_2, A) \leq \left[\frac{1+\epsilon}{2}H\Big(\frac{2(\epsilon+\delta)}{1+\epsilon}\Big) + H(\delta)\right] n$$

and so

$$\begin{aligned} 2\mu[A] - (1-\gamma+\delta)n &\leq H(\mathbf{P}_1, \mathbf{P}_2|A) + H(\mathbf{S}_1, \mathbf{S}_2|A) \\ &\leq (1 + \epsilon + 2\left[\frac{1+\epsilon}{2}H\left(\frac{2(\epsilon+\delta)}{1+\epsilon}\right) + H(\delta)\right])n. \end{aligned}$$

Solving, we find

$$\mu[A] \le \left(1 + \frac{1}{2}\left[\epsilon + \delta + (1+\epsilon)H\left(\frac{2(\epsilon+\delta)}{1+\epsilon}\right) + 2H(\delta) - \gamma\right]\right)n$$
$$= (1 + \frac{b(\epsilon,\delta) - \gamma}{2})n$$

which is what we wanted to prove. □

8 Proof of Lemma 5

We being with the proof of the first part of the lemma. This will follow immediately from two additional lemmas.

For any $n \times n$ matrix M and any $0 \le \beta \le 1$, define $\Phi_M(\beta)$ to be $1/n$ times the minimum rank of any $\beta n \times \beta n$ minor of M that does not include any diagonal element of M. Trivially, $\Phi_M(\beta) = \Phi_N(\beta)$ if M and N have the same off-diagonal elements; in particular $\Phi_M(\beta) = \Phi_{M^{[0]}}(\beta)$.

Lemma 25. *Let M be a symmetric matrix over $GF(q)$ and $D \subseteq GF(q)$. Let f be the restriction of of QF_M to D^n. Then $\Psi_f(\beta) \ge \Phi_M(\beta)$ for $\beta \in [0,1]$.*

Let G be a finite group. A *G-matrix* over $GF(q)$ is a matrix N whose rows are indexed by G, such that for all $g_1, g_2 \in G$ and $j \in [1,n]$, $N_{g_1 \cdot g_2, j} = N_{g_1,j} N_{g_2,j}$ (i.e., each column is a multiplicative character of G).

Lemma 26. *Let N be an $n \times n$ G-matrix of full rank, where G is a group of order n, and let M be any $n \times n$ matrix that agrees with N off the diagonal. Then $\Phi_M(\beta) = \beta^2$.*

Since a GFT matrix associated to a group G is a G-matrix of full rank, Part 1 of the Theorem follows immediately from the lemmas. So we now prove them.

Proof of Lemma 25. Let $g = g_1 \wedge g_2$ be a portion of f such that for some disjoint sets of variables A_1 and A_2 of size each βn, g_i does not depend on the variables in A_i. Let us denote assignments to the variables as $(\rho, \sigma_1, \sigma_2)$ where ρ is an assignment to the variables outside of $A_1 \cup A_2$, and σ_j is an assignment to A_j. For fixed ρ, there are $|D|^{2\beta n}$ assignments $(\rho, \sigma_1, \sigma_2)$ that extend it. Let $\Gamma = \Gamma(\rho)$ denote the subset of these assignments that are accepted by g. We will show that $|\Gamma(\rho)| \le |D|^{(2\beta - \Phi_M(\beta))n}$; the desired bound is then obtained by summing over the $|D|^{(1-2\beta)n}$ choices of ρ.

So let us prove the upper bound on $|\Gamma|$. Since g_i does not examine any variables in A_i, $\Gamma = \{\rho\} \times \Sigma_1 \times \Sigma_2$, where

$$\Sigma_1 = \{\sigma_1 : \ (\rho, \sigma_1, \sigma_2) \in \Gamma \text{ for some } \sigma_2\}$$
$$\Sigma_2 = \{\sigma_2 : \ (\rho, \sigma_1, \sigma_2) \in \Gamma \text{ for some } \sigma_1\}$$

Substitute the values assigned by ρ into X^TMX. Because M is symmetric, we obtain a polynomial of the form $A_1^T N A_2 + F_1(A_1) + F_2(A_2)$, where N is a $\beta n \times \beta n$ matrix equal to twice the minor of M indexed by $A_1 \times A_2$ and each F_i is some polynomial function of the entries of A_i. Lemma 27 below, which is a slight generalization of results in [BRS93, Tha98], implies the required bound on $|\Gamma|$.

