Lower Bounds for Nondeterministic Semantic Read-Once Branching Programs

Stephen Cook* Jeff Edmonds † Venkatesh Medabalimi ‡ Toniann Pitassi §

August 10, 2015

Abstract

We prove exponential lower bounds on the size of semantic read-once 3-ary nondeterministic branching programs. Prior to our result the best that was known was for D-ary branching programs with $|D| \geq 2^{13}$.

1 Introduction

A major question in complexity theory is whether polynomial-time is the same as log-space or nondeterministic log-space. One approach to this problem is to study time/space tradeoffs for problems in P. For example, for natural problems in P, does the addition of a space restriction prevent a polynomial time solution? In the uniform setting, time-space tradeoffs for SAT were achieved in a series of papers [6, 13, 7]. The best current result shows that any algorithm for SAT running in space $n^{o(1)}$ requires time at least $\Omega(n^{\phi-\epsilon})$ where ϕ is the golden ratio $((\sqrt{5}-1)/2)$ and $\epsilon > 0$.

In the nonuniform setting, the standard model for studying time/space tradeoffs is the *branching program.* In this model, a program for computing a function $f(x_1, \ldots, x_n)$ (where the variables take values from a finite domain D) is represented by a directed acyclic graph with a unique source node called the start node. Each nonsink node is labelled by a variable and the edges out of a node correspond to the possible values of the variable. Each sink node is labelled by an output value. For Boolean functions, there is one sink node called the accept node, and all other sink nodes are rejecting nodes. Executing the program on an input corresponds to following a path from the start node, using the values of the input variables to determine which edges to follow. The output of the program is the value labeling the sink node reached. A D-ary branching program is deterministic if each non-sink node has exactly D edges, one for every value in D.

The length of a branching program is the number of edges in the longest path. It is clear that length of a branching program can be seen as a measure of computation time. The size of a branching program is the number of nodes in the program. For a boolean function f_n, let $BP(f_n)$ denote the minimal size of a branching program computing f_n. The space complexity $S(f_n)$ of non-uniform Turing machine computing f_n and $BF(f_n)$ are tightly related, $S(f_n) = O(\log(\max\{BP(f_n), n\}))$

*University of Toronto, email: sacook@cs.toronto.edu
†York University, email: jeff@cs.yorku.ca
‡University of Toronto, email: venkatm@cs.toronto.edu
§University of Toronto, email: toni@cs.toronto.edu

and $BP(f_n) = 2^{O(\max\{S(f_n), \log n\})}$ [5, 15]. This motivates the study of branching program size lower bounds. In particular, size lower bounds on length restricted branching programs translate to time/space tradeoffs.

The state of the art time/space tradeoffs for branching programs were proven in the remarkable papers by Ajtai [1] and Beame-et-al [3]. In the first paper, Ajtai exhibited a polynomial-time computable Boolean function such that any subexponential size deterministic branching program requires superlinear length. This result was significantly improved and extended by Beame-et-al who showed that any subexponential size randomized branching program requires length $\Omega(n \frac{\log n}{\log \log n})$.

Lower bounds for nondeterministic branching programs have been more difficult to obtain. In this model, there can be several arcs (or no arcs) out of a node with the same value for the variable associated with the node. An input is accepted if there exists at least one path consistent with the input from the source to a 1-node. A nondeterministic branching program computes a function f if its accepted inputs are exactly equal to $f^{-1}(1)$. From here on, we shall restrict our attention to non-deterministic branching programs.

Length-restricted nondeterministic branching programs come in two flavors: *syntactic* and *semantic*. A length l syntactic model requires that every path in the branching program has length l, and similarly a read-k syntactic model requires that every path in the branching program reads every variable at most k times. In the less restricted semantic model, the requirement is only for *consistent* source to sink paths; that is, paths along which no two tests $x_i = d_1$ and $x_i = d_2$, $d_1 \neq d_2$ are made. This is equivalent to requiring that for every accepting path, each variable is read at most k times. Thus for a nondeterminsitic read-k semantic branching program, the overall length of the program can be unbounded.

Note that any syntactic read-once branching program is also a semantic read-once branching program, but the the opposite direction does not hold. In fact, Jukna [8] proved that semantic read-once branching programs are exponentially more powerful than semantic read-once branching programs, via the "Exact Perfect Matching"(EPM) problem. The input is a (Boolean) matrix A, and A is accepted if and only if every row and column of A has exactly one 1 and rest of the entries are 0's i.e if its a permutation matrix. Jukna gave a polynomial-size semantic read-once branching program for EPM, while it was known that syntactic read-once branching programs require exponential size [12, 11].

Lower bounds for syntactic read-k (nondeterministic) branching programs have been known for some time [14, 4]. However, for *semantic* nondeterministic branching programs, even for read-once, no lower bounds are known for polynomial time computable functions for the $|D| = 2$ case. The best lower bound known prior to our work is an exponential lower bound for semantic read-once (nondeterministic) $|D|$-way branching programs, where $|D| = 2^{13}$ [9]. In fact this lower bound actually holds more generally for semantic read-k but where $|D| = 2^{3k+10}$.

