Journal of Computer and System Sciences **63**, 542–572 (2001)
doi:10.1006/jcss.2001.1778, available online at http://www.idealibrary.com on IDEAL®

Time–Space Tradeoffs for Branching Programs[1]

Paul Beame[2]

Department of Computer Science and Engineering, University of Washington,
Seattle, Washington 98195
E-mail: beame@cs.washington.edu

T. S. Jayram[3]

IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120
E-mail: jayram@almaden.ibm.com

and

Michael Saks[4]

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08854
E-mail: saks@math.rutgers.edu

Received March 3, 1999; Revised August 1, 2000

We obtain the first non-trivial time–space tradeoff lower bound for functions $f\colon \{0,1\}^n \to \{0,1\}$ on general branching programs by exhibiting a Boolean function f that requires exponential size to be computed by any branching program of length $(1+\varepsilon)\,n$, for some constant $\varepsilon > 0$. We also give the first separation result between the syntactic and semantic read-k models (A. Borodin *et al.*, *Comput. Complexity* **3** (1993), 1–18) for $k > 1$ by showing that polynomial-size semantic read-twice branching programs can compute functions that require exponential size on any semantic read-k branching program. We also show a time–space tradeoff result on the more general R-way branching program model (Borodin *et al.*, 1993): for any k, we give a function that requires exponential size to be computed by length kn q-way

[1] A preliminary version of this paper appeared in "Proceedings of 39th annual IEEE Symposium on Foundations of Computer Science, November, 1999," pp. 254–263.

[2] Research supported by NSF Grant CCR-9303017.

[3] This work was part of this author's Ph.D. dissertation, completed at University of Washington and was supported by NSF Grant CCR-9303017. The author was formerly known as Jayram S. Thathachar.

[4] Research supported by NSF Grant CCR-9700239. This work was done in part while on sabbatical at University of Washington.

0022-0000/01 $35.00

branching programs, for some $q = q(k)$. This result gives a similar tradeoff for RAMs, and thus provides the first nontrivial time–space tradedoff for decision problems in this model.

1. INTRODUCTION

One of the long-standing open questions of complexity theory is whether polynimial-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 bounds? Despite significant progress given by the recent time-space tradeoff lower bounds for SAT by Fortnow [For97] and its subsequent improvements [LV99, FvM00], this question remains unsolved.

One natural model for studying this question is the Boolean branching program model, which simultaneously captures time and space in a clean combinatorial manner. In this model, a program for computing a function $f(x_1, x_2, ..., x_n)$ 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. Each node is also labeled by a (possibly null) 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 and outputting the sequence of output values at those nodes. 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 the 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, 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 multivariate 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, athough time-space tradeoffs for element distinctness on the structured *comparison branching problem* 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 intersting 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 $\mathscr{F}_q = (f_n^{(q)} : n$ 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 $\mathscr{F}_q$ require size 2^S where $S = \Omega(n(\log q)^{1-\varepsilon})$, for any $\varepsilon > 0$. In particular, if one chooses $q = \Theta(n)$ then any branching program of length $o(n \log \log n)$ require size at least 2^S where $S = n \log^{1-\varepsilon} n$ for any $\varepsilon > 0$. As noted by Borodin and Cook [BC82], this result gives a similar tradeoff for RAMs, and thus provides the first nontrivial time-space tradeoff for decision problems in this model.

With respect to the branching program model this result is not entirely satisfying, 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 more interesting model of 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.[5]

The proofs use a variety of techniques, some of which extend techniques of [BRS93, Tha98]. First, we show that if 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.

Each of the inner $\bigwedge$ terms is called a *decision forest*, and each must accept a subset of the 1's of f. We then show that the set of inputs accepted by a decision forest whose component trees are shallow can be decomposed into not too many *pseudo-rectangles*, which are objects related to, but more general than, the rectangles that are routinely analyzed in communication complexity. We derive two

[5] Since the publication of a preliminary version of our results, Ajtai [Ajt99b, Ajt98, Ajt99a] used related techniques to exhibit an explicit family of boolean functions for which any linear size branching program must have exponential size. Subsequently, Ajtai's bounds were quantitatively improved, and also extended to randomized computation [BSSV00].

such decomposition results for decision forests. In the first (and easier) decomposition result, the upper bound on the number of pseudo-rectangles is non-trivial only in the case that the underlying domain of the functions is large. To get a non-trivial bound for the boolean domain, we prove a second decomposition result which applies to a very restrictive situation, when the decision forest consists of two trees, each of depth slightly more than $n/2$. The proof uses an interesting entropy argument.

Combining the decompositions associated to each decision forest in the $\bigvee$ for f gives a decomposition of the original function f into pseudo-rectangles, where we can bound the number of pseudo-rectangles in the decomposition. Expanding on arguments in [BSR93, Tha98], we show that certain explicit functions, can not be decomposed into such a small number of pseudo-rectangles. In the large domain case, this will give exponential size lower bounds for linear depth branching programs that compute these functions. In the boolean domain case, the exponential size lower bounds are derived only for branching programs of depth less than $1.0178n$.

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 [Weg86, BHST87, Weg88, Raz91, SS93, G9́7]).

(Another restricted class of branching programs that has been studied is the class of *oblivious* branching programs. An oblivious branching program is a leveled program where all nodes at the same level are labeled by the same variable (so that the variable read at a particular time step is independent of the path followed). 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 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 program 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.[6] (It is easy to bserve that there is

[6] The new results of [Ajt99b] mentioned earlier yield such bounds.

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 the 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 show that for each $k \geqslant 1$, the syntactic read-k model is more powerful than the syntactic read-$(k+1)$ model. These functions are exponentially hard for syntactic read-k, and while they also seem to be hard for semantic read-k, we are able to construct a modified version of these functions so as to make them easy for semantic read-twice while still retaining hardness for syntactic read-k.

2. NOTATION

For an integer n, $[n]$ denotes the set $\{1, ..., n\}$.

2.1. Variables, Assignments, and Functions

Throughout, X is a set of variables (usually $\{x_1, ..., x_n\}$) 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 *decision function* f over X is a function mapping D^n to $\{0, 1\}$. In the case $|D| = 2$, f is a *Boolean function.* Given a partial input ρ, the *restriction* of f by ρ, $f\lceil_\rho$ is the function on $X' = X \backslash vars(\rho)$ 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 with value b.

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

If $A_1, A_2, ..., A_t$ are subsets of X we say that f is $(A_1, A_2, ..., A_t)$-free if f can be written as $\bigwedge_{i=1}^{t} g_i$, where each g_i does not depend on the values of variables in A_i. If A_1, A_2 are disjoint subsets of X, we say that f is a *pseudo-rectangle* with associated sets A_1, A_2 if f is (A_1, A_2)-free. If $|A_1| = |A_2| = \ell$ and $|A_2| \geqslant \ell$, we say that f is a pseudo-rectangle to *order* ℓ or an ℓ-pseudo-rectangle. If ℓ is not an integer then an ℓ-pseudo-rectangle means a $\lceil \ell \rceil$- 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 $\Gamma_1 \times \Gamma_2$ where Γ_i is a set of assignments to Y_i. A function f is a pseudo-rectangle with associated sets A_1, A_2, if and only if for each 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 Forests

Since we are only concerned here with the computation of decision functions (those whose output is a single bit), 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 a branching program can be modified to one without free arcs with no change in length and 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 $V_0, V_1, ..., V_d$ 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 $0 \leqslant i \leqslant d$. 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 nodes 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 forest* is an $\bigwedge$ of decision trees. More precisely, a (r, ε) decision forest P over D is a collection $T_1, ..., T_r$ of decision trees on D^n such that each tree has height at most $\lceil \varepsilon n \rceil$. The function computed by P is $\bigwedge_{i=1}^{r} T_i$. A decision forest 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.

THEOREM 1. *Let $k, n, s \in \mathbf{N}$ with $n \geqslant k(k+1)^2\, 2^{2(k+4)}$. Let f be a boolean function on D^n for some finite set D. If f can be computated 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 a βn-pseudo-rectangle and $m \leqslant 2^{4(k+2)\,\beta n}(2s)^{k(k+1)\,2^{k+4}}$ for $\beta = 1/2^{k+2}$.