Lemma 27. *Let N be a $t \times t$ matrix over $GF(q)$. For $i = 1, 2$, let $F_i : GF(q)^t \to GF(q)$ be arbitrary functions, and let F denote the function from $GF(q)^t \times GF(q)^t$ to $GF(q)$ given by $F(\sigma_1, \sigma_2) = \sigma_1 N \sigma_2 + F_1(\sigma_1) + F_2(\sigma_2)$. For a set $D \subseteq GF(q)$, suppose that $\Sigma_1, \Sigma_2 \subseteq D^t$ satisfy $F(\sigma_1, \sigma_2) = 0$ for every $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$. Then $|\Sigma_1 \times \Sigma_2| \leq |D|^{2t-\mathrm{rank}(N)}$.*

Proof. For $i = 1, 2$ fix some $\sigma_i^* \in \Sigma_i$. Then for any $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$, we have $(\sigma_1 - \sigma_1^*)N(\sigma_2 - \sigma_2^*) = F(\sigma_1, \sigma_2) + F(\sigma_1^*, \sigma_2^*) - F(\sigma_1, \sigma_2^*) - F(\sigma_1^*, \sigma_2) = 0$. Defining V_i for $i = 1, 2$ to be the linear span of $\Sigma_i^* = \{\sigma - \sigma_i^* : \sigma \in \Sigma_i\}$ we have that $v_1 N v_2 = 0$ for all $v_1 \in V_1$ and $v_2 \in V_2$. This implies $\dim(V_1) + \dim(V_2) \leq 2t - \mathrm{rank}(N)$. The lemma now follows since $|\Sigma_i| = |\Sigma_i^*| \leq |D|^{\dim(V_i)}$. $\square$

This completes the proof of Lemma 25. $\square$

We now turn to the proof of Lemma 26. This lemma is an immediate consequence of the following lemma, which says, roughly, that every large minor of a GFT matrix has large rank. The lemma both simplifies and improves a bound in [BRS93] which showed that every $u \times t$ minor of such a matrix has rank at least $ut/(\eta(n, u, t)\, n)$, where $\eta(n, u, t)$ is a function that is typically logarithmic in n. (This new bound also improves the lower bound on the size of read-k branching programs proved in that paper by shaving off a factor of k in the exponent.)

Lemma 28. *Let N be any G-matrix over $GF(q)$ where G is a group of order n. If N has full column rank, the rank of any $u \times t$ minor of N is at least ut/n.*

Proof. For $V \subseteq G$ and any set J of columns, let $N_{V,J}$ denote the sub-matrix of N corresponding to the rows of V and columns of J. Fix V of size u and J of size t. Since N has full column rank, the columns of $N_{G,J}$ are independent implying that it has a $t \times t$ minor $N_{W,J}$, for some $W \subseteq G$, which has full rank.

For any fixed $g^* \in V$ and a random $g \in G$, $\Pr[gg^* \in W] = \Pr[g \in W(g^*)^{-1}] = t/n$. By linearity of expectation, when g is randomly chosen from G, the expected number of $g^* \in V$ for which $gg^* \in W$ is ut/n. Therefore, for some fixed g and some $H \subseteq V$ where $|H| \geq ut/n$, we have $gH \subseteq W$. We show that $N_{H,J}$ has rank at least ut/n from which the lemma follows.

Because G is a group, $|gH| = |H| \geq ut/n$. Since $N_{gH,J}$ is a sub-matrix consisting of at least ut/n rows of $N_{W,J}$, it follows that $\mathrm{rank}(N_{gH,J}) \geq ut/n$. By the definition of N, for each $j \in J$, $N_{gH,j} = N_{g,j} N_{H,j}$, that is, each column of $N_{H,W}$ is multiplied by some constant to get the corresponding column in $N_{gH,W}$. Therefore $\mathrm{rank}(N_{H,J}) \geq \mathrm{rank}(N_{gH,W}) \geq ut/n$. $\square$

This completes the proof of part 1 of Lemma 5. We now consider part 2. Let M be a GFT matrix. It suffices to show that for any partial assignment ρ that fixes all but 2 variables, z_1, z_2 of X, the restriction $QF_{M^{[0]}}\lceil_\rho$ is not the constant function. Since M is symmetric, and its off-diagonal elements are non-zero, the restriction satisfies $QF_{M^{[0]}}\lceil_\rho = 2a(z_1+b)(z_2+c)+d$ for some constants $a \neq 0, b, c, d \in GF(q)$. Since q is odd, setting z_1 to any value in $GF(q)-\{-b\}$, we have $2a(z_1+b) \neq 0$ and it follows that $2a(z_1+b)(z_2+c)+d$ takes on all possible values in $GF(q)$ by varying z_2. Thus, $QF_{M^{[0]}}\lceil_\rho$ is non-constant.

Finally, to prove Part 3, let M be the $n \times n$ Sylvester matrix over $GF(3)$, and let $f = BQF_M$. Notice that the above argument fails in this case, even for $M^{[0]}$, because the values of z_1, z_2 are restricted to $\{0,1\}$. To handle this case, we first need a lemma showing that in every sufficiently large principal minor of M, there exist principal minors whose entries sum to arbitrary values.

Lemma 29. *Let M be the $n \times n$ Sylvester matrix over $GF(3)$ where $n = 2^k$. Let $I \subseteq [1,n]$ be an arbitrary subset of size at least $4\sqrt{n}\log n$. For every $a \in GF(3)$, there exists $J \subseteq I$, with $|J| \leq 3$ such that the sum of the entries in $M_{J,J}$ is a.*

Proof. For $a = 0$, set $J = \emptyset$. So assume $a \in \{1,-1\}$. Recall that the row and column index set of the Sylvester matrix is the set of binary vectors of length k, which is identified naturally with the set of subsets of $[1,k]$ and we view I as a collection of $4\sqrt{n}\log n$ such subsets. For $a = 1$ and for $a = -1$, we want a sub-collection J of I such that the sum of entries in $M_{J,J}$ is a. The following easily verifiable fact gives a criterion for a collection of size 3 to satisfy this.

Proposition 30. *Suppose A_0, B_1, B_2 are distinct subsets of $[1,k]$ such that $|A_0|$, $|B_1|$, and $|B_2|$ are all even or all odd and let $J = \{A_0, B_1, B_2\}$.*

1. *If $|A_0 \cap B_1|$ and $|A_0 \cap B_2|$ are both even and $|B_1 \cap B_2|$ is odd then the sum of entries in $M_{J,J}$ is -1*
2. *If $|A_0 \cap B_1|$ and $|A_0 \cap B_2|$ are both odd and $|B_1 \cap B_2|$ is even then the sum of entries in $M_{J,J}$ is +1.*

We will also need the so-called "Eventown-Oddtown" theorems (see [BF92]), stated as a proposition below. We sketch the proof for the sake of completeness.

Proposition 31. *Let $\mathcal{F}$ be a family of sets. We say that $\mathcal{F}$ has property $P_{i,e}$ (respectively $P_{i,o}$) if the common intersection of every collection of i distinct sets from $\mathcal{F}$ is even (respectively odd). The following table gives an upper bound on the size of any family of subsets of $[1,k]$ satisfying some of these properties:*

	$P_{1,e}$	$P_{1,o}$
$P_{2,e}$	$2^{k/2}$	k
$P_{2,o}$	k	$2^{k/2}$

Proof Sketch. All these bounds can be proved using simple linear algebra by associating each set with its characteristic function, which can also be viewed as vector of length k over $GF(2)$. Let $|\mathcal{F}| = m$ and let S be the associated set of m characteristic vectors.

First, suppose $\mathcal{F}$ satisfies $P_{1,o}$ and $P_{2,e}$. Define an $m \times k$ matrix M (called the *incidence matrix*) whose row vectors belong to S. Consider the $m \times m$ matrix MM^T. The $(i,j)^{\text{th}}$ entry equals the parity of intersection of the i^{th} and j^{th} sets. By the property satisfied by $\mathcal{F}$, it follows that MM^T is an identity matrix of order m. Since rank is sub-multiplicative, and bounded by both the row and column dimension, $m = \text{rank}(MM^T) \leq \text{rank}(M) \leq k$.