Jukna obtains his result by showing that any time restricted semantic branching program of small size has a large rectangle in $f^{-1}(1)$. He uses the explicit function of computing the characteristic function of a linear code having minimum distance $m + 1$ defined over $GF(q)$. Given a parity matrix Y,the function $g(Y, x) = 1$ iff x is a codeword. Since codewords in a linear code of minimum distance $m + 1$ can only have an m-rectangle of size 1 he argues that a time restricted branching program of length kn computing g requires a size of $2^{\Omega(n/k^2 4^k)}$. This exponential lower bound can be obtained whenever D is sufficiently large in comparison to k, $|D| = q \geq 2^{3k+10}$.

Jukna's result is an improvement from exponential lower bounds with a domain requirement of $2^{2^{ck}}$ obtained in [2]. Beame et.al [2] obtain their result by characterizing the function computed by a time restricted branching programs of small size as a union of shallow decision forests where the size of the union depends on the size of the branching program. Each shallow forest is then shown to be representable by a collection of *small* number of βn-pseudo-rectangles in $f^{-1}(1)$. (Pseudo-rectangles are a generalization of what we call embedded rectangles later). This gives a representation of the branching program as a union of small(in the size 's') number of βn-pseudo-rectangles. Now, if for some function f the maximum size of a βn-pseudo-rectangle is $|D|^{(1-\psi_f(\beta))n}$ and the number of *yes*-instances $|f^{-1}(1)| \geq |D|^{(1-\eta(f))n}$ then the number of βn-pseudo-rectangles will be at least $|D|^{(\psi_f(\beta)-\eta(f))n}$. This yields an exponential lower bound on s for sufficiently large $|D|$ whenever $(\psi_f(\beta) - \eta(f))$ is bounded away from 0 by some $\epsilon > 0$. They then exhibit an explicit function with this property. Their function $QF_M : GF(q^n) \rightarrow \{0, 1\}$ is based on quadratic forms using a modified Generalized Fourier Transform matrix. They show that there exists a constant $c > 0$ such that for all k and $\epsilon \in (0, 1)$, if $D \geq 2^{2^{\frac{c}{\epsilon}k}}$ then a non-deterministic BP of length kn computing QF_M needs size at least $S = 2^{n \log^{1-\epsilon}|D|}$. For the specific case of $k = 1$, it can be shown that if their analysis of maximum size of βn-pseudo-rectangles in QF_M is tight, a domain size of at least $|D| \geq 2^{64}$ is needed.

Our main result is an exponential lower bound on the size of semantic read-once nondeterministic branching programs for a polynomial time decision problem f for 3-ary inputs. Similar in spirit to these previous results [9, 2] we show that a small sized semantic read once branching program is bound to have a large rectangle in $f^{-1}(1)$. However in addition, we show that one can always find a balanced rectangle in $f^{-1}(1)$ of size r^2 where r is some large constant $(\star)$. A balanced rectangle is one which is reasonably close to being a square.

The particular polynomial time decision problem we use to prove the lower bound is: to decide if a polynomial over a finite field K evaluates to a value less than a certain threshold at a given input. The input is a pair (u, x) where u is the description of a degree $d-1$ polynomial over $[K]$ and $x \in [K]$ and we want to accept if and only if $u(x) < K^{1-\delta}$. We actually prove a stronger theorem: with high probability over all polynomials u, any nondeterministic semantic read-once branching program for what we shall call $Poly_u$ requires exponential size. That is, even if the branching program knows the polynomial u, for a typical u it cannot efficiently do polynomial evaluation. The main properties of polynomials over finite fields we are using are polynomial interpolation, and lemma 7, which might be interpreted to mean something like: the spread of values of a typical random polynomial of degree d over a field K is roughly close to being uniform over K, provided K is sufficiently large.

Continuing with the above observation $(\star)$, since the number of balanced rectangles of a certain size $d = r^2$ is *small* and since each one of them can be a rectangle in $f^{-1}(1)$ for a relatively *small* number of degree d polynomials over K as a consequence of polynomial interpolation, we argue that there must be a polynomial with no balanced rectangle in $f^{-1}(1)$ and hence the branching program computing it should be *large*. A key idea of this argument is that for a balanced rectangle the sum of the lengths of the rectangle can be at most a *small* fraction of its area.

By a simple padding argument, we can modify our problem $Poly_u$ and actually achieve the lower bound for domain size $2 + \epsilon$ for arbitrarily small $\epsilon > 0$. In this model, we can define the problem to have $N = n + M$ variables, $M = \Theta(N)$ of them with domain size 3 and the rest, with domain size 2, do not affect the output.

2 Definitions

Throughout this article, D denotes a finite set. For finite set N, D^N is the set of maps from N to D. An element of N is called a *variable index* or simply an *index*. We normally take N to be $[n]$ for some integer n, and write D^N for $D^{[n]}$. If $A \subseteq N$, a point $\sigma \in D^A$ is a *partial input* on A. For a partial input σ, $fixed(\sigma)$ denotes the index set A on which it is defined and $unfixed(\sigma)$ denote the set $N - A$. If σ and π are partial inputs with $fixed(\sigma) \cap fixed(\pi) = \emptyset$, then $\sigma\pi$ denote the partial input on $fixed(\sigma) \cup fixed(\pi)$ that agrees with σ on $fixed(\sigma)$ and with π on $fixed(\pi)$.