This theorem is quite general but, as we will see, the upper bound that it provides on m is so large, that it only yields 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 $(1+\varepsilon)\, n$. In order to state this decomposition theorem it will be convenient to make the following technical definition. For $\varepsilon, \delta \in (0, 1/4)$ define

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

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

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

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

where $m \leqslant s\,2^{\gamma n}$, $|f_0^{-1}(1)| \leqslant s2^{(1+b(\varepsilon,\delta)/2-\gamma/2)^n}$, *and, for* $i \geqslant 1$, *each* f_i *is a* $(1-\varepsilon-\delta)\,n/2$-*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, assuming that f is computable by a decision tree of length kn and size s, 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 $(r, k/r)$ decision forest where $r \geqslant k$ can be chosen arbitrarily. 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 the depth $(k/r)\,n$ of each decision tree is sufficiently small compared to n. Thus on any input x, each constituent decision tree of the $(r, k/r)$ forest 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 five sections we give all the details necessary for the obtaining these lower bounds. We prove the decomposition into decision forests in Section 5; give the detailed proofs of Theorem 1 and 2 in Sections 6 and 7. In Sections 8 and 9, 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 decision function f on domain D^n and $\beta \in [0, 1]$, let

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

The reader should keep in mind that $\eta(f) \in [0, 1]$ and that "small" values of $\eta(f)$ indicate that $f^{-1}(1)$ is "big". Also, for each $\beta \in [0, 1]$, $\Psi_f(\beta) \in [0, 1]$ and a lower bound on $\Psi_f(\beta)$ implies an upper bound on the size of the largest βn-pseudo-rectangle contained in f.

We use Theorem 1 to get the following lower bound on the size of any depth kn branching program for computing f, in terms of these two parameters:

COROLLARY 3. *Let $n, k \in \mathbf{N}$ with $n \geqslant (k+1)\, 2^{k+4}$. 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*

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

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

Proof. Apply Theorem 1 to f to get the specified decomposition $f = \bigvee_{i=1}^{m} f_i$. Then each f_i is a portion of f that is an βn-pseudo-rectangle with $\beta = 1/2^{k+2}$ and one of them must have $|f_i^{-1}(1)| \geqslant |f^{-1}(1)|/m$. This implies $m \geqslant |f^{-1}(1)|/|D|^{1-\Psi_f(\beta)\,n} = |D|^{(\Psi_f(\beta)-\eta(f))\,n}$.

Combining this with the upper bound on m given by Theorem 1 gives

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

and the corollary follows. ∎

Remark. If all 0-certificates of f have size at least $t = (1-\delta)\,n$, then $\eta(f) \leqslant \delta + 1/n$, 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.

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

If $|D| \leqslant 2^{2(k+2)}$ then Corollary 3 is not useful. This is because for any function f one can construct a βn-pseudo-rectangle that is a portion of f and is 1 on a large fraction of $f^{-1}(1)$: Let A_1 and A_2 be arbitrary disjoint variable subsets of size $\lceil \beta n \rceil$ and let $A_0 = X - A_1 - A_2$. Among the elements of $f^{-1}(1)$, choose the most popular assignment σ_1 to A_1. So at least $f^{-1}(1)\,|D|^{-\lceil \beta n \rceil}$ points of $f^{-1}(1)$ are consistent with σ_1. Let g_1 accept input σ if and only if the input σ', obtained by modifying σ on A_1

to σ_1, is in $f^{-1}(1)$; let g_2 accept σ if and only if σ agrees with σ_1 to A_1. Because g_1 does not depend on A_1 and g_2 does not depend on A_2, $g = g_1 \wedge g_2$ is a pseudo-rectangle. Moreover it is a portion of f accepting at least $|f^{-1}(1)|\,|D|^{-\lceil \beta n \rceil} \geqslant |f^{-1}(1)|\,|D|^{-2\beta n}$ inputs by the bound on n. Thus $\Psi_f(\beta) - \eta(f) \leqslant 2\beta$ and for $|D| \leqslant 2^{2(k+2)}$ the size lower bound provided by Corollary 1 is less than 1.

In the Boolean case, we use Theorem 2 to derive:

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

$$s \geqslant \tfrac{1}{2}\,2^{\alpha n},$$

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

Proof. 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 $(1+\varepsilon)\,n$. For any $\gamma \in (0, 1)$, we can find $f_0, \ldots, f_m$ as in Theorem 2. For each $i \geqslant 1$, f_i is a βn-pseudo-rectangle for $\beta = (1-\varepsilon-\delta)/2$. By the definition of $\Psi_f(\beta)$, for each $i \geqslant 1$, $|f_i^{-1}(1)| \leqslant 2^{(1-\Psi_f(\beta))\,n}$, so, using the bounds of Theorem 2 on m and $|f_0^{-1}(1)|$, we have

$$|f^{-1}(1)| \leqslant s 2^{(\gamma + 1 - \Psi_f(\beta))\,n} + s 2^{(1 + b(\varepsilon, \delta)/2 - \gamma/2)\,n}$$

Choosing $\gamma = 2\Psi_f(\beta)/3 + b(\varepsilon, \delta)/3$ to make the two summands equal, and rewriting the inequality to lower bound s yields the desired lower bound. ▮

If the value α in the conclusion of the corollary can be bounded below by 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 $|f^{-1}(1)|$ 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 be sufficiently close to n.

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: $GF(q)^n \to \{0, 1\}$ to be true if and only if for any input σ (viewed as a vector of length n), $\sigma^T M \sigma = 0 (\text{mod}\; q)$. 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 $n = 2^k$, the $n \times n$ Sylvester matrix N 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 modulo 2.

For any odd prime power q, we can interpret the Sylvester matrix as a quadratic form over $GF(q)^n$. Sylvester matrices are examples of Generalized Fourier Transform (GFT) matrices (see [BSR93]). For any finite Abelian group G, let G^* be the set of multiplicative characters of G, i.e., functions $\chi: G \rightarrow GF(q)^*$ that satisfy $\chi(g_1 g_2) = \chi(g_1)\,\chi(g_2)$ for any $g_1, g_2 \in G$. Provided that q is relatively prime to $|G|$, it is well known (see, e.g., [BRS93]), that there are $|G|$ distinct characters and that they are linearly independent when viewed as vectors over $GF(q)$. Let $N = N_{G, G^*}$ be the matrix in which the (g, χ)th element equals $\chi(g)$, for all $g \in G$ and $\chi \in G^*$. Sylvester matrices of dimension $2^k \times 2^k$ can be shown to be special cases of GFT matrices corresponding to the additive group of $GF(2)^k$.

If N is any square matrix, the *modification of* N, $N^{[0]}$, is the matrix obtained by setting the diagonal entries of N to 0.

The following lemma, which we prove in Section 8, will allow us to apply Corollary 3 and get branching program lower bounds in the case D is large.

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 or $QF_{M^{[0]}}$ to the domain D^n, then $\Psi_f(\beta) \geqslant \beta^2$.*

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

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

THEOREM 6. *There is a constant $c > 0$ such that the following holds. Let k be a positive integer and $\varepsilon \in (0, 1)$. Let q be a prime power that satisfies $\log\log q \geqslant \frac{c}{\varepsilon} k$, and let n be an integer satisfying $n \geqslant 2^{2k+5}$. Let $N^{[0]}$ be the modification of an $n \times n$ GFT matrix over $GF(q)$. Then any non-deterministic branching program for $QF_{N^{[0]}}$ of length kn requires size at least 2^S where $S = n \log^{1-\varepsilon} q$.*

Proof. The function $f = QF_{N^0}$ is a function on D^n where $|D| = q$. Let $\beta = 1/2^{k+2}$. By Lemma 5 and the remark following Corollary 3, $\Psi_f(\beta) - \eta(f) \geqslant \beta^2 - \frac{2}{n}$, which is at least $\beta^2/2$ by the hypothesis on n. For c large enough (independent of k), the hypothesis on q implies $q^{\beta/4} > 2^{4(k+1)}$, which implies that $q^{\Psi_f(\beta)-\eta(f)}/2^{4(k+1)\beta} \geqslant q^{\beta^2/4}$. Corollary 3 now implies a size bound of the form $q^{n\beta^2/(4k(k+1)\,2^{k+4})}$. Using the hypothesis on q (with c large enough), this is lower bounded by $2^{n(\log q)^{1-\varepsilon}}$ as required. ∎

It is illuminating to formulate a version of the above theorem in which the parameters q and k are chosen to depend on n. Let c be the constant of the above theorem. Given n and ε, let $k = \lfloor \frac{\varepsilon}{c} \log\log n \rfloor$ and let q be the smallest prime greater than or equal to n (so $q < 2n$.) Then the hypothesis of the above theorem is satisfied and we obtain:

COROLLARY 7. *Let N be an $n \times n$ Sylvester matrix and consider the associated function $QF_{N^{[0]}}$ defined on a domain D of size $q, n \leqslant q < 2n$. For any $\varepsilon > 0$ 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 $S = \Omega(n \log^{1-\varepsilon} n)$.*

To prove lower bounds in the case that D is a boolean domain, we consider the function BQF_M, where M is a Sylvester matrix. In Section 9, we will prove:

Lemma 8. *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 Lemma 5 and 8 together with Corollary 4 we obtain:

Theorem 9. *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 $\varepsilon = 0.0178n$, that $\alpha > 0$. By Lemma 8, 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, $\eta(f) \leqslant \frac{1}{n} + \delta$ which is also $o(1)$ in n. Therefore by Lemma 5, $\Psi_f(\beta) \geqslant \beta^2$, we get: $\alpha = \frac{1}{3}\,(\frac{(1-\varepsilon)^2}{4} - \varepsilon - (1+\varepsilon)\, H(\frac{2\varepsilon}{1+\varepsilon})) + o(1)$. It can be checked that for $\varepsilon \leqslant 0.0178$, 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. ∎

5. DECOMPOSITION INTO DECISION FORESTS

Lemma 10. *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 $r \geqslant k$, f can be expressed as:*

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

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

Furthermore, (i) *if the branching program is deterministic then each of the decision forests 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 forests is oblivious.*

Proof. Let B be any branching program of size s computing f of length $d \leqslant kn$. As mentioned in Section 2.2, there is a leveled branching program B' of length d with at most s nodes per level that also computes f. 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 $1 \leqslant i \leqslant r-1$, define $l_i = \lceil \frac{id}{r} \rceil$. Note that $l_1 < \cdots < l_{r-1} < d$ divides the integral $[0, d]$ into r intervals each of size at most $\lceil \frac{d}{r} \rceil \leqslant \lceil \frac{kn}{r} \rceil$. An input is accepted by P if and only if 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 forest.

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

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 10.)

6. TRADEOFFS OVER LARGE DOMAINS AND FOR OBLIVIOUS BRANCHING PROGRAMS

In this section we prove Theorem 1 which says that every function computable by a branching program of suitably small depth and size can be decomposed as the $\bigvee$ of not too many pseudo-rectangles of large order. We will use Lemma 10 to write f as a $\bigvee$ of functions that are each computed by shallow decision forests. So it will suffice to find a suitable decomposition of functions computed by a shallow decision forest into pseudo-rectangles. To do this we will use the following combinatorial lemma which will allow us to find for a given decision forest and input σ, two large sets A_1 and A_2 so that on input σ, each decision tree in the forest looks at no variables in A_1 or no variables in A_2. This lemma is a generalization of a lemma from [Tha98].

6.1. A Combinatorial Lemma

LEMMA 11. *Let k, p, n be positive integers with $p=(k+1)\,2^{k+4}$ and let $n \geqslant 8kp$. 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 disjoint 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 12. *Let $\mathscr{F}$ denote a family of non-empty sets each containing at most m elements. Let $X = \bigcup_{Y \in \mathscr{F}} Y$ and suppose $|X| = n$. Let $\lambda = \sum_{Y \in \mathscr{F}} |Y|/n$. Then, X has disjoint sets S and T, each of size at least $(1-\delta)\,2^{-\lambda} n$, where $\delta = \sqrt{\lambda m\, 2^{1+\lambda/n}}$, such that each $Y \in \mathscr{F}$ is disjoint from S or from T.*

The quantity λ can be interpreted naturally as the average number of times each element in X is covered in $\mathscr{F}$. The lemma is a straightforward strengthening of a lemma of [Tha98], in which λ is replaced by the maximum number of times each element of X is covered.

Using Lemma 12, we prove Lemma 11. If $|\bigcap_i S_i| \geqslant 2\lceil \beta n \rceil$, 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 $\mathscr{F}$ to consist of S'_i for $1 \leqslant i \leqslant kp$, and singleton sets $\{j\}$ for each $j \notin \bigcup_i S'_i$ so that the union of $\mathscr{F}$ is $[n]$. The number of singleton sets is at most $2\lceil \beta n \rceil - 1$. Therefore, the sum of the sizes of the sets in $\mathscr{F}$ is at most $2\lceil \beta n \rceil - 1 + kp\lceil n/p \rceil \leqslant (k+2\beta)\,n + 1 + kp \leqslant (k+2\beta+1/4)\,n$, by the hypothesis on n. In the terminology of Lemma 12, we have $\lambda \leqslant (k+2\beta+1/4)\,n/n = k+2\beta+1/4 \leqslant k+1/2$. Moreover, setting $m = \lceil n/p \rceil$, it follows that

$\lambda m \leqslant (k+2\beta+1/4)(n/p+1) \leqslant (k+1)\, n/p$. Applying Lemma 12 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

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

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 $[kp]$ 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 11.

To prove Lemma 12, we use two elementary inequalities due to Chebyschev. The first appears in any elementary probability text, and the second can be found, for example, in [HLP52, Theorem 43, p. 43]).

Proposition 13. *Let $E[\mathbf{Z}_x]$ denote the mean and $var[\mathbf{Z}_x]$ denote the variance of a random variable $\mathbf{Z}_x$. Then, for any $\delta > 0$,*

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

Proposition 14. *Let $a_1, ..., a_N$ be a non-decreasing sequence and $b_1, ..., b_N$ be a non-increasing sequence of non-negative numbers. Then, $\sum_i a_i b_i \leqslant (\sum_i a_i)(\sum_i b_i)/N$.*

Proof of Lemma 12. Randomly color each set $Y \in \mathscr{F}$ red or blue uniformly and independently. Call an element red (resp. blue) if all the sets containing it are red (resp. blue), and let S (resp. T) be the set of red (blue) elements. Since every element of X occurs in at least one set, it follows that S and T are disjoint. Moreover, for each $Y \in \mathscr{F}$, 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 $\sum_{x \in X} d_x = \sum_{Y \in \mathscr{F}} |Y| = \lambda n$. 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 $E[\mathbf{Z}_x] = \mathbf{Prob}[\mathbf{Z}_x = 1] = 2^{-d_x}$. Let $\mathbf{Z} = \sum_x \mathbf{Z}_x$ and observe that $\mathbf{Z} = |S|$. Using the arithmetic-geometric mean inequality, we obtain

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

Next, we show that $\mathbf{Z}$ is close to its expected value with high probability. The variance of $\mathbf{Z}$ is given by

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

where $\operatorname{cov}(\mathbf{Z}_x, \mathbf{Z}_y) = E[\mathbf{Z}_x \mathbf{Z}_y] - E[\mathbf{Z}_x]\, E[\mathbf{Z}_y]$ denotes the covariance of $\mathbf{Z}_x$ and $\mathbf{Z}_y$. Consider the first term in the right hand side of (2). For any x, $\mathbf{Z}_x$ is a Bernoulli random variable, so $var[\mathbf{Z}_x] = E[\mathbf{Z}_x](1 - E[\mathbf{Z}_x]) \leqslant E[\mathbf{Z}_x]$, implying that

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

To bound the second term in the right hand side of (2), observe that if no set Y contains both x and y, the events $\mathbf{Z}_x$ and $\mathbf{Z}_y$ are independent implying that $\operatorname{cov}(\mathbf{Z}_x, \mathbf{Z}_y) = 0$. 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 pair, $\operatorname{cov}(\mathbf{Z}_x, \mathbf{Z}_y) \leqslant E[\mathbf{Z}_x \mathbf{Z}_y] \leqslant E[\mathbf{Z}_x] = 2^{-d_x}$. Therefore,

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

The last term above can be bounded as follows: Order the x's so that the sequence $\{d_x\}$ is non-decreasing. Proposition 14 can be applied to the sequences $\{d_x\}$ and $\{2^{-d_x}\}$ to give

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

The equality follows from $(\sum_X d_x)/n = \lambda$, and $\sum_x 2^{-d_x} = E[\mathbf{Z}]$. Substitute the bounds in (3) and (4) in (2). We obtain

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

where the last inequality holds because each $x \in X$ occurs in at least one set, implying that $\lambda = \sum_x d_x / n \geqslant 1$. Using Proposition 13, we have

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

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

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

Similarly, we obtain $\mathbf{Prob}[|T| < (1-\delta) \cdot E[\mathbf{Z}]] < 1/2$. Thus with positive probability both S and T have size at least $(1-\delta)\, 2^{-\lambda}$. 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 11 yields a time–space tradeoff for oblivious branching programs.

THEOREM 15. *Let $k, n, s \in \mathbf{N}$ with $n \geqslant k(k+1)\, 2^{k+7}$. 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 $m \leqslant s^{k(k+1)\, 2^{k+4}-1}$ and each f_i is a βn-pseudo-rectangle for $\beta = 1/2^{k+2}$.