Next, consider the case where $\mathcal{F}$ satisfies $P_{1,e}$ and $P_{2,o}$. Define two families of sets

$$\mathcal{G} = \{A \cup \{k+1\} : A \in \mathcal{F}\} \cup \{[1,k]\}$$
$$\mathcal{H} = \{A \cup \{k+1\} : A \in \mathcal{F}\} \cup \{1\}.$$

Observe that $|\mathcal{G}| = |\mathcal{H}| = m+1$. Now let M and N be the $(m+1) \times (k+1)$ incidence matrices of $\mathcal{G}$ and $\mathcal{H}$ respectively. Consider the $(m+1) \times (m+1)$ matrix MN^T. Each diagonal entry equals 1 and the remaining entries except for those in the $(m+1)^{\text{st}}$ column equal 0. Thus MN^T is an upper triangular matrix with all diagonal entries non-zero, implying that $m+1 = \text{rank}(MN^T) \leq \text{rank}(M) \leq k+1$, which finishes the proof for this case.

For the case where $\mathcal{F}$ satisfies $P_{1,e}$ and $P_{2,e}$, the vector space orthogonal to S, namely $S^\perp$, contains S as a subset. Since the sum of dimensions of S and $S^\perp$ is k, the dimension of S is at most $k/2$, implying the bound on the size of S.

The last case where $\mathcal{F}$ satisfies $P_{1,o}$ and $P_{2,o}$ follows from the previous case by fixing a vector $s \in S$ and applying the above argument to $S' = \{v - s : v \in S\}$. $\square$

Continuing the proof of Lemma 29, we now show that I contains a collection $J = \{A_0, B_1, B_2\}$ satisfying the hypothesis of the first part of Proposition 30 and thus the sum of entries in $M_{J,J}$ is -1. A similar argument handles the other case.

Since $|I| \geq 4\sqrt{n}\log n$, there is a sub-family $\mathcal{F}$ of size at least $2\sqrt{n}\log n$ such that every set in $\mathcal{F}$ has even size or every set in $\mathcal{F}$ has odd size. Consider a sub-family $\mathcal{G}$ of $\mathcal{F}$ which is maximal subject to the condition that for any distinct $A, B \in \mathcal{G}$, $|A \cap B|$ is odd. For each set $A \in \mathcal{G}$, let $\mathcal{E}(A)$ be the subfamily consisting of those sets $C \in \mathcal{F} - \mathcal{G}$ such that $|A \cap C|$ is even. By the maximality of $\mathcal{G}$, $\cup_{A \in \mathcal{G}} \mathcal{E}(A) = \mathcal{F} - \mathcal{G}$ which implies $\sum_{A \in \mathcal{G}} |\mathcal{E}(A)| \geq |\mathcal{F} - \mathcal{G}|$. Choose $A_0 \in \mathcal{G}$ for which $|\mathcal{E}(A_0)|$ is maximum and write $\mathcal{E} = \mathcal{E}(A_0)$. Then $|\mathcal{E}|\,|\mathcal{G}| \geq |\mathcal{F} - \mathcal{G}|$, or $(|\mathcal{E}|+1)\,|\mathcal{G}| \geq |\mathcal{F}|$. To finish the proof it suffices to show that there are $B_1, B_2 \in \mathcal{E}$ whose intersection has odd size. Using Proposition 31, if every pair of distinct sets in $\mathcal{E}$ has an intersection of even size, then

$$|\mathcal{G}|\,(|\mathcal{E}|+1) \leq \max\{2^{k/2}(k+1), k(2^{k/2}+1)\} < 2k2^{k/2} = 2\sqrt{n}\log n \leq |\mathcal{F}|$$

which contradicts $(|\mathcal{E}|+1)\,|\mathcal{G}| \geq |\mathcal{F}|$. Therefore, the claim holds. $\square$

We now have the tools to prove Part 3 of Lemma 5. Let ρ be a partial assignment to all but $24\sqrt{n}\log n$ variables of X and let $Z \subseteq X$ be the variables unset by ρ. Then $BQF_M\lceil_\rho = Z^TBZ + A \cdot Z + C$, where B denotes the sub-matrix of M corresponding to the rows and columns corresponding to Z, and the vector A and scalar C are determined by ρ and M. It suffices to show that $q(Z) = Z^TBZ + A \cdot Z$ takes on all possible values in $GF(3)$ for the various choices of 0-1 assignments to Z.