For $x \in D^N$ and $A \subseteq N$, the *projection* x_A of x onto A is the partial input on A that agrees with x. For $S \subseteq D^N$, $S_A = \{x_A \mid x \in S\}$.

2.1 Nondeterministic Read-Once Semantic Branching Programs

Let $f : D^N \to \{0, 1\}$ be a boolean function whose input is given in $|D|$-ary. Let the input variables be $x_1, \ldots, x_n$ where $x_i \in D$ for all $i \leq n$. A $|D|$-way nondeterministic branching program (for f) is an acyclic directed graph G with a distinguished source node q_{start} and a distinguished sink node (the accept node) q_{accept}. We refer to nodes as *states*. Each non-sink state is labelled with some input variable x_i, and each edge directed out of a state is labelled with some value $b \in D$ for x_i. For each $Z \in D^N$, the branching program accepts Z if and only if there exists at least one (directed) path starting at the q_{start} and leading to the accepting state q_{accept}, and such that all labels along this path are consistent with Z. The size of a branching program is the number states (i.e. nodes) in the graph.

A branching program is *semantic read-k* if for every path from q_{start} to q_{accept} that is consistent with some input, each variable occurs at most k times along the path. For the read-once case, a semantic branching program allows variables to be read more than once, but each accepting path may only query each variable once.

2.2 Polynomial Evaluation Problem

Our hard computational problem is the polynomial evaluation problem, *Poly*, with parameters K, d, δ, where $0 < \delta < 1$. The input is a pair (u, Z) where $u \in [K]^d$ specifies a degree $d - 1$ polynomial over the field $[K]$ (K a prime power), and $Z \in [K]$ specifies a field value. $Poly(u, Z) = 1$ if and only if the polynomial specified by u on input Z evaluates to a number less than $K^{1-\delta}$. (We compare two field elements by comparing them using the natural ordering on ternary strings.)

We will work with $|D|$-ary branching programs (with $|D|$ prime), and let $K = |D|^N$. The input will be given as a vector in $D^{(d+1)N}$. The first dN coordinates specifies u and the last N coordinates specifies Z. Thus the total input length is $(d + 1)N$. In the remainder of the paper, $|D| = 3$, and thus the parameters of *Poly* are d, δ, N. Both d and δ will be fixed constants. Let $Poly_u$ denote the polynomial evaluation problem with parameters d, δ, n where the polynomial u is fixed.

2.3 Rectangles and Embedded Rectangles

We use the same definitions and conventions as in [3].

A product $U \times V$ is called a (combinatorial) rectangle. If $A \subseteq N$ is an index subset and $Y \subseteq D^A$ and $Z \subseteq D^{N-A}$, then the product set $Y \times Z$ is naturally identified with the subset $R = \{\sigma\rho \mid \sigma \in Y,\ \rho \in Z\}$ of D^N, and a set of this form is called a *rectangle* in D^N.

An *embedded rectangle* R in D^N is a triple $(\pi_{red}, \pi_{white}, C)$ where π_{red}, π_{white} are disjoint subsets of N, and $C \subseteq D^N$ satisfies: (i) The projection $C_{N-\pi_{red}-\pi_{white}}$ consists of a single partial input w, (ii) if $\tau_1 \in C_{\pi_{red}}$ and $\tau_2 \in C_{\pi_{white}}$, then the point $\tau_1\tau_2 w \in C$. C is called the *body* of R. The sets π_{red}, π_{white} are called the *feet* of the rectangle; the sets $C_{\pi_{red}}$ and $C_{\pi_{white}}$ are the *legs*, and w is the *spine*. We can also specify an embedded rectangle by its feet, legs and spine: $(\pi_{red}, \pi_{white}, A, B, w)$ where π_{red}, π_{white} are the feet, $A = C_{\pi_{red}}, B = C_{\pi_{white}}$ are the legs, and w is the spine.

We will sometimes refer to A as the *red* side of the rectangle and to B as the *white* side of the rectangle. The *size* of the rectangle is $|A| \cdot |B|$, and the dimension of the rectangle is m_r-by-m_w where $m_r = |\pi_{red}|$ and $m_w = |\pi_{white}|$.

3 Lower Bound for $|D| = 3$

Theorem 1 *There exists constants d, δ such that for sufficiently large n, for a random u, with probability greater than 1/4, any 3-ary nondeterministic semantic read-once branching program for $Poly_u$ requires size at least $2^{\Omega(n)}$.*

Corollary 2 *There exists constants d, δ such that for sufficiently large n, any 3-ary nondeterministic semantic read-once branching program for Poly with parameters d, δ, n requires size at least $2^{\Omega(n)}$.*

Overview of Proof Call a degree $d-1$ polynomial "good" if the fraction of accepting instances is roughly what you would expect from a random function; that is, if the fraction of yes instances is at least $\frac{1}{2}K^{-\delta}$. Lemma 7 shows that at least half of all degree $d-1$ polynomials are good.

The main lemma (Lemma 3) shows that for all good polynomials, we can associate with every size $s = 2^{o(n)}$ branching program, $\mathcal{P}$, an m_r-by-m_w embedded rectangle $R_{\mathcal{P}}$ of size r^2, where r will be a large constant, and m_r and m_w will be roughly equal, and will each be a constant fraction of n. For simplicity of calculations for now, assume that $m_r = m_w = m$. The rectangle will have the property that $\mathcal{P}$ accepts every input in $R_{\mathcal{P}}$; in other words, $R_{\mathcal{P}}$ is a 1-rectangle of $\mathcal{P}$. Choosing $d = r^2$, each rectangle of size r^2 can be a 1-rectangle for very few degree $d-1$ polynomials – at most a $|D|^{-n\delta r^2}$ fraction of all degree $d-1$ polynomials. (This is Lemma 6.) Secondly, the total number of such rectangles is fairly small – of size roughly $|D|^{O(rm)}$ (Lemma 5). The key point is that the *number* of rectangles is roughly $|D|^{2rm}$ – the exponent grows linearly in r. (More precisely it grows linearly in the sum of the lengths of the sides of the rectangle). But on the other hand, the probability that a degree $d = r^2$ polynomial takes on values less than $K^{1-\delta}$ within the rectangle is roughly $|D|^{-mr^2}$ – that is, the exponent grows *quadratically* with r. Because $|D|^{-n\delta r^2}|D|^{O(rn)}$ is less than 1/4, this implies that many good degree $d-1$ polynomials have no size r^2 1-rectangle, thus proving the theorem.

Note that we set our parameters so that the area of the rectangle $R_{\mathcal{P}}$ is at least the degree d of the polynomial u. (Thus $r^2 \geq d$.) A crucial point in the above argument is that the sum of the lengths of the sides of $R_{\mathcal{P}}$ must be at most a fraction of its area. This requires that the rectangle is reasonably close to being square. We put extra effort into making sure that the rectangle is square (without compromising too much of its size in order to make it square). This enables us to achieve domain size 3; a somewhat simpler argument achieves domain size 5.

Lemma 3 *(Main Lemma) Let u be a degree $d-1$ polynomial over $[K]$ such that the fraction of inputs that map to $[K^{1-\delta}]$ is at least $\frac{1}{2}K^{-\delta}$. Suppose that the following inequalities are satisfied* *for our parameters: (1) $m_w = 2m_r = \gamma n$; (2) $|D|^{m_r} \geq |D|^{m_w}(1/2 - 2\gamma)^{m_w}$; (3) $r \leq 1/4(1/2 - \gamma)^{m_r}|D|^{m_r - \delta N}/s$. Then if $\mathcal{P}$ is a $|D|$-way nondeterministic semantic read-once branching program of size s for $Poly_u$ with parameters $d = r^2, \delta, n$ then there is an m_r-by-m_w embedded rectangle $R = (\pi_{red}, \pi_{white}, A, B, w)$ such that every input in R is accepted by $\mathcal{P}$, and where $|A|, |B| = r$*

Proof: Let u be a degree $d-1$ polynomial such that the density of 1's is at least $\frac{1}{2}K^{-\delta}$, and consider a size s nondeterministic semantic read-once branching program, $\mathcal{P}$ for $Poly_u$. Let S_0 be the set of inputs that are accepted by $\mathcal{P}$; since $\mathcal{P}$ is assumed to be correct for all inputs for u, $|S_0| \geq \frac{1}{2}K^{-\delta}|D|^n$. For each accepting instance $I \in S_0$, fix one accepting path, p_I, in the branching program. Each of the n variables must be read along this path at most once and thus each accepting instance I has an associated permutation π_I of the n variables. (It is possible that some input variables are not read in p_I. We place all such variables at the beginning of π_I, as though though they were read first.) Designate state q_I as the state in p_I which occurs just after the first half of the variables in π_I. Now define q to be the most common designated state (over all accepting inputs $I \in S_0$), and let $S_1 \subseteq S_0$ denote the corresponding set of inputs whose designated state is q. Thus for each input I in S_1, there is an accepting path p_I that passes through state q. Because $\mathcal{P}$ has size s, it follows that

$$|S_1| \geq |S_0|/s \geq \frac{1}{2}K^{-\delta}|D|^n/s = \frac{1}{2}|D|^{-\delta n}|D|^n/s \tag{1}$$

We now want to pick two subsets of coordinates $\pi_{red} \subseteq N$ and $\pi_{white} \subseteq N$, of size m_r and m_w respectively, and a set $S^* \subseteq S_1$ of inputs with the property that for every input $I \in S^*$, and associated accepting path p_I, not only does it pass through state q, but every coordinate in π_{red} is read before state q (or not read at all), and every coordinate in π_{white} read at or after state q (or not read at all). We will first pick π_{red} greedily. For each $I \in S_1$, at most $n/2$ of the n coordinates in p_I occur in π_I before reaching state q, and thus there is some coordinate i such that for at least half of the inputs $I \in S_1$, i occurs in π_I before reaching state q. After choosing the first coordinate, there are at least $|S_1|/2$ inputs remaining. Continue greedily until we pick m_r coordinates, π_{red}, always choosing the most popular coordinate that occurs in π_I before reaching state q. By averaging, when the i^{th} coordinate, $i \leq m_r < \gamma n$ is chosen, the fraction of inputs that remain is at least $\frac{(n/2-i)}{(n-i)} \geq \frac{(n/2-\gamma n)}{(n-\gamma n)} \geq \frac{(n/2-\gamma n)}{n} = (1/2 - \gamma)$. Let $S_2 \subseteq S_1$ denote the set of inputs such that all coordinates in π_{red} are read before reaching q (or not read at all). It follows that

$$|S_2| \geq (1/2 - \gamma)^{m_r}|S_1| \tag{2}$$

By assumption (3) in the statement of the Lemma, we have

$$r \leq 1/4(1/2 - \gamma)^{m_r}|D|^{m_r - \delta n}/s \tag{3}$$

Then from (1), (2), and (3) we have

$$|S_2| \geq 2r|D|^{n - m_r} \tag{4}$$

For each $w \in D^{N - \pi_{red}}$, the average number of extensions of w in S_2 is $2r$. We want to prune S_2 such that every $w \in D^{N - \pi_{red}}$ has at least r extensions. To do this, define $S_3 \subseteq S_2$, where we remove all inputs $(w, *)$ from S_2 such that w has less than r extensions in S_2. Since we delete at most $r|D|^{n-m_r}$ elements from S_2, and $|S_2| \geq 2r|D|^{n-m_r}$, it follows that

$$|S_3| \geq r|D|^{n-m_r} \tag{5}$$

Next we will choose m_w coordinates, π_{white} in the same greedy fashion, and let S_4 denote the set of all inputs in S_3 such that all coordinates in π_{white} are read after reaching q. Again by averaging,

$$|S_4| \geq (1/2 - 2\gamma)^{m_w}|S_3| \tag{6}$$

We will express S_4 as the disjoint union of sets R_w: choose a value w for the coordinates outside of $\pi_{red} \cup \pi_{white}$. The corresponding set $R_w \subseteq S_4$ consists of all inputs (α, w, β) such that α is an assignment to the variables in π_{red}, β is an assignment to the variables in π_{white}, and $(\alpha, w, \beta) \in S_4$.

Claim 4 *For each w: (i) R_w is an embedded rectangle and (ii) as long as R_w is not empty, the size of the red side is at least r.*

Proof: We will first show that R_w is an embedded rectangle. Let $S_{red} \subseteq D^{\pi_{red}}$ be the projection of R_w onto the coordinates of π_{red} and let $S_{white} \in D^{\pi_{white}}$ be the projection of R_w onto the coordinates of π_{white}. Setting $A = S_{red}$, $B = S_{white}$ and $w = w$, we claim that R_w is equal to the embedded rectangle defined by $(\pi_{red}, \pi_{white}, A, B, w)$. To see this, consider $x, x' \in A$ and $y, y' \in B$ such that $xyw \in R_w$, and $x'y'w \in R_w$. Let I be the input corresponding to xyw and let p_I be the corresponding path going thru state q. Note that in p_I the x-variables are all read prior to reaching q, and the y-variables are read after reaching q, and there is some split of the w variables into w_1, w_2 where the w_1 variables are read prior to q and the w_2 variables are read after q. Similarly, let I' be the input corresponding to $x'y'w$ and let $p_{I'}$ be the corresponding path. There is now a possibly different split of w into w'_1, w'_2, so x', w'_1 are read before q and y', w'_2 are read after q. We claim that $xy'w \in R_w$: consider the path (x, w_1) (the first half of p_I) and (y', w'_2) (the second half of $p_{I'}$). This path must be consistent since w_1 and w'_2 are consistent and x, y' are on disjoint variables. Thus there is an input consistent with this path; it is an accepting path going through q and consistent with w; the variables in π_{red} are all read before q, and the variables in π_{white} are all read after q. Thus it is in R_w. An analogous argument shows that $x'yw \in R_w$. Thus R_w is an embedded rectangle.

Secondly we will show (ii) for each $R_w \subseteq S_4$, the size of the red side is at least r. (That is, $|A| \geq r$.) Consider a nonempty rectangle R_w with red side A, white side B and stem w. Recall that the inputs in S_3 consist of a partial input $w+ \in D^{N-\pi_{red}}$ together with a set $A \subseteq D^{\pi_{red}}$ such that $|A| \geq r$. We obtain S_4 from S_3 by selecting m_w coordinates from $N - \pi_{red}$, one at a time, choosing each coordinate greedily, where a coordinate is chosen if it is read *after* state q in the most inputs. Consider a block of inputs $(A, w+) \in S_3$. If some input $(\alpha, w+) \in (A, w+)$ survives, then all coordinates in π_{white} that were chosen must all be read after state q on input $(\alpha, w+)$. But this means that for every input $(\alpha', w+) \in (A, w+)$, all coordinates in π_{white} are also read after q. (Otherwise, some coordinate would be read twice along this accepting input, violating the read-once condition.) Thus, either the entire block $(A, w+)$ is in S_4, or the entire block is removed from S_4.

Now let $R_w = (\pi_{red}, \pi_{white}, A, B, w) \subseteq S_4$ be a nonempty rectangle, $w \in D^{N-\pi_{red}-\pi_{white}}$. R_w is obtained by taking the union of (nonempty) blocks $(A', w+) \in S_4$, $w+ \in D^{N-\pi_{white}}$. Since as we argued above, for each such block, $|A'| \geq r$, it follows that $|A| \geq r$ as well.

■

Let r_{avg} denote the average size of the white side of the rectangle over all rectangles $R_w \subseteq S_4$ It is easy to see that $|D|^{n-m_w} r_{avg} \geq r|D|^{n-m_r}(1/2-2\gamma)^{m_w}$. Setting $r_{avg} = r$, we get $|D|^{m_r} \geq |D|^{m_w}(1/2-2\gamma)^{m_w}$. Thus we can pick some setting w^* to the remaining $n - m_r - m_w$ uncolored coordinates (the coordinates that are not in π_{red} or π_{white}) such that the white side of the rectangle R_{w^*} has size at least $r_{avg} = r$. Let S^* equal R_{w^*}. By construction, both the red side of $S^* = R_{w^*}$ and the white side of R_{w^*} have size at least r. Prune S^* so that each has size exactly r, thus completing the proof of the lemma. ■

Lemma 5 *Let $\mathcal{R}$ be the set of all m_r-by-m_w embedded rectangles over D^N such that $|A| = |B| = r$, where $m_w = \gamma n$ and $m_r = m_w/2$. Then $|\mathcal{R}| \leq (e/\gamma)^{\frac{3}{2}m_w}|D|^{\frac{3}{2}rm_w + m_w/\gamma}$*

Proof: The number of choices for π_{red}, the coordinates of A, is $\binom{n}{m_r}$. Given π_{red}, we choose r vectors from the $|D|^{m_r}$ possible values for the elements of A. Thus the total number of possible sets A is at most $\binom{n}{m_r}|D|^{rm_r}$. Similarly the number of choices for the set B is at most $\binom{n}{m_w}|D|^{rm_w}$. The number of choices for $w \in D^{N-\pi_{red}-\pi_{white}}$ is $|D|^{n-m_r-m_w}$. Thus $|\mathcal{R}|$ is at most $\binom{n}{m_r}\binom{n}{m_w}|D|^{rm_r}|D|^{rm_w}|D|^{n-\frac{3}{2}m_w}$. Using the inequality $\binom{n}{k} \leq (\frac{en}{k})^k$ we conclude the number of choices for $|\mathcal{R}|$ is at most $(en/m_w)^{m_w}(2en/m_w)^{\frac{1}{2}m_w}|D|^{n-\frac{3}{2}m_w}|D|^{\frac{3}{2}rm_w} \leq (e/\gamma)^{\frac{3}{2}m_w}|D|^{m_w/\gamma+\frac{3}{2}rm_w}$ ■

Lemma 6 *Define the predicate Good(R, u) to be true if for every input Z in the rectangle R, the polynomial u on input Z is less than $K^{1-\delta}$ (i.e. $Poly_u(Z)$ is true). Then for all embedded rectangles R of size d, $Pr_u[Good(R, u)] \leq p$ where $p = |D|^{-\delta nd}$.*

Proof: Assume $Good(R, u)$. Then the branching program accepts all instances in R. Suppose that $|R| = d$ and let $B' \in [K^{1-\delta}]^d$ specify a vector of d accepting values. Let $Good_{B'}(R, u)$ to be the event that for all $Z \in R$, $F_u(Z) = B'(Z)$. Then $Pr_u[Good(R, u)] = K^{(1-\delta)d} \cdot Pr_u[Good_{B'}(R, u)]$.

To bound $Pr_u[Good_{B'}(R, u)]$, suppose that it is true that $\forall Z \in R$, $F_u(Z) = \sum_{i \leq d} u_i Z^i = B'(Z)$. Note that this fixes the output of the degree $d-1$ polynomial for d values of Z. By interpolation, this uniquely determines the polynomial, u'. Thus, $Pr_u[Good_{B'}(R, u)] = Pr_u[u = u'] = K^{-d} = |D|^{-nd}$. Overall, $Pr_u[Good(R, u)] \leq K^{(1-\delta)d}|D|^{-nd} = |D|^{n(1-\delta)d}|D|^{-nd} = |D|^{-\delta nd}$. This completes the proof of Lemma 6.

■

Lemma 7 *For any given $\tau \in (0, 1)$, for fixed parameters $d \geq 2, \delta$ and for sufficiently large K, with probability over u greater than $1/2$, $Poly_u(\delta)$ accepts at least a fraction $\tau K^{-\delta}$ of all the inputs.*

Proof: Consider the Boolean random variable a_x which takes the value $poly_{u,\delta}(x)$ over randomly sampled $u \in [K]^{d+1}$. Let $N = \sum_{x \in K} a_x$. At a given u, $N(u)$ denotes the number of elements in the domain $[K]$ that get mapped to a value less than $K^{1-\delta}$. One can show that the random variables a_x are $d-$wise independent. For our lemma it suffices to show that a_x are pairwise independent. Let $Polyval_u(x_1) = k_1$ denote the event that the degree d polynomial obtained from a uniformly randomly sampled coefficient vector $u \in [K]^{d+1}$ evaluates to k_1 at $x_1 \in K$. For every $x_1, k_1 \in [K]$, for any arbitrary choice of all coefficients but the constant term in u there is exactly one value of the constant term such that $Polyval_u(x_1) = k_1$, and thus $Pr_u(Polyval_u(x_1) = k_1) = \frac{1}{K}$. Now consider the joint event $Q = \{Polyval_u(x_1) = k_1 \wedge Polyval_u(x_2) = k_2\}$, where $x_1 \neq x_2$. Make an arbitrary choice of the first $d-1$ coordinates in u leaving u_1, the coefficient of the linear term and u_0 the constant term. We get a system of two equations in two variables corresponding to the event Q.

$$c_{x_1} + u_1 x_1 + u_0 = k_1$$
$$c_{x_2} + u_1 x_2 + u_0 = k_2$$

where c_{x_1}, c_{x_2} are the respective constants from partial evaluation. Since $x_1 \neq x_2$ there is a unique solution for the pair (u_1, u_2) and hence a unique completion of u. So the probability of Q,

$$Pr(\{Polyval_u(x_1) = k_1 \wedge Polyval_u(x_2) = k_2\}) = \frac{1}{K^2}$$

This proves the pairwise independence of the events $Polyval_u(x_1) = k_1$ and $Polyval_u(x_2) = k_2$ when $x_1 \neq x_2$. The pairwise independence of a_{x_1} and a_{x_2} follows from this since $Pr_u(a_{x_1} = 1) = Pr[\vee_{k \in [K^{1-\delta}]} Polyval_u(x_1) = k]$.

Now by linearity of expectation,

$$E_u[N] \quad = \textstyle\sum_{x \in K} E_u[a_x] = K \cdot Pr_u(Polyval_u(x) < K^{1-\delta}) \quad = K \cdot K^{-\delta} = K^{1-\delta}$$

We shall show that for most u (at least half of them), $N(u)$ is close to $K^{-\delta}$. The variance σ^2 can be computed as follows.

$$\begin{aligned}
\mathbb{E}_u[(N - \mathbb{E}_u[N])^2] = \mathbb{E}_u[N^2] - (\mathbb{E}_u[N])^2 &= \mathbb{E}_u[\sum_x a_x^2 + \sum_{x \neq y} a_x a_y] - K^{2(1-\delta)} \\
&= \mathbb{E}_u[\sum_x a_x + \sum_{x \neq y} a_x a_y] - K^{2(1-\delta)} \\
&= K^{1-\delta} + \sum_{x \neq y} \mathbb{E}_u[a_x]\mathbb{E}_u[a_y] - K^{2(1-\delta)} \\
&\qquad \text{(by pairwise independence)} \\
&= K^{1-\delta} + K(K-1)K^{-2\delta} - K^{2(1-\delta)} \\
&= K^{1-\delta} + K^{2-2\delta} - K^{1-2\delta} - K^{2-2\delta} \\
&= K^{1-\delta} - K^{1-2\delta}
\end{aligned}$$

By Chebycheff's inequality, $\forall \eta > 0$ we have $Pr_u(|N - \mathbb{E}_u(N)| \geq \eta\sigma) < \frac{1}{\eta^2} \implies$

$$Pr_u(|N - K^{1-\delta}| \geq \eta(K^{1-\delta} - K^{1-2\delta})^{\frac{1}{2}}) < \frac{1}{\eta^2}$$
$$\implies Pr_u(|N - K^{1-\delta}| \geq \eta K^{\frac{(1-\delta)}{2}}) < \frac{1}{\eta^2}$$
$$Pr_u\{K^{1-\delta} - \eta K^{\frac{(1-\delta)}{2}} \leq N \leq K^{1-\delta} + \eta K^{\frac{(1-\delta)}{2}}\} \geq 1 - \frac{1}{\eta^2}$$

For any given $\tau \in (0,1), \eta > 0$ for a sufficiently large K we have that $K^{1-\delta} - \eta K^{\frac{1-\delta}{2}} > \tau K^{1-\delta}$. So given any $\tau \in (0,1)$ for $\eta = \sqrt{2}$ for a sufficiently large K we have $Pr_u(N \geq \tau K^{1-\delta}) \geq \frac{1}{2}$. This completes the proof of the lemma. ■

We are now ready to complete the proof of the theorem. Call a polynomial u “good” if $Poly_u$ accepts at least a $\frac{1}{2}K^{-\delta}$ fraction of all inputs. By Lemma 7 with $\tau = \frac{1}{2}$, we know that at least half of all u's are good. For each good u, Lemma 3 tells us that any small branching program for $Poly_u$ implies that there exists an m-rectangle of size r^2 that is accepted (assuming that conditions (1), (2), and (3) are satisfied).

On the other hand, Lemmas 5 and 6 together tell us that at most a $p|\mathcal{R}|$ fraction of degree $d-1$ polynomials u have such monochromatic m-rectangles of size r^2. Suppose we can choose a setting of the parameters so that $p|\mathcal{R}| < 1/4$. Then this implies that at least 1/4 of all good polynomials cannot have small branching programs, and thus the theorem is proven.

It is left to show that we can set the parameters such that $p|\mathcal{R}| < 1/4$, and properties (1), (2), and (2) of Lemma 3 are satisfied. We will set the parameters as follows: $|D| = 3$, $m_w = 2m_r = \gamma n$, $\gamma = .01$, $\delta = \gamma/300$, $r = 3000$, and $d = r^2$. To achieve $p|\mathcal{R}| < 1/4$, we require $|D|^{\delta m_w r^2/\gamma - m_w/\gamma - \frac{3}{2} r m_w} > 4(e/\gamma)^{\frac{3}{2} m_w}$. Using $|D| = 3$ and factoring out m_w, it is sufficient if we have $3^{\delta r^2/\gamma - 1/\gamma - \frac{3}{2} r} > 4(e/\gamma)^{\frac{3}{2}}$. With our choice of parameters, this is satisfied for $r \geq 3000$.

For Lemma 3, we also require assumptions (2) and (3). First for (2): $|D|^{m_r} \geq |D|^{m_w}(1/2 - 2\gamma)^{m_w}$. For $|D| = 3$ and $m_w = 2m_r$, this is satisfied. For (3) we require: $r \leq 1/4(1/2 - \gamma)^{m_r}|D|^{m_r - \delta n}/s = 1/4(1/2 - \gamma)^{m_r}|D|^{m_r(1-2\delta/\gamma)}/s$. For $|D| = 3$, $\gamma = .01$, $\delta = \gamma/300$, we have $(1/2 - \gamma)|D|^{(1-2\delta/\gamma)} \geq 1.45$ and thus it suffices to show $r \leq 1/4(1.45)^{m_r}/s$. This holds for our choice $r = 3000$ when $s \leq 2^{cm_r} = 2^{cn/(2\gamma)}$ for some $c > 0$ and sufficiently large n.

4 Conclusion

We have proved an exponential lower bound on the size of non-deterministic semantic read once branching programs computing a polynomial time computable function $f : D^n \to \{0,1\}$ when $|D| = 3$. Our contribution is that we bring down the size of the domain required to achieve this. Prior to our result the best that was known was for D-ary branching programs with $|D| \geq 2^{13}$. The explicit function f for which we show the lower bound is the decision problem of determining whether a certain degree d polynomial over a finite field K evaluates to a value less than a certain threshold at a given input. However, interestingly the case where D is boolean $\{0,1\}$ still remains open and no non-trivial lower bounds are known for binary non-deterministic semantic read once branching programs [10].

Acknowledgements

We thank Paul Beame and Siu Man Chan for helpful discussions.

This work was supported by the Natural Science and Engineering Research Council of Canada.

References

[1] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings 40th FOCS*, pages 60–70, 1999.

[2] P. Beame, T. Jayram, and M. Saks. Time-space tradeoffs for branching programs. *J. Comput. Syst. Sci*, 63(4):542–572, 2001.

[3] P. Beame, M. Saks, X. Sun, and E. Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, 2003.

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

[5] A. Cobham. The recognition problem for the set of perfect squares. In *Switching and Automata Theory, 1966., IEEE Conference Record of Seventh Annual Symposium on*, pages 78–87. IEEE, 1966.

[6] L. Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time space tradeoffs for satifiability. In *Proceedings 12th Conference on Computational Complexity*, pages 52–60, 1997.

[7] L. Fortnow and D. V. Melkebeek. Time-space tradeoffs for nondeterministic computation. In *Proceedings 15th Conference on Computational Complexity*, pages 2–13, 2000.

[8] S. Jukna. A note on read-k times branching programs. *Informatique théorique et applications*, 29(1):75–83, 1995.

[9] S. Jukna. A nondeterministic space-time tradeoff for linear codes. *Information Processing Letters*, 109(5):286–289, 2009.

[10] S. Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.

[11] S. P. Jukna. The effect of null-chains on the complexity of contact schemes. In *Fundamentals of Computation Theory*, pages 246–256. Springer, 1989.

[12] M. Krause, C. Meinel, and S. Waack. Separating the eraser turing machine classes le, nle, co-nle and pe. In *Mathematical Foundations of Computer Science 1988*, pages 405–413. Springer, 1988.

[13] R. Lipton and A. Viglas. Time-space tradeoffs for sat. In *Proceedings 40th FOCS*, pages 459–464, 1999.

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

[15] P. Pudlak and S. Zak. Space complexity of computations. *Preprint Univ. of Prague*, 1983.