Proof. Let f be as hypothesized. Apply lemma 10 with $r = kp$ and $p = (k+1)\,2^{k+4}$, to get a decomposition of f as a $\bigwedge P_i$ of s^r oblivious decision forests each consisting of r trees of depth at most $\lceil n/p \rceil$. Consider one such forest $P_i = T_1, \ldots, T_r$ and let S_i be the variables that don't appear in tree T_i, so $|S_i| \geqslant \lfloor n(1-1/p) \rfloor$. By Lemma 11, for $p = (k+1)\,2^{k+4}$ and $\beta = 1/2^{k+2}$, there are sets A_1, A_2 of size at least βn and a partition of $[r]$ into I_1, I_2 such that for $j = 1, 2$, $g_j = \bigwedge_{i \in I_j} T_i$ does not depend on A_j. Thus P_i is, in fact, a βn-pseudo-rectangle. ▌

In the same way that Theorem 1 was used to deduce Corollary 3, we can use Theorem 15 to deduce the following lower bound result for oblivious branching programs. (Again, something similar is implicit in [BRS93]).

Corollary 16. *Let $n, k \in \mathbf{N}$ with $n \geqslant k(k+1)\,2^{k+7}$. 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*

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

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. 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 10, we can write $f = \bigvee_{i=1}^{m} P_i$ where $m \leqslant s^{kp-1}$ and each of the P_i was a βn-pseudo-rectangle for appropriate β. For arbitrary decision forests this need not be true; instead we show that each decision forest can be written as an $\bigvee$ of not too many βn-pseudo-rectangles.

Lemma 17. *Let k, p, n be positive integers with $p = (k+1)\,2^{k+4}$, and $n \geqslant 8kp$. Let $\beta = 1/2^{k+2}$. Let $P = (T_1, \ldots, T_{kp})$ be a $(kp, \frac{1}{p})$ decision forest. Then P can be written as*

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

where $t \leqslant 2^{4\beta(k+2)\,n + kp}$, 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 $s^{kp-1} 2^{4\beta(k+2)\,n + kp}$ functions that are each βn-pseudo-rectangles, as required to prove the theorem. So it remains to prove the lemma.

Proof. Let $T_1, \ldots, T_{kp}$ be the decision trees of 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

$\{1, 2, ..., kp\}$. Call such a quadruple *eligible*. Note that the number of eligible quadruples is bounded above by $2^{kp}\binom{n}{\beta n}^2 \leqslant 2^{kp}2^{2H(\beta)\,n}$, using the standard inequality for binomial coefficients given below in Proposition 25. Since $\beta \leqslant 1/2$, $-\beta \log_2 \beta \geqslant -(1-\beta)\log_2(1-\beta)$ and so $H(\beta) \leqslant -2\beta \log_2 \beta$. Thus the number of eligible quadruples is at most $2^{kp+4(k+2)\,\beta n}$.

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 11 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 $I \subseteq \{1, 2..., kp\}$, 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. ▮

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 $(1+\varepsilon)\,n$. We first apply Lemma 10 with $r=2$ and $k=1+\varepsilon$. Thus we can write $f=\bigvee_{j=1}^{s} P_j$ where each P_j is a $(2, (1+\varepsilon)/2)$-decision forest, and we may assume that P_j is deterministic since we are only considering deterministic branching programs. We will prove the following:

Lemma 18. *Let* $P=(T_1, T_2)$ *be a deterministic* $(2, \frac{1+\varepsilon}{2}\,n)$*-decision forest with* $\varepsilon \in (0, 1)$ *and suppose that the smallest* 1*-certificate of P is of size at least* $(1-\delta)\,n$. *If* $\delta+\varepsilon \leqslant 1/4$ *then, for any* $\gamma \in (0, 1)$, *we can write P as:*

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

where $t \leqslant 2^{\gamma n}$, *each* h_i *is a* $\lfloor (1-\varepsilon-\delta)\,n/2 \rfloor$*-pseudo-rectangle and* $|h_0^{-1}(1)| \leqslant 2^{(1+b(\varepsilon,\delta)/2-\gamma/2)\,n}$.

Using the lemma, we easily complete the proof of Theorem 2. Let $P_1, ..., P_s$ be the decision forests in the representation of f in Lemma 10. Observing that any 1-certificate of a P_j must also be a 1-certificate of f, we apply Lemma 18 to each P_j. For $0 \leqslant i \leqslant 2^{\gamma n}$, 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 $1 \leqslant j \leqslant s$ and $1 \leqslant i \leqslant 2^{\gamma n}$, as required to prove the Theorem.

So it remains to prove the lemma. Let $P=(T_1, T_2)$ be a decision forest satisfying the hypotheses of the lemma. We may assume that every path in T_1 and T_2 has size exactly $\frac{1+\varepsilon}{2}\,n$. As in the previous proof, we define $S_i(\sigma)$, for input σ and $i=1, 2$, to be the 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 $\frac{1-\varepsilon}{2}n$. 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 S. 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 $\frac{1-\varepsilon}{2}n$. Observe that each input σ belongs to exactly one of the sets $\mathrm{Miss}(S_1, S_2)$ for $(S_1, S_2) \in Q$, namely $\mathit{Miss}(S_1(\sigma), S_2(\sigma))$ and if σ is accepted by P it 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 dividing their overlap evenly, we can find two disjoint sets U_1 and U_2 of size at least $\lfloor\frac{(1-\varepsilon)n - |S_1 \cap S_2|}{2}\rfloor$, such that $U_1 \subseteq S_1$ and $U_2 \subseteq S_2$. Therefore g_{S_1,S_2} is a $\lfloor\frac{(1-\varepsilon)n-|S_1\cap S_2|}{2}\rfloor$-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 $(S_1, S_2) \in Q$ such that $|S_1 \cap S_2| \leqslant \delta n$, we have $P = \bigvee_{(S_1,S_2)\in P} g_{S_1,S_2}$. As noted, each of these terms is a $\lfloor\frac{(1-\varepsilon-\delta)}{2}\rfloor$-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^{vn} terms for some small v).

Next we divide the terms of the $\bigvee$ into two parts depending on the size of $\mathrm{Miss}(S_1, S_2)$. Let $0 \leqslant \gamma \leqslant 1$. Call a pair $(S_1, S_2) \in R$ *common* if $\mathrm{Miss}(S_1, S_2) \geqslant 2^{(1-\gamma)n}$, 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 for $(S_1, S_2) \in Q$, 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 $|g^{-1}_{\mathrm{rare}}(1)|$ 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 19. *Let $P = (T_1, T_2)$ be a $(2, \frac{1+\varepsilon}{2} n)$-decision forest with $\varepsilon \in (0, 1)$. Let the size of the smallest* 1*-certificate of P be at least $(1-\delta)\, n$ and suppose that $\delta + \varepsilon \leqslant 1/4$. Let $\gamma \in (0, 1)$ and let $A = \bigcup_{(S_1, S_2) \in R_{\text{rare}}} \text{Miss}(S_1, S_2)$. Then*

$$\log_2 |A| \leqslant \left(1 - \frac{\gamma - b(\varepsilon, \delta)}{2}\right) n. \tag{5}$$

This lemma is the crux of the argument. Its proof uses elementary information theory. We review some basic definitions and results. Let Ω 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 in 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 20. *If $\mathbf{C}$ is a random variable taking values in S then $H(\mathbf{C}) \leqslant \log_2 |S|$.*

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

PROPOSITION 21. *Let $\mathbf{C}$ be some random variable taking values in S and let A be an event. Let $S_{\mathbf{C}}(A)$ denote the set of $s \in S$ such that $\mathbf{Prob}[\mathbf{C} = s \mid A] > 0$. Then $H(\mathbf{C} \mid A) \geqslant \log_2 \mathbf{Prob}[A] - \max_{s \in S_{\mathbf{C}}(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s]$.*

Proof.

$$\begin{aligned} H(\mathbf{C} \mid A) &= -\sum_{s \in S} \mathbf{Prob}[\mathbf{C} = s \mid A] \log_2 \mathbf{Prob}[\mathbf{C} = s \mid A] \\ &\geqslant -\max_{s \in S_{\mathbf{C}}(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s \mid A] \\ &\geqslant -\max_{s \in S_{\mathbf{C}}(A)} \log_2 \frac{\mathbf{Prob}[\mathbf{C} = s]}{\mathbf{Prob}[A]} \\ &\geqslant \log_2 \mathbf{Prob}[A] - \max_{s \in S_{\mathbf{C}}(A)} \log_2 \mathbf{Prob}[\mathbf{C} = s] \quad \blacksquare \end{aligned}$$

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 $H(\mathbf{B}_1 \mid \mathbf{B}_2) = H(\mathbf{B}_1, \mathbf{B}_2) - H(\mathbf{B}_2) = \sum_{s \in S_2} H(\mathbf{B}_1 \mid \mathbf{B}_2 = s) \cdot \mathbf{Prob}[\mathbf{B}_2 = s]$. 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 22. *Let $\mathbf{B}_1, \mathbf{B}_2$ be random variables taking values on finite sets S_1 and S_2. Then*

1. $H(\mathbf{B}_1, \mathbf{B}_2) = H(\mathbf{B}_1 \mid \mathbf{B}_2) + H(\mathbf{B}_2 \mid \mathbf{B}_1) + I(\mathbf{B}_1, \mathbf{B}_2)$.
2. $H(\mathbf{B}_1, \mathbf{B}_2) \leqslant H(\mathbf{B}_1) + H(\mathbf{B}_2)$.
3. $H(\mathbf{B}_1 \mid \mathbf{B}_2) \leqslant \max_{s \in S_2} H(\mathbf{B}_1 \mid \mathbf{B}_2 = s)$.