Setting all variables to 0 makes the function 0. Fix $a \in \{-1, 1\}$. Our goal is to identify three variables such that setting them to 1 and everything else to 0 will make the function equal to a. Classify each variable x_j by the pair $(M_{j,j}, A_j) \in \{-1,1\} \times \{-1,0,1\}$. There are 6 possible values of this pair, and so there is a set of at least $4\sqrt{n}\log n$ variables $Z' \subseteq Z$ that belong to the same class. If we set any three variables in Z' to 1, and everything else to 0, $q(Z)$ evaluates to the sum of the off-diagonal entries in the 3×3 principal minor corresponding to these variables. By Lemma 29 such a minor exists whose sum evaluates to $a \pmod 3$, and Part 3 of the lemma follows.

9 Semantic versus syntactic branching programs

A path P in a branching program is a *semantic path* if it is consistent with some input. A path is not semantic if and only if there is some pair of nodes v and w on the path that have the same variable label x, such that the arcs following them have different labels. A path P is *read-k* for some integer k, if no variable appears in P more than k times. If Z is a subset of the variables of a branching program B, we say that B is *syntactic (semantic) read-k on Z* if every path (respectively every semantic path) in B is read-k. (If Z contains all the variables, we omit the qualifying phrase in which case our definition is the standard one.)

In this section we exhibit, for every k, a simple function f_k that can be computed in linear size by a semantic read-twice branching program but requires an exponential size syntactic read-k branching program. The key to defining our separating functions is the construction of functions $g_k(X,Y)$ that can be computed by linear size branching programs that are semantic read-twice on X but require exponential size on any branching program that is syntactic read-k on X. Before we give the construction of $g_k(X,Y)$, we describe the relationship between $g_k(X,Y)$ and the separating functions.

Definition 32. *For a variable set Y, let $Y_1, Y_2, \ldots, Y_k$ be disjoint copies of Y. Write $y \cong y'$ if for some i, j, $y \in Y_i$ and $y' \in Y_j$ both correspond to the same variable in Y. The k^{th} extension of g on Y, denoted by $g^{(k)}(X, Y_1, \ldots, Y_k)$, is defined to be one if and only if (i) $g(X, Y_1) = 1$ and (ii) all of the blocks $Y_1, Y_2, \ldots, Y_k$ have the same setting.*

The relationship between computing $g(X,Y)$ and its k^{th} extension on Y is given by the following lemma:

Lemma 33. *Let $g(X,Y)$ be a Boolean function.*

1. *If $g(X,Y)$ can be computed by branching program P that is syntactic read-k on Y, then $g^{(k)}(X, Y_1, \ldots, Y_k)$ can be computed by a branching program Q that is syntactic read-twice on $\bigcup_i Y_i$ and satisfies $\mathrm{size}(Q) = \mathrm{size}(P) + O(\sum_i |Y_i|)$. Furthermore, any syntactic or semantic properties of P with respect to X also hold in Q.*

2. *If $g^{(k)}(X, Y_1, \ldots, Y_k)$ can be computed by a syntactic read-k branching program, then $g(X,Y)$ can be computed by a branching program that is syntactic read-k on X and has the same size.*

Proof. Part 2 is easy: if Q' computes $g^{(k)}(X, Y_1, \ldots, Y_k)$, replace each occurrence of a variable of any Y_i in Q' by its corresponding variable of Y; the resulting branching program remains syntactic read-k on X and has the same size.

For Part 1, Q consists of two blocks and accepts an input if and only if both blocks accept. The first block of Q checks that all corresponding variables in the various Y_i's are equal. This can be done in size $O(\sum_i |Y_i|)$, looking at each variable exactly once. For the second block, transform P as follows: at any node accessing a variable $y \in Y$, replace it with its corresponding variable in Y_i, where i denotes the maximum number of occurrences of y along any path from the source to that node.

Observe that Q is syntactic read-twice on $\bigcup_i Y_i$ and has the desired size. Moreover, the first block of Q does not access X, and the second block of Q is structurally equivalent to P where the nodes that are labeled with variables in X are identical. It follows that the syntactic or semantic properties with respect to X are identical. □

Suppose that $g(X, Y)$ can be computed efficiently by a branching program that is syntactic read-k on Y and semantic

read-once on X, but requires exponential size to be computed by any branching program that is syntactic read-k on X. Lemma 33 implies that the $g^{(k)}(X, Y_1, \ldots, Y_k)$ can be computed efficiently by a semantic read-twice branching program but requires an exponential size syntactic read-k branching program. Thus $g^{(k)}(X, Y_1, \ldots, Y_k)$ witnesses the desired separation. For the rest of this section, we focus on producing such a $g(X, Y)$.

Definition 34. *For a k-dimensional hypercube $[1, n]^k$ of side n, the n hyperplanes perpendicular to the d^{th} axis, $d \in [0, k-1]$ are referred to as* d-planes. *In other words, the i^{th} d-plane, for $i \in [1, n]$, is the set $\{v \in [1, n]^k : v_d = i\}$. We define the predicate $g_k(X, Y)$ as follows. Without loss of generality, let $k + 1 = 2^r$, for some r. Let X and Y be sets of variables corresponding to a $(k+1)$-dimensional hypercube of side n. The variables of X are Boolean but we use the Fourier representation where -1 and 1 are identified with* **true** *and* **false** *respectively and also treated as elements of $GF(3)$. For each $v \in [1, n]^{k+1}$, Y contains r variables $y_v^0, y_v^1, \ldots, y_v^{r-1}$ which together determine an integer $y_v = y_v^{r-1} \ldots y_v^1 y_v^0 \in [0, k]$. For any $d \in [0, k]$, $v \in [1, n]^{k+1}$, define*

$$x_v^d = \begin{cases} x_v & \text{if } y_v = d \\ 1 & \text{otherwise,} \end{cases}$$

If $y_v = d$, we say that x_v is active for d. *Define the following polynomial over $GF(3)$:*

$$H_d(X, Y) = \sum_{i \in [1,n]} \prod_{\substack{v \in [1,n]^{k+1}, \\ v_d = i}} x_v^d$$

and set

$$g_k(X, Y) = \bigwedge_{d \in [0,k]} (H_d(X, Y) \equiv 0 \pmod 3)$$

Informally, we can describe $g_k(X,Y)$ as follows. The variables of X and Y can each be viewed as defining two $(k+1)$-dimensional arrays, denoted x and y respectively, where the entries of x are in $\{-1,1\}$ and those in y are in $[0,k]$. We construct $k+1$ additional arrays $(x^d : 0 \leq d \leq k)$ whose entry in position v is x_v if $y_v = d$ and is 1 otherwise. We say that x_v is *active in* x^d if $y_v = d$, and let $X^d \subseteq X$ be the set of such variables. Note that the sets $(X^d : 0 \leq d \leq k)$ partition X. Each function H_d is computed by considering each i-plane of x^d (for $0 \leq d \leq k$) and multiplying the entries on the i-plane, and then summing over i. This can be done by a branching program of size $O(|X^d| + |Y|)$ that reads each variable of $Y \cup X^d$ exactly once and no other entry of X. Thus $g_k(X,Y)$ can be computed by a branching program of size $O(|X| + (k+1)|Y|)$ that is read-$(k+1)$ on Y and read-once on X. On the other hand, we have the following hardness result.

Theorem 35. *Any non-deterministic branching program that is syntactic read-k on X requires exponential size to compute $g_k(X,Y)$.*

As noted above, Lemma 33 then implies:

Corollary 36. *Let $f_k = g_k^{(k+1)}$ be the $(k{+}1)$-st extension of g_k on Y. There is a simple semantic read-twice branching program of linear size computing f_k. On the other hand, any non-deterministic syntactic read-k branching program for f_k requires exponential size.*

The proof of Theorem 35 relies heavily on machinery developed in [BRS93, Tha98]. We call a function $R(X)$ a *planar pseudo-rectangle of order m* if there exist disjoint sets $A, B \subseteq X$ such that

1. $R(X)$ can be expressed as $R'(X \backslash B) \wedge R''(X \backslash A)$, for some functions R' and R''.

2. For some d and index set $J \subseteq [1,n]$ of size m, $A = \{a_j : j \in J\}$ and $B = \{b_j : j \in J\}$ such that for each $j \in J$, a_j and b_j correspond to points in the j^{th} d-plane.

Proposition 37 ([BRS93, Tha98]). *Let $f(X)$ be a Boolean function, where X corresponds to the $(k{+}1)$-dimensional hypercube of side n. Let $m = n/(6(k+1)2^{k+1})$ and $p = 36 \cdot k \cdot 2^k$. Suppose $\left|R^{-1}(1)\right| \leq t$ for any planar pseudo-rectangle R of order m that is a portion of f. Then, any non-deterministic read-k branching program for f requires size $\left(\left|f^{-1}(1)\right|/t\right)^{1/(2kp)}$.*

The above proposition can be extended to handle branching programs that are syntactic read-k on a subset of the variables. Suppose $f(X,Y)$ is a Boolean function. We say that $R(X,Y)$ a *planar pseudo-rectangle of order m on X* if there exist disjoint sets $A, B \subseteq X$ such that

1. $R(X,Y)$ can be expressed as $R'(X \backslash B, Y) \wedge R''(X \backslash A, Y)$, for some functions R' and R''.

2. For some d and index set $J \subseteq [1,n]$ of size m, $A = \{a_j : j \in J\}$ and $B = \{b_j : j \in J\}$ such that for each $j \in J$, a_j and b_j correspond to points in the j^{th} d-plane.

We obtain:

Corollary 38. *Proposition 37 holds for branching programs computing $g(X,Y)$ that are syntactic read-k on X when we consider planar pseudo-rectangles $R(X,Y)$ on X that are portions of $g(X,Y)$.*

Proof (Sketch). Let P be a branching program computing $f(X,Y)$. The proof is similar to that of Proposition 37. A critical part of the proof of Proposition 37 involves keeping track of variables read along the various paths. We have to only modify the proof so that along each path in P, we keep track of the variables in X but ignore the variables in Y. The rest of the proof can be duplicated; the only modification is that each function that arises as part of the proof has an additional dependence on Y. $\square$

Corollary 38 together with the following two claims imply Theorem 35.

Claim 39. $\left|g_k^{-1}(1)\right| \geq 2^{|X\cup Y|-5(r+1)(k+1)}$

Let $R(X,Y) = R'(X\backslash B, Y) \wedge R''(X\backslash A, Y)$ be a planar pseudo-rectangle of order $m = \Omega(n)$ on X according to Corollary 38.

Claim 40. *If $R(X,Y)$ is a portion of $g_k(X,Y)$, then* $\left|R^{-1}(1)\right| \leq 2^{|X\cup Y|-m/(4(k+1)^2)}$.

Proof of Claim 39. Let $X \supseteq X' = \bigcup_{d\in[0,k]} X'_d$, where X'_d is a set of 5 variables $\{x_v : v \in I_d\}$ where $I_d = \{v \in [1,n]^{k+1} : v_d \in [2,6] \text{ and } v_{d'} = 1, \text{ for } d' \neq d\}$. Let $Y' \subseteq Y$ be the variables in the same positions in $[1,n]^{k+1}$ as X'. Fix any assignment ρ to $(X\cup Y) - (X' \cup Y')$; we show that it can always be completed to some assignment such that for each d, $H_d(X,Y) \equiv 0 \pmod 3$, thus satisfying $g_k(X,Y)$. Because $|X' \cup Y'| = 5(r+1)(k+1)$, the claim follows.

Fix the variables in Y' so that for each d, the variables in X'_d are active for d. This is possible since the various I_d are disjoint. Fix any d and write $X'_d = \{z_1, \ldots, z_5\}$. Substituting the values according to ρ, $H_d(X,Y)$ can be written as $\sum_i u_i z_i + b$, for some constants $u_1, \ldots, u_5$, and b. It is not too hard to show that there is a setting of the z_i's such that $\sum_i u_i z_i + b = 0 \pmod 3$. (See [Tha98] for details.) Repeating this for all d's gives a satisfying assignment for $g_k(X,Y)$. $\square$

Proof of Claim 40. By definition, there exists a d and an index set $J \subseteq [1,n]$ of size m such that $A = \{a_j : j \in J\}$ and $B = \{b_j : j \in J\}$, and a_j and b_j correspond to points in the j^{th} d-plane for each $j \in J$. For any setting π of 0-1 values to Y, we say that a pair (a_j, b_j) for $j \in J$ is *good for* π if both a_j and b_j are active for d under π; let G_π be the set of pairs good for π. Fix an arbitrary assignment ρ, consistent with π, to the variables outside G_π. Let Γ_ρ be the satisfying assignments of $R(X,Y)$ consistent with ρ. Because $R(X,Y)$ is a portion of $g_k(X,Y)$, every assignment in Γ_ρ satisfies $H_d(X,Y) \equiv 0 \pmod 3$. Under this assignment, the polynomial $H_d(X,Y)$ reduces to $(\sum_{(a_j,b_j)\in G_\pi} u_j a_j b_j) + c$, where $u_j \in \{-1,1\}$ for all j and $c \in GF(3)$ are constants. Applying Lemma 27 to the $|G_\pi| \times |G_\pi|$ matrix of full rank whose diagonal entries are the u_j's, we obtain $|\Gamma_\rho| / 2^{2|G_\pi|} \leq 2^{-|G_\pi|}$.

For a randomly and uniformly chosen π, and any fixed pair (a_j, b_j), the probability that (a_j, b_j) is good is $1/(k+1)^2$. Therefore, by Chernoff's bound, $\Pr[|G_\pi| \leq m/(3(k+1)^2)] \leq e^{-2m/(9(k+1)^2)}$. Since ρ was picked uniformly given π, we can finish the proof as follows:

$$\left|R^{-1}(1)\right| / 2^{|X\cup Y|} \leq e^{-2m/(9(k+1)^2)} + 2^{-m/(3(k+1)^2)} \leq 2^{-m/(4(k+1)^2)}. \quad \square$$

Acknowledgements. We thank Amir Shpilka and Xiaodong Sun for pointing out some errors in an earlier version of this paper.

References

[Abr90] Karl R. Abrahamson. A time-space tradeoff for Boolean matrix multiplication. In *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pages 412–419, St. Louis, MO, October 1990. IEEE.

[Abr91] Karl R. Abrahamson. Time–space tradeoffs for algebraic problems on general sequential models. *Journal of Computer and System Sciences*, 43(2):269–289, October 1991.

[Ajt99b] Miklós Ajtai. A Non-linear Lower bound for Boolean Branching Programs preprint,1999

[Ajt99a] Miklós Ajtai. Determinism versus Non-Determinism for Linear Time RAMs with Memory Restrictions In *Proceedings of 31st ACM Syposium on Theory of Computing*, Atlanta, GA, May, 1999.

[AM88] Noga Alon and Wolfgang Maass. Meanders and their applications in lower bounds arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.

[AS92] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* John Wiley & Sons, 1992.

[BC82] Allan Borodin and Stephen A. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, May 1982.

[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.

[Bea91] Paul W. Beame. A general time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2):270–277, 1991.

[BF92] Laszlo Babai and Peter Frankl. *Linear Algebra Methods in Combinatorics with Applications to Geometry and Computer Science (Preliminary Version 2).* University of Chicago, 1992.

[BFMadH+87] Allan Borodin, Faith E. Fich, Friedhelm Meyer auf der Heide, Eli Upfal, and Avi Wigderson. A time-space tradeoff for element distinctness. *SIAM Journal on Computing*, 16(1):97–99, February 1987.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, October 1992.

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

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

[For97] Lance Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time-space tradeoffs for satisfiability. In *Proceedings, Twelfth Annual IEEE Conference on Computational Complexity*, pages 52–60, Ulm, Germany, 24–27 June 1997. IEEE Computer Society Press.

[HLP52] G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities.* Cambridge University Press, 1952.

[KW88] K. Kriegel and S. Waack. Lower bounds on the complexity of real-time branching programs. newblock *RAIRO Theoret. Inform. Appl.* 22:447-459, 1988.

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

[Pip79] Nicholas J. Pippenger. On simultaneous resource bounds. In *20th Annual Symposium on Foundations of Computer Science*, pages 307–311, San Juan, Puerto Rico, October 1979. IEEE.

[Raz91] A. A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In Lothar Budach, editor, *Fundamentals of Computation Theory: 8th International Conference, FCT '91*, volume 529 of *Lecture Notes in Computer Science*, pages 47–60, Gosen, Germany, September 1991. Springer-Verlag.

[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.

[Tha98] J. S. Thathachar. On separating the read-k-times branching program hierarchy. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, Dallas, TX, May 1998. To appear.

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

[Weg88] Ingo Wegener On the complexity of branching programs and decision trees for the clique functions *JACM*, 35:461-471, 1988.

[Yao88] A. C. Yao. Near-optimal time-space tradeoff for element distinctness. In *29th Annual Symposium on Foundations of Computer Science*, pages 91–97, White Plains, NY, October 1988. IEEE.