A nondeterministic space-time tradeoff for linear codes

S. Jukna*†

To appear in IPL

Abstract

We are interested in proving exponential lower bounds on the size of *nondeterministic* D-way branching programs computing functions $f : D^n \to \{0,1\}$ in linear time, that is, in time at most kn for a constant k. Ajtai has proved such lower bounds for explicit functions over domains D of size about n, and Beame, Saks and Thathachar for functions over domains of size about 2^{2^k}. We prove an exponential lower bound $2^{\Omega(n/c^k)}$ for an explicit function over substantially smaller domain D of size about 2^k. Our function is a universal function of linear codes.

1 Introduction

We consider functions $f : D^n \to \{0,1\}$, where D is a finite domain. A standard model to compute such functions $f(x_1,\ldots,x_n)$ is that of *deterministic* branching programs, called also D-way branching programs. Such a program is a directed acyclic graph with a unique start node. Each non-sink node is labeled by a variable and the edges out of a node correspond to the possible values of the variable. Each sink node is labeled by 0 or 1. 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 edges to follow. The output of such a computation is the label of the sink node reached. If $D = \{0,1\}$ then the program is called *boolean*.

The *nondeterminism* can be introduced by allowing so-called guessing nodes. These nodes are unlabeled and have an arbitrary out-degree. If a computation reaches such a node, then it can proceed further by following any of the outgoing edges. Such a program accepts an input vector if and only if at least one path from the source to a 1-sink is consistent with this input. The *size* of a branching program is the number of non-guessing nodes. The logarithm of this number gives the space required to compute a given function.

If we put no further restrictions on the branching programs, then the best remains the lower bound $\Omega(n^2/\log^2 n)$ for nondeterministic boolean branching programs proved by Nechiporuk in [9]. Exponential lower bounds were only proved under additional restrictions on the structure of branching programs; see [11] or the monograph [13] for a comprehensive survey.

In this paper we are interested in proving large lower bounds on the size of branching programs when the computation time is bounded by kn for some constant k. More precisely, we say that a program computes a given function f in *time* T if for every input $a \in f^{-1}(1)$ there is a path from the source to a 1-sink which is consistent with a and along which at most T tests are made.

Important here is that the restriction concerns only *consistent* paths, that is, paths along which no two tests $x_i = d_1$ and $x_i = d_2$ for $d_1 \neq d_2$ are made. The "syntactic" case, where we require that along *all* paths—be they consistent or not—at most kn tests can be made, is easier to deal with and exponential lower bounds are known even for $D = \{0,1\}$ and for nondeterministic branching programs [10, 5, 6].

The boolean "non-syntactic" case is more difficult. In this case, exponential lower bounds were first proved for *deterministic* branching programs working in time $T \leq n + o(n/\log n)$ [12, 8], then

*Research supported by a DFG grant SCHN 503/4-1.

†Address: Institute of Mathematics, Akademijos 4, LT-80663 Vilnius, Lithuania

for deterministic programs working in time $T \leq n + \epsilon n$ for a very small (but constant!) $\epsilon > 0$ [3, 7], and finally, for deterministic programs working in time $T \leq kn$ for any constant k [2]; this was extended to randomized branching programs in [4].

The situation with *nondeterministic* branching programs is much worse. In the boolean case, when $D = \{0, 1\}$, no exponential lower bounds are known even for programs working in time $T > n$. Such bounds were only proved for functions working on large domains, namely – when $|D|$ is either linear in n [1], or is about 2^{2^k} [3].

In this paper we do this for a substantially smaller domain containing about 2^k elements. As a domain D we take a Galois field $\mathrm{GF}(q)$ with q about 2^k. The function $g(Y, \vec{x})$ itself has $n^2 + n$ variables, the first n^2 of which are arranged in an $n \times n$ matrix Y. The values of the function are defined by $g(Y, \vec{x}) = 1$ iff the vector $\vec{x}$ is orthogonal over $\mathrm{GF}(q)$ to all rows of Y. In other words, $g(Y, \vec{x}) = 1$ iff the vector $\vec{x}$ belongs to a linear code defined by the parity-check matrix Y.

Theorem 1. *For every $k \geq 1$ and every prime power $q \geq 2^{3k+10}$, every nondeterministic branching program computing $g(Y, \vec{x})$ in time kn must have size exponential in $\Omega\left(n/k^2 4^k\right)$.*

The time restriction in this theorem concerns only the last n variables—the first n^2 variables from Y can be tested an arbitrary number of times.

Like in [5] and in subsequent papers, our goal is to show that, if the size of a branching program is small, then it must accept all vectors of a large "rectangle". Given a set X of variables, an *m-rectangle* is a set of vectors $R \subseteq D^X$ of the form $R = R_0 \times \{w\} \times R_1$, where $R_0 \subseteq D^{X_0}$ and $R_1 \subseteq D^{X_1}$ for some pair of disjoint m-element subsets X_0 and X_1 of X. Note that every m-rectangle can have at most $|D|^{2m}$ vectors.

A function $f : D^n \to \{0, 1\}$ is a *code function* if any two accepted vectors differ in at least two coordinates. The only property of such functions we will use is that in any branching program computing such a function, along any accepting computation each variable must be tested at least once.

The density of $f : D^n \to \{0, 1\}$ is $\mu(f) = |f^{-1}(1)|/|D|^n$.

Lemma 1. *If a code function $f : D^n \to \{0, 1\}$ can be the computed by a nondeterministic branching program of size s working in time kn, then for every $m \leq n/2^{k+1}$ the function accepts all vectors of some m-rectangle $R = R_0 \times \{w\} \times R_1$ of size*

$$|R| \geq \frac{\mu(f)}{(2s)^r \binom{n}{m}^2} \cdot |D|^{2m}, \tag{1}$$

where $r = 8k^2 2^k$.

2 Proof of Lemma 1

For each input $a \in f^{-1}(1)$, fix one accepting computation path $comp(a)$, and split it into r sub-paths $p_1, \ldots, p_r$ of length at most $\ell = kn/r$; the length of a sub-path p_i is the number of tests made along it. That is, we have r *time segments* $1, \ldots, r$, and in the i-th of them the computation on a follows the sub-path p_i.

Say that two inputs a and b in $f^{-1}(1)$ are equivalent if the starting nodes of the corresponding sub-paths $comp(a) = (p_1, \ldots, p_r)$ and $comp(b) = (q_1, \ldots, q_r)$ coincide. Since we have at most s nodes in the program, the number of possible equivalence classes does not exceed s^r. Fix some largest equivalence class $A \subseteq f^{-1}(1)$; hence,

$$|A| \geq |f^{-1}(1)|/s^r .$$

We say that a pair of disjoint subsets of variables X_0 and X_1 is *good* for a set of vectors B if there is a coloring of time segments $1, \ldots, r$ in red and blue such that, along each computation $comp(a) = (p_1, \ldots, p_r)$ on a vector $a \in B$, the variables from X_0 are tested only in red and those from X_1 only in blue sub-paths.

Claim 1 ([3])**.** *Let $r = 8k^2 2^k$. Then for every vector $a \in f^{-1}(1)$, at least one pair of disjoint m-element subsets of variables with $m \geq n/2^{k+1}$ is good for a.*

Proof. For a variable $x \in X$, let d_x be the number of sub-paths in $comp(a) = (p_1, \ldots, p_r)$ along which this variable is tested. Since the computed function $f(X)$ is a code function, we know that each variable $x \in X$ is tested at least once along $comp(a)$. Since the program computes $f(X)$ in time kn, we also know that at most kn tests can be made along the whole computation $comp(a)$. Hence, $\sum_{x \in X} d_x \leq kn$, implying that average number $\sum_{x \in X} d_x/n$ of tests made on a single variable does not exceed k. Finally, we know that each sub-path can make at most $\ell = kn/r$ tests.

Color the sub-paths $p_1, \ldots, p_r$ red or blue uniformly and independently. Call a variable $x \in X$ red (resp., blue) if all sub-paths testing this variable are red (resp., blue). This way, each variable is red as well as blue with probability 2^{-d_x}. Hence, we can expect

$$\sum_{x \in X} 2^{-d_x} \geq n \Big(\prod_{x \in X} 2^{-d_x} \Big)^{1/n} = n 2^{-\sum_x d_x/n} \geq n 2^{-k}$$

red variables as well as at least $n2^{-k}$ blue variables. Using the Chebyshev inequality it is not difficult to show (see Lemma 12 in [3]) that then at least one coloring must produce at least $m \geq (1-\delta)n2^{-k}$ red variables *and* at least so many blue variables, where $\delta = \sqrt{k\ell 2^{1+k}/n} = \sqrt{k^2 2^{1+k}/r} = \sqrt{1/4} = 1/2$. $\square$

We have only 2^r possible colorings of time intervals $1, \ldots, r$, and at most $\binom{n}{m}^2$ pairs of disjoint m-element subsets of variables. Hence, by Claim 1, some of these pairs X_0, X_1 must be good for a subset $B \subseteq A$ of size

$$|B| \geq \frac{|A|}{2^r \binom{n}{m}^2}\,.$$

We can write each vector $a \in D^n$ as $a = (a_0, w, a_1)$, where a_0 is the projection of a onto X_0, a_1 is the projection of a onto X_1, and w is the projection of a onto $X \setminus (X_0 \cup X_1)$. Say that two vectors $a = (a_0, w, a_1)$ and $b = (b_0, w', b_1)$ are equivalent if $w = w'$. Since the sets of variables X_0 and X_1 are disjoint, each equivalence class is a rectangle.

Let $R \subseteq B$ be a largest equivalence class lying in B; hence

$$\begin{aligned} |R| &\geq \frac{|B|}{|D|^{n-2m}} \geq \frac{|A|}{2^r \binom{n}{m}^2 |D|^{n-2m}} \\ &\geq \frac{|f^{-1}(1)|}{s^r 2^r \binom{n}{m}^2 |D|^{n-2m}} = \frac{\mu(f)}{(2s)^r \binom{n}{m}^2} \cdot |D|^{2m}\,. \end{aligned}$$

So, it remains to show that all vectors of the rectangle R are accepted by the program. This is a direct consequence of the following more general claim.

Claim 2. *If both vectors $a = (a_0, w, a_1)$ and $b = (b_0, w, b_1)$ belong to B, then the combined vector (a_0, w, b_1) belongs to A.*

Proof. Let $comp(a) = (p_1, \ldots, p_r)$ be an accepting computation on $a = (a_0, w, a_1)$, and $comp(b) = (q_1, \ldots, q_r)$ an accepting computation on $b = (b_0, w, b_1)$. Consider the combined vector $c = (a_0, w, b_1)$. Our goal is to show that then $p_t(c) \vee q_t(c) = 1$ for all $t = 1, \ldots, r$. That is, that for each $t = 1, \ldots, r$, the combined vector c must be accepted by (must be consistent with) at least one of the sub-paths p_t or q_t.

To show this, assume that c is not accepted by p_t. Since p_t accepts the vector $a = (a_0, w, a_1)$, and this vector coincides with the combined vector $c = (a_0, w, b_1)$ on all the variables outside X_1, this means that at least one variable from X_1 must be tested along p_t. But then, by the goodness of the pair X_0, X_1, no variable from X_0 can be tested along the sub-path q_t. Since q_t accepts the vector $b = (b_0, w, b_1)$, and the combined vector $c = (a_0, w, b_1)$ coincides with this vector on all the variables outside X_0, the sub-path q_t must accept the vector c, as desired.

This completes the proof of Claim 2, and thus the proof of Lemma 1. $\square$

3 Proof of Theorem 1

Fix an arbitrary prime power $q \geq 2^{3k+10}$, and let $d = m + 1$ where $m := \lfloor n/2^{k+1} \rfloor$. By the Gilbert–Varshamov bound, linear codes $C \subseteq \mathrm{GF}(q)^n$ of distance d and size $|C| \geq q^n/V(n,m)$ exist, where

$$V(n,m) = \sum_{i=0}^{m} (q-1)^i \binom{n}{i} \leq dq^m \binom{n}{m}$$

is the number of vectors in a Hamming ball of radius m around a vector in $\mathrm{GF}(q)^n$.

Let Y be the parity-check matrix of such a code, and consider the function $f : \mathrm{GF}(q)^n \to \{0,1\}$ such that $f(\vec{x}) = 1$ iff $Y \cdot \vec{x} = \vec{0}$. That is, $f(\vec{x}) = 1$ iff $\vec{x} \in C$. The function $f(\vec{x})$ is a sub-function of $g(Y, \vec{x})$. Hence, if the function $g(Y, \vec{x})$ can be computed by a nondeterministic branching program working in time kn, then the size of this program must be at least the size s of a nondeterministic branching program computing $f(\vec{x})$ in time kn. To finish the proof of Theorem 1, it remains therefore to show that s must be exponential in m/r, where $r = 8k^2 2^k$ is from Lemma 1.

The function $f(\vec{x})$ has density $\mu(f) = 1/V(n,m)$. Hence, by Lemma 1, the code C must contain an m-rectangle $R = R_0 \times \{w\} \times R_1$ of size

$$\begin{aligned} |R| &\geq \frac{\mu(f)}{(2s)^r \binom{n}{m}^2} \cdot q^{2m} = \frac{q^{2m}}{(2s)^r \binom{n}{m}^2 V(n,m)} \\ &\geq \frac{q^m}{(2s)^r d \binom{n}{m}^3} . \end{aligned} \tag{2}$$

On the other hand, since the Hamming distance between any two vectors in C is at least $d = m+1$, none of the sets R_0 and R_1 can have more than one vector. Hence, $|R| \leq 1$. Remembering that $m = \lfloor n/2^{k+1} \rfloor$ and $q \geq 2^{3k+10}$ this, together with (2) and

$$\binom{n}{m}^3 \leq \left(\frac{en}{m}\right)^{3m} \leq (2^{3k+9})^m \leq (q/2)^m ,$$

implies that $(2s)^r \geq 2^m/d = 2^{\Omega(m)}$, and the desired lower bound $s = 2^{\Omega(m/r)} = 2^{\Omega(n/k^2 4^k)}$ follows.

4 Conclusion

We have proved an exponential lower bound on the size of *nondeterministic* branching programs computing explicit function $f : D^n \to \{0,1\}$ in time $T = o(n \log n)$. Our contribution is that the bound holds for a function working over much smaller domain D than those considered in [1] and [3]. However, the *boolean* case (where $D = \{0,1\}$) remains open: in this case no non-trivial lower bounds are known even for $T \leq (1+\epsilon)n$ for an arbitrary small constant $\epsilon > 0$.

Even worse, no exponential lower bounds are known for read-once(!) switching networks. A switching network is just a directed acyclic graph whose edges are labeled by variables and their negations (see, e.g., [11]). A vector $a \in \{0,1\}^n$ is accepted iff it is consistent with all the labels of at least one path from the source to a sink. A network is read-once if, along any consistent path each variable is tested at most once. Important here, again, is that the restriction only concerns *consistent* paths—along paths, containing a variable and its negation, each variable may appear many times. As noted in [8], such networks seem to be the weakest nondeterministic model for which no nontrivial lower bounds are known.

References

[1] M. Ajtai, Determinism versus non-determinism for linear time RAMs with memory restrictions, J. Comput. Syst. Sci. 65 (2002) 2–37.

[2] M. Ajtai, A non-linear time lower bound for boolean branching programs, Theory of Comput. 1 (2005) 149–17.

[3] P. W. Beame, T. S. Jayram,[1] M. Saks, Time-space tradeoffs for branching programs, J. Comput. Syst. Sci. 63(4) (2001) 542–572.

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

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

[6] S. Jukna, A note on read-k-times branching programs, Theoret. Informat. and Appl. 29(1) (1995) 75–83.

[7] S. Jukna, Expanders and time-restricted branching programs, Theoret. Comput. Sci., DOI 10.1016/j.tcs.2008.09.012.

[8] S. Jukna, A. Razborov, Neither reading few bits twice nor reading illegally helps much, Discrete Appl. Math. 85 (1998) 223-238.

[9] E. I. Nečiporuk, On a Boolean function, Soviet Math. Doklady 7(4) (1966) 999–1000.

[10] E. A. Okolnishnikova, Lower bounds for branching programs computing characteristic functions of binary codes, Metody discretnogo analiza 51 (1991) 61–83 (in Russian).

[11] A. A. Razborov, Lower bounds for deterministic and nondeterministic branching programs, in: Lecture Notes in Comput. Sci., vol. 529, Springer, Berlin, 1991, pp. 47–60.

[12] P. Savický, S. Žák, A lower bound on branching programs reading some bits twice, Theor. Comput. Sci. 172(1-2) (1997) 293-301.

[13] I. Wegener, *Branching Programs and Binary Decision Diagrams*, SIAM, 2000.

[1] Formely Jayram S. Thathachar