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

PROPOSITION 23. 1. *If* $\mathbf{B}$ *and* $\mathbf{C}$ *are random variables such that* $\mathbf{B}$ *determines* $\mathbf{C}$ *then* (a) $H(\mathbf{C}) \leqslant H(\mathbf{B})$ *and* (b) $H(\mathbf{C} \mid \mathbf{B}) = 0$.

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* $I(\mathbf{C}_1, \mathbf{C}_2) \leqslant I(\mathbf{B}_1, \mathbf{B}_2)$.

PROPOSITION 24. *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*

$$H(\mathbf{B}_1, \mathbf{B}_2) + H(\mathbf{C}_1, \mathbf{C}_2) \leqslant H(\mathbf{B}_1) + H(\mathbf{B}_2) + H(\mathbf{C}_1 \mid \mathbf{C}_2) + H(\mathbf{C}_2 \mid \mathbf{C}_1).$$

Proof. By Proposition 22(1) and Proposition 23(2),

$$\begin{aligned} H(\mathbf{C}_1, \mathbf{C}_2) &= H(\mathbf{C}_1 \mid \mathbf{C}_2) + H(\mathbf{C}_2 \mid \mathbf{C}_1) + I(\mathbf{C}_2, \mathbf{C}_1) \\ &\leqslant H(\mathbf{C}_1 \mid \mathbf{C}_2) + H(\mathbf{C}_2 \mid \mathbf{C}_1) + I(\mathbf{B}_2, \mathbf{B}_1) \\ &= H(\mathbf{C}_1 \mid \mathbf{C}_2) + H(\mathbf{C}_2 \mid \mathbf{C}_1) + H(\mathbf{B}_1) + H(\mathbf{B}_2) - H(\mathbf{B}_1, \mathbf{B}_2). \quad \blacksquare \end{aligned}$$

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 25. *If* $k \leqslant n/2$, $\log_2(\sum_{i \leqslant k} \binom{n}{i}) \leqslant nH(k/n)$.

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

Proof of Lemma 19. 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 A by something like $2^{(1-\nu)n}$ where $\nu > 0$, i.e., that $\mu[A] \leqslant (1-\nu)\, n$. Note that A is defined to be the event that $(\mathbf{S}_1, \mathbf{S}_2) \in R_{\text{rare}}$, and so by Proposition 21, $\mu[A] \leqslant H(\mathbf{S}_1, \mathbf{S}_2 \mid A) + \max_{(S_1, S_2) \in R_{\text{rare}}} \mu[(\mathbf{S}_1, \mathbf{S}_2) = (S_1, S_2)]$. The second term is at most $(1-\gamma)\, n$ by the definition of R_{rare}. Thus we want to upper bound $H(\mathbf{S}_1, \mathbf{S}_2 \mid A) = H(\mathbf{S}_1 \mid \mathbf{S}_2, A) + H(\mathbf{S}_2 \mid \mathbf{S}_1, A) + I(\mathbf{S}_1, \mathbf{S}_2 \mid A)$ 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 23 to upper bound the third term by $I(\mathbf{P}_1, \mathbf{P}_2 \mid A)$. 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 24 and do not explicitly mention mutual information.

We now proceed with the proof by considering $H(\mathbf{P}_1, \mathbf{P}_2 \mid A) + H(\mathbf{S}_1, \mathbf{S}_2 \mid A)$. As noted, Proposition 21 implies $H(\mathbf{S}_1, \mathbf{S}_2 \mid A) \geqslant \mu[A] - (1-\gamma)\, n$. To apply the same proposition to $H(\mathbf{P}_1, \mathbf{P}_2 \mid A)$ 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 $\mu[(\mathbf{P}_1, \mathbf{P}_2) = (P_1, P_2)] \leqslant \delta n$. Consequently $H(\mathbf{P}_1, \mathbf{P}_2 \mid A) \geqslant \mu[A] - \delta n$ and so

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

On the other hand, by Proposition 24,

$$\begin{aligned} H(\mathbf{P}_1, \mathbf{P}_2 \mid A) + H(\mathbf{S}_1, \mathbf{S}_2 \mid A) &\leqslant H(\mathbf{P}_1 \mid A) + H(\mathbf{P}_2 \mid A) + H(\mathbf{S}_1 \mid \mathbf{S}_2, A) + H(\mathbf{S}_2 \mid \mathbf{S}_1, A) \\ &\leqslant (1+\varepsilon)\, n + H(\mathbf{S}_1 \mid \mathbf{S}_2, A) + H(\mathbf{S}_2 \mid \mathbf{S}_1, A) \end{aligned}$$

where the last inequality comes from Proposition 20 and the fact that the number of paths in each tree is $2^{(1+\varepsilon)\, n/2}$.

Since the two remaining terms are symmetric it remains to upper bound $H(\mathbf{S}_1 \mid \mathbf{S}_2, A)$. 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

$$\begin{aligned} H(\mathbf{S}_1 \mid \mathbf{S}_2, A) &\leqslant H(\mathbf{S}_2, \mathbf{I}, \mathbf{J} \mid \mathbf{S}_2, A) && \text{by Proposition 23(1)} \\ &\leqslant H(\mathbf{S}_2 \mid \mathbf{S}_2, A) + H(\mathbf{I} \mid \mathbf{S}_2, A) + H(\mathbf{J} \mid \mathbf{S}_2, A) && \text{by Proposition 22(2)} \\ &= H(\mathbf{I} \mid \mathbf{S}_2, A) + H(\mathbf{J} \mid \mathbf{S}_2, A) && \text{by Proposition 23(1)} \end{aligned}$$

Since A implies $|\mathbf{I}| \leqslant \delta n$, Proposition 20 and Proposition 25 imply that $H(\mathbf{I} \mid \mathbf{S}_2, A) \leqslant \log_2 \sum_{i \leqslant \delta n} \binom{n}{i} \leqslant nH(\delta)$. Now, by Proposition 22(3), $H(\mathbf{J} \mid \mathbf{S}_2, A) \leqslant \max_{S_2} H(\mathbf{J} \mid \mathbf{S}_2 = S_2, A)$. Given A, $|\mathbf{J}| \leqslant (\delta+\varepsilon)\, n$ and given $\mathbf{S}_2 = S_2$, $\mathbf{J}$ is contained in $\overline{S_2}$ which is a set of size $(1+\varepsilon)\, n/2$. Again, using Proposition 20 and Proposition 25 we conclude $H(\mathbf{J} \mid \mathbf{S}_2, A) \leqslant n \frac{1+\varepsilon}{2} H(\frac{2(\varepsilon+\delta)}{1+\varepsilon})$ (here we use the hypothesis that $\varepsilon + \delta \leqslant 1/4$) Thus

$$H(\mathbf{S}_1 \mid \mathbf{S}_2, A) \leqslant \left[\frac{1+\varepsilon}{2} H\left(\frac{2(\varepsilon+\delta)}{1+\varepsilon} \right) + H(\delta) \right] n$$

and so

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

Solving, we fine

$$\begin{aligned} \mu[A] &\leqslant \left(1 + \frac{1}{2} \left[\varepsilon + \delta + (1+\varepsilon)\, H\left(\frac{2(\varepsilon+\delta)}{1+\varepsilon} \right) + 2H(\delta) - \gamma \right] \right) n \\ &= \left(1 + \frac{b(\varepsilon, \delta) - \gamma}{2} \right) n, \end{aligned}$$

which is what we wanted to prove. ∎

8. PROOF OF LEMMA 5

For the first part of the lemma, we want to prove a lower bound on $\Psi_f(\beta)$ where f is a restriction of QF_M or $QF_{M^{[0]}}$ for a GFT matrix M. We start with a lemma that holds for arbitrary symmetric matrices over a finite field. For a $n \times n$ matrix M and any $\beta \in (0, 1)$, define $\Phi_M(\beta)$ to be $1/n$ times the minimum rank of any $\lceil \beta n \rceil \times \lceil \beta n \rceil$ 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 26. *Let M be a symmetric matrix over $GF(q)$ with q odd and let $D \subseteq GF(q)$ with $|D| \geqslant 2$. Let f be the restriction of QF_M to D^n. Then $\Psi_f(\beta) \geqslant \Phi_M(\beta)$ for $\beta \in [0, 1]$.*

Proof. Let g be a βn-pseudo-rectangle that is a portion of f. Let A_1 and A_2 be the associated sets and let g_1 be the function on $X - A_1$ and g_2 the function on $X - A_2$ such that $g = g_1 \wedge g_2$. Our goal is to show that $|g^{-1}(1)| \leqslant |D|^{(1-\Phi_M(\beta))\,n}$.

Let us denote assignments to X 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 $|D|^{2\lceil \beta n \rceil}$ 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 $|\Gamma| \leqslant |D|^{(2\lceil \beta n \rceil - \Phi_M(\beta))\,n}$; by summing over the $|D|^{n-2\lceil \beta n \rceil}$ choices of ρ, we get $|g^{-1}(1)| \leqslant |D|^{(1-\Phi_M(\beta))\,n}$ as required.

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

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

Substitute the values assigned by ρ into $(\rho, \sigma_1, \sigma_2)^T M (\rho, \sigma_1, \sigma_2)$. Because M is symmetric, we obtain a polynomial of the form $\sigma_1^T N \sigma_2 + F_1(\sigma_1) + F_2(\sigma_2)$, where N is a $\lceil \beta n \rceil \times \lceil \beta n \rceil$ matrix equal to twice the minor of M indexed by $A_1 \times A_2$ and each F_i is a polynomial function. Lemma 27 below, which is a slight generalization of results in [BRS93, Tha98], immediately implies the required bound $|\Gamma|$.

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

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 $(\sigma_1 - \sigma_1^*)^T N (\sigma_2 - \sigma_2^*) = F(\sigma_1, \sigma_2) + F(\sigma_1^*, \sigma_2^*) - F(\sigma_1, \sigma_2^*) - F(\sigma_1^*, \sigma_2) = 0$. 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 $v_1^T N v_2 = 0$ for all $v_1 \in V_1$ and $v_2 \in V_2$. This implies $\dim(V_1) + \dim(V_2) \leqslant 2t - \operatorname{rank}(N)$. The lemma now follows since $|\Sigma_i| = |\Sigma_i^*| \leqslant |D|^{\dim(V_i)}$. ∎

This completes the proof of Lemma 26. ∎

Next we want to apply Lemma 26 in the case that M is a GFT matrix, and to do this we want to lower bound $\Phi_M(\beta)$. Recall that a GFT matrix group G of order n over field $GF(q)$, with q and n relatively prime, is a square matrix N whose rows are indexed by elements of G and whose columns correspond to distinct characters of G over $GF(q)$. The two key properties of N are: (1) It has rank n, and (2) For elements $g_1, g_2 \in G$ and $j \in [n]$, $N_{g_1 g_2, j} = N_{g_1, j} N_{g_2, j}$.

LEMMA 28. *Let N be a GFT metrix corresponding to the group G of order n, over the field $GF(q)$ with q relatively prime to n. Any $u \times t$ minor of N has rank at least ut/n.*

This 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.)

Proof. For $H \subseteq G$ and any set C of columns, let $N_{H, C}$ denote the sub-matrix of N indexed by $H \times C$. Fix U of size u and J of size t, and consider the rank of $N_{U, J}$. Since N is nonsingular, $N_{G, J}$ has rank t implying that it has a $t \times t$ minor $N_{W, J}$, for some $W \subseteq G$, of full rank. Furthermore, for any $g \in G$, the matrix $N_{gW, J}$ is also a $t \times t$ matrix of full rank since, by property (2) above of GFT matrices, for $j \in J$ the jth column of $N_{gW, J}$ is the jth column of $N_{W, J}$ multiplied by the nonzero value $N_{g, j}$. Thus for any $g \in G$ the submatrix $N_{gW \cap U, J}$ of $N_{U, J}$ has rank $|gW \cap U|$, so it suffices to show that we can choose g so that $|gW \cap U| \geqslant ut/n$. Choose $g \in G$ at random. For each $g^* \in W$, $\mathbf{Prob}[gg^* \in U] = \mathbf{Prob}[g \in U(g^*)^{-1}] = u/n$. By linearity of expectation, the expected number of $g^* \in W$ for which $gg^* \in U$ is ut/n. Therefore, for some fixed g, $|gW \cap U| \geqslant ut/n$ and so $N_{gW \cap U, J}$ has rank at least ut/n. ▮

It follows from Lemma 28 that for a GFT matrix N, $\Phi_N(\beta) \geqslant \beta^2$. Combining this with Lemma 26, we have $\Psi_f(\beta) \geqslant \beta^2$ as required to prove the first part of Lemma 5.

We now turn to the second part. 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 $GF(q) - \{-b\}$, this becomes $a'(z_2 + c) + d$ for some $a' \neq 0$ which takes on all possible values in $GF(q)$ by varying z_2. Thus $QF_{M^{[0]}}\lceil_\rho$ is non-constant.

9. PROOF OF LEMMA 8

Let M be the $n \times n$ Sylvester matrix over $GF(3)$, and let $f = BQF_M$. As in the second half of Lemma 5 we want to show that to make f constant, we need to fix almost all of the variables. The argument used for the second half of Lemma 5 does not work here, because the values of z_1, z_2 are restricted to $\{0, 1\}$. 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 $I \subseteq [n]$ 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 $|J| \leqslant 3$ such that the sum of the entries in $M_{J,J}$ is a.*

Proof. For $a = 0$, set $J = \varnothing$. So assume $a \in \{1, -1\}$. Recall that the rows and columns of the Sylvester matrix are indexed by binary vectors of length k, which are identified naturally with subsets $[k]$ 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 fact, which we leave the reader to verify, 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 $[k]$ 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 an $|B_1 \cap B_2|$ 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 $\mathscr{F}$ be a family of sets. We say that $\mathscr{F}$ has property $P_{i,e}$ (respectively $P_{i,o}$) if the common intersection of every collection of i distinct sets from $\mathscr{F}$ is even (respectively odd). The following table gives an upper bound on the size of any family of subsets of $[k]$ 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 viewed as vector of length k over $GF(2)$. Let $|\mathscr{F}| = m$ and let S be the associated set of m characteristic vectors.

First, suppose $\mathscr{F}$ 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)th entry equals the parity of intersection of the ith and jth sets. By the property satisfied by $\mathscr{F}$, 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, $m = \operatorname{rank}(MM^T) \leqslant \operatorname{rank}(M) \leqslant k$.

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

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

Observe that $|\mathscr{G}| = |\mathscr{H}| = m+1$. Now let M and N be the $(m+1) \times (k+1)$ incidence matrices of $\mathscr{G}$ and $\mathscr{H}$ 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 $(m+1)$st column equal 0. Thus MN^T is an upper triangular matrix with all diagonal entries non-zero, implying that $m+1 = \text{rank}(MN^T) \leqslant \text{rank}(M) \leqslant k+1$, 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 $S' = \{v-s : v \in S\}$. ▮

Continuing the proof of Lemma 29, we now show that I contains a collection $J = \{A_0, B_1, B_2\}$ such that they all have even size or all have odd size, and exactly one pair of them has intersection of odd size. Then by the first part of Proposition 30, the sum of entries in $M_{J,J}$ is -1. A similar argument handles the other case.

Since $|I| \geqslant 4\sqrt{n}\log n$, 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. Define a graph on vertex set $\mathcal{F}$ where there is an edge between $A, B \in \mathcal{F}$ if $|A \cap B|$ is odd, and let α be the size of the largest independent set and ω be the size of the largest clique. If there is no triple A_0, B_1, B_2 as required to apply Proposition 30, then the graph is a union of disjoint complete graphs, which implies that $|\mathcal{F}| \leqslant \alpha\omega$. But Proposition 31 implies $\alpha\omega \leqslant k2^{k/2} = \sqrt{n}\log n < |\mathcal{F}|$, which is a contradiction proving that the required triple exists. ▮

We now have the tools to prove Lemma 8. 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 for an assignment σ to Z we have: Then $BQF_M\lceil_\rho(\sigma) = \sigma^T B\sigma + A\cdot\sigma + C$, where B denotes the sub-matrix of M corresponding to the rows and columns of Z, and the vector A and scalar C are determined by ρ and M. It suffices to show that $q(\sigma) = \sigma^T B\sigma + A\cdot\sigma$ takes on all possible values in $GF(3)$ for the various choices of 0–1 assignments σ 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 then 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, $q(\sigma)$ 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 (mod 3), and the lemma follows.

10. 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 u and w on the path that have the same variable label x, such that the arcs following them have different labels. If Z is a subset of variables and k is an integer, a path P is *read-k* on Z, if no variable of Z appears in P more than k times. We say that B is *syntactic (respectively, semantic) read-k on Z* if every path (respectively, every semantic path) in B is read-k on Z. (If Z contains all the variables, we omit the qualifying phrase "on Z" which conforms to the standard definition.)

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 functions is the construction of functions g_k on variable set $X \cup Y$, 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, we describe how the function f_k is computed from g_k.

Definition 32. Let g be a boolean function on variable set $X \cup Y$ and let $Y_1, Y_2, ..., Y_k$ be disjoint copies of Y. The kth extension of g on Y, denoted by $g^{(k)}$, is defined over variable set $X \cup Y_1 \cup \cdots \cup Y_k$. For $x \in \{0, 1\}^X$ and $y^1 \in \{0, 1\}^{Y_1}, ..., y^k \in \{0, 1\}^{Y_k}$, $g(x, y^1, ..., y^k) = 1$ if and only if (i) $g(x, y^1) = 1$ and (ii) $y^1 = \cdots = y^k$.

The relationship between computing g and its kth extension on Y is given by the following lemma:

Lemma 33. *Let g be a Boolean function on variable set $X \cup Y$, and let $g^{(k)}$ be its kth extension on Y.*

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

2. *If $g^{(k)}$ can be computed by a syntactic read-k branching program, then g 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 $g^{(k)}$, construct a branching program for g by replacing 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. On input $x, y^1, ..., y^k$, the first block of Q checks that $y^1 = \cdots = y^k$. 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 v accessing a variable v of Y, let i be the maximum number of occurrences of v along any path from the source to v and replace v with its corresponding variable in Y_i.

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 has the same graph structure as P and the nodes labeled with variables in X are identical in both. It follows that the syntactic or semantic properties with respect to X are identical. ∎

Suppose that for positive integer k, g is a function on $X \cup Y$ that satisfies: (i) it can be computed efficiently by a branching program that is syntactic read-k on Y and semantic read-once on X and (ii) it requires exponential size to be computed by any branching program that is syntactic read-k on X. Lemma 33 implies that its kth extension on Y, $g^{(k)}$ can be computed efficiently by a semantic read-twice branching program but requires an exponential size syntactic read-k branching program. Thus $g^{(k)}$ witnesses the desired separation. Our goal is to construct for each $k \geqslant 2$ a function g_k on $X \cup Y$ that satisfies (i) and (ii) above. We now do this.

Definition 34. • For positive integers n and k, we refer to the set $[n]^k$ as the *k-dimensional hypercube of side n.* A point v of the hypercube is denoted $(v_0, v_1, ..., v_{k-1})$, with coordinates indexed from 0. For d in $\{0, ..., k-1\}$ and $i \in [n]$, the ith *d-plane* is the set $P_i^d = \{v \in [n]^k : v_d = i\}$, i.e., P_i^d is the ith hyperplane perpendicular to the d axis.

• Without loss of generality, let $k+1 = 2^r$ for some integer r. Let X and Y be two sets of variables. The variables of X correspond to the points of $v \in [n]^{k+1}$ and take on values in $\{-1, 1\}$ which are also treated as elements of $GF(3)$ and the variables of Y correspond to ordered pairs (v, d) where $v \in [n]^{k+1}$ and $d \in \{0, ..., r-1\}$ and take on boolean values. Henceforth, x denotes an assignment to X and y denotes an assignment to Y. The assignment x can be viewed as an array x indexed by $[n]^{k+1}$ with entries in $\{-1, 1\}$. An assignment to Y can be viewed as an array y also indexed by $[n]^{k+1}$ whose entries are binary strings $y_v = y_v^{r-1}, ..., y_v^0$, each such string is interpreted as an integer in the range $\{0, ..., k\}$.

• For $d \in \{0, ..., k\}$ and $i \in [n]$, $\Pi_d^i(x)$ denotes the product of the entries of x that lie on the ith d plane and let $S_d(x) = \sum_{i=1}^n \Pi_d^i(x)$. Here all arithmetic is over $GF(3)$.

• For $d \in \{0, ..., k-1\}$, let $X^d = X^d(y)$ be the set of those $v \in [n]^k$ for which $y_v = d$. The sets $\{X^d : 0 \leqslant d \leqslant k\}$ partition X. The variables in X^d are said to be active for d.

• Given assignments x and y to X and Y, define x^d to be the assignment to X given by

$$x_v^d = \begin{cases} x_v & \text{if } \ v \in X^d \\ 1 & \text{otherwise.} \end{cases}$$

• Define the function H_d on $X \cup Y$ by $H_d(x, y) = S_d(x^d)$.

• Define the function g_k on $X \cup Y$

$$g_k(x,y) = \bigwedge_{d \in \{0,...,k\}} (H_d(x,y) \equiv 0 \pmod 3)$$

Note that for each d, $H_d(x, y)$ can be computed by a branching program of size $O(|X^d| + |Y|)$ that reads each variable of $Y \cup X^d$ exactly once and reads no other variables of X. Since the set X^d depends on Y, the branching program depends syntactically on all of the variables of X, but semantically only on X^d. Thus $g_k(x, y)$ can be computed by a branching program of size $O(|X|+(k+1)\,|Y|)$ that is syntactic read-$(k+1)$ on Y and semantic 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.*

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]. If $d \in \{0, ..., k\}$ and $J \subseteq [n]$, a *(d, J) transversal* is a pair A, B of disjoint subsets $A = \{a_j : j \in J\}$ and $B = \{b_j : j \in J\}$ of $[n]^{k+1}$ where a_j, b_j both belong to the jth d-plane. We call a function R on variables set X a *planar pseudo-rectangle of order m* if for some d and J of size m there is a (d, J)-transversal A, B such that R can be expressed as $R' \wedge R''$ where R' depends only on $X \setminus B$ and R'' depends only on $X \setminus A$.

PROPOSITION 37 ([BRS93, Tha98]). *Let f be a Boolean function on variable set $X = [n]^{k+1}$. Let $m = n/(6(k+1)\,2^{k+1})$ and $p = 36 \cdot k \cdot 2^k$. Suppose $|R^{-1}(1)| \leqslant t$ for any planar pseudo-rectangle R of order m that is portion of f. Then, any non-deterministic branching program for f that is syntactic read-k requires size $(|f^{-1}(1)|/t)^{1/(2kp)}$.*

The above proposition can be extended to handle branching programs that are syntactic read-k on a subset of the variables. Suppose f is a Boolean function on $X \cup Y$. We say that a function R on $X \cup Y$ is a *planar pseudo-rectangle of order m on X* if for some d and J of size m there is a (d, J)-transversal A, B such that $R = R' \wedge R''$, where R' depends on $Y \cup X \setminus B$ and R'' depends on $Y \cup X \setminus A$.

We have the following straightforward extension of Proposition 37:

PROPOSITION 38. *Let g be a Boolean function on variables set $X \cup Y$. Let $m = n/(6(k+1)\,2^{k+1})$ and $p = 36 \cdot k \cdot 2^k$. Suppose $|R^{-1}(1)| \leqslant t$ for any planar pseudo-rectangle R of order m on X that is a portion of g. Then, any non-deterministic branching program for g that is syntactic read-k on X requires size $(|f^{-1}(1)|/t)^{1/(2kp)}$.*

Proof (Sketch). Let P be a branching program computing g. The proof is essentially the same as 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. ▌

To apply Proposition 38 we first need to lower bound $|g_k^{-1}(1)|$.

Claim 39. $|g_k^{-1}(1)| \geqslant 2^{|X \cup Y| - 3(r+1)(k+1)}$

Proof. For each $d \in [0, k]$, let $I_d = \{v \in [n]^{k+1} : v_d \in \{2, 3, 4\}$ and $v_{d'} = 1$, for $d' \neq d\}$. Let X'_d (resp., Y'_d) be the set of variables of X (resp. Y) corresponding to the indices I_d. Let $X' = \bigcup_{d \in [0,k]} X'_d$ and $Y' = \bigcup_{d \in [0,k]} Y'_d$. 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 $|X' \cup Y'| = 3(r+1)(k+1)$, the claim follows.

Fix the variables in Y'_d 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 $X'_d = \{z_1, z_2, z_3\}$. Substituting the values according to $\rho, H_d(X, Y)$ can be written as $\sum_i u_i z_i + b$, for some constants $u_1, u_2, u_3 \in \{-1, 1\}$, and b. It is not too hard to show for any b, u_1, u_2, u_3 that there is a setting of the z_i's in $\{-1, 1\}$ so that $\sum_i u_i z_i + b = 0 \pmod 3$ Repeating this for all d's gives a satisfying assignment for $g_k(X, Y)$. ▮

Next we upper bound the size of a planar pseudo-rectangle of order m that is a portion of g_k.

CLAIM 40 *Let $R = R' \cup R''$ be a planar pseudo-rectangle of order m with associated (d, J)-transversal A, B. If R is a portion of g_k, then $|R^{-1}(1)| \leqslant 2^{|X \cup Y| - m/(4(k+1)^2)}$.*

Proof of Claim 40. By definition, there exists a d and an index set $J \subseteq [n]$ of size m such that $A = \{a_j : j \in J\}$ and $B = \{b_j : j \in J\}$ is a (d, J)-transversal. For an assignment π of Y, we say that the pair (a_j, b_j) with $j \in J$ is *good for y* if $a_j, b_j \in X^d(\pi)$. Let G_π be the set of $j \in J$ such that (a_j, b_j) is good for π. Let us write an assignment to $Y \cup X$ as (π, ρ, v) where π is the assignment to Y, ρ is the assignment to all of X except the positions indexed by G_π and v is the assignment to X for the indices G_π. We select a random assignment to $Y \cup X$ by first choosing π uniformly at random and then choosing ρ and v uniformly at random. We want to upper bound the probability that $(\pi, \rho, v) \in R^{-1}(1)$.

First consider fixed settings of π and ρ. Let Γ be the set of satisfying assignments of R consistent with the given fixed π and ρ. Because R is a portion of g_k, every assignment in Γ satisfies $H_d(x, y) \equiv 0 \pmod 3$. Having fixed π and ρ, the polynomial $H_d(X, Y)$ reduces to $(\sum_{j \in G_\pi} u_j a_j b_j) + c$, 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 that $|\Gamma_\rho| \leqslant 2^{|G_\pi|}$, while there are $2^{2|G_\pi|}$ possible assignments to v. From this we conclude that conditioned on fixed values of π and ρ the probability that $(\pi, \rho, v) \in R$ is at most $2^{-|G_\pi|}$.

For each fixed pair $j \in J$, the probability that $j \in G_\pi$ is $1/(k+1)^2$, so the expected size of G_π is $m/(k+1)^2$. Therefore, by Chernoff's bound, $\mathbf{Prob}[|G_\pi| \leqslant m/(3(k+1)^2)] \leqslant e^{-2m/(9(k+1)^2)}$.

So now we can then upper bound the probability that $(\pi, \rho, v) \in R^{-1}(1)$ by $\mathbf{Prob}[|G_\pi| \leqslant m/3(k+1)^2] + 2^{-m/3(k+1)^2} \leqslant e^{-2m/(9(k+1)^2)} + 2^{-m/3(k+1)^2)} \leqslant 2^{-m/(4(k+1)^2)}$. ▮

Using the claims, we apply Proposition 38 to g_k, with $t = 2^{|X \cup Y| - m/(4(k+1)^2)}$, to get a lower bound of $2^{(m/4(k+1)^2) - 3(r+1)(k+1))/(72k^2 2^k)}$ on the size of any branching program that is syntactic read-k on X, provided that $m = n/(6(k+1)\,2^{k+1})$. Taking n sufficiently large, this lower bound is bounded below by $2^{n/b(k)}$ for some function $b(k)$. This completes the proof of Theorem 35.

ACKNOWLEDGMENTS

We thank Amir Shpilka and Xiaodong Sun for pointing out some errors in an earlier version of this paper, and a referee for many helpful comments.

REFERENCES

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

[Abr91] K. R. Abrahamson, Time-space tradeoffs for algebraic problems on general sequential models, *J. Comput. System Sci.* **43** (1991), 269–289.

[Ajt98] M. Ajtai, Determinism versus non-determinism for linear time RAMs with memory restrictions, Technical Report TR98-077, Electronic Colloquium in Computation Complexity, Revision 1, 1998. [Available at `http://www.eccc.uni-trier.de/eccc/`.]

[Ajt99a] M. Ajtai, Determinism versus non-determinism for linear time RAMs with memory restrictions, *in* "Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing," pp. 632–641, 1999.

[Ajt99b] M. Ajtai, A non-linear time lower bound for boolean branching programs, *in* "Proceedings of the 40th Annual Symposium on Foundations of Computer Science," pp. 60–69, IEEE, New York, 1999.

[AM88] N. Alon and W. Maass, Meanders and their applications in lower bounds arguments, *J. Comput. System Sci.* **37** (1988), 118–129.

[BC82] A. Borodin and S. A. Cook, A time-space tradeoff for sorting on a general sequential model of computation, *SIAM J. Comput.* **11** (1982), 287–297.

[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 Trans. Comput. Aided Design Integrated Circuits Systems* **13** (1994), 401–424.

[Bea91] P. W. Beame, A general time-space tradeoff for finding unique elements, *SIAM J. Comput.* **20** (1991), 270–277.

[BF92] L. Babai and P. Frankl, Linear algebra methods in combinatorics with applications to geometry and computer science (Preliminary Version 2), University of Chicago, 1992.

[BFMadH$^+$87] A. Borodin, F. E. Fich, F. Meyer auf der Heide, E. Upfal, and A. Wigderson, A time-space tradeoff for element distinctness, *SIAM J. Comput.* **16** (1987), 97–99.

[BHST87] L. Babai, P. Hajnal, E. Szemerédi, and Turán, A lower bound of read-once-only branching programs, *J. Comput. System Sci.* **35** (1987), 153–162.

[BNS92] L. Babai, N. Nisan, and M. Szegedy, Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs, *J. Comput. System Sci.* **45** (1992), 204–232.

[BRS93] A. Borodin, A. A. Razborov, and R. Smolensky, On lower bounds for read-k times branching programs, *Comput. Complexity* **3** (1993), 1–18.

[Bry86] R. E. Bryant, Graph-based algorithms for boolean function manipulation, *IEEE Trans. Comput.* **35** (1986), 677–691.

[BSSV00] P. Beame, M. Saks, X. Sun, and E. Vee, Super-linear time-space tradeoff lower bounds for randomized computation, *in* "Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, November 2000," pp. 169–179, IEEE, New York.

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

[FvM00] L. Fortnow and D. van Melkebeek, Time-space tradeoffs for nondeterministic computation, *in* "Proceedings, Fifteenth Annual IEEE Conference on Computational Complexity," pp. 2–13, IEEE Computer Society Press, Los Alamitos, CA, 2000.

[G97] A. Gál, A simple function that requires exponential size read once branching programs, *Inform. Process. Lett.* **62** (1997), 13–16.

[HLP52] G. H. Hardy, J. E. Littlewood, and G. Polya, "Inequalities," Cambridge Univ. Press, Cambridge, UK, 1952.

[KW88] M. Karchmer and A. Wigderson, Monotone circuits for connectivity require super-logarithmic depth, *in* "Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, IL, May 1988," pp. 539–550.

[LV99] R. Lipton and A. Viglas, Time-space tradeoffs for sat, *in* "Proceedings of the 40th Annual Symposium on Foundations of Computer Science," pp. 459–463, IEEE, New York, 1999.

[MNT93] Y. Mansour, N. Nisan, and P. Tiwari, The computational complexity of universal hashing, *Theoret. Comput. Sci.* **107** (1993), 121–133.

[Pip79] N. J. Pippenger, On simultaneous resource bounds, *in* "20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1979," pp. 307–311, IEEE, New York.

[Raz91] A. A. Razborov, Lower bounds for deterministic and nondeterministic branching programs, *in* "Fundamentals of Computation Theory: 8th International Conference, FCT '91" (L. Budach, Ed.), Vol. 529, Lecture Notes in Computer Science, pp. 47–60, Springer-Verlag, Berlin, 1991.

[SS93] J. Simon and M. Szegedy, A new lower bound theorem for read only once branching programs and its applications, *in* "Advances in Computational Complexity" (J. Cai, Ed.), Vol. 13, DIMACS Series in Discrete Mathematics, pp. 183–193, Amer. Math. Soc., Providence, RI, 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," pp. 653–662, 1998.

[Weg86] I. Wegener, Time-Space trade-offs for branching programs, *J. Comput. System Sci.* **32** (1986) 91–96.

[Weg87] I. Wegener, "The complexity of Boolean Functions," 1st ed., Teubner, Stuttgart, 1987.

[Weg88] I. Wegener, On the complexity of branching programs and decision trees for clique functions, *J. Assoc. Comput. Math.* **35** (1988), 461–471.

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