Electronic Colloquium on Computational Complexity, Report No. 61 (2012)

Formulas are exponentially stronger than monotone circuits in non-commutative setting

Pavel Hrubeš* Amir Yehudayoff†

Abstract

We give an example of a non-commutative monotone polynomial f which can be computed by a polynomial-size non-commutative formula, but every monotone non-commutative circuit computing f must have an exponential size. In the non-commutative setting this gives, a fortiori, an exponential separation between monotone and general formulas, monotone and general branching programs, and monotone and general circuits. This answers some questions raised in [6].

1 Introduction

Albegraic complexity investigates the complexity of computing polynomials over fields. The basic and standard models of computation are arithmetic circuits, branching programs, and formulas (circuits being the computationally strongest and formulas the weakest). The general goal is to understand computations performed by these algebraic devices. The main open problem is to prove strong complexity lower bound for explicit polynomials.

In the *non-commutative* setting, computations are weaker than in the (more common) commutative setting, in that a device may not rely on commutativity of variables during the computation. It computes a non-commutative polynomial over a given field—one can imagine a polynomial whose variables take matrix values. The main motivation for the study of this model is that, while we do not know how to prove strong lower bounds in the commutative setting, we may have better luck with the (easier, in this respect) non-commutative one. Indeed, in his seminal paper [6], Nisan proved an exponential lower bound for the size of non-commutative branching programs (and hence formulas). A super-polynomial lower bound for non-commutative circuit-size, however, remains an open problem.

A *monotone* arithmetic circuit is a circuit that uses only non-negative real numbers. Valiant showed that general circuits are exponentially more powerful

*Department of Mathematics, University of Calgary, Calgary, Canada. pahrubes@gmail.com.

†Department of Mathematics, Technion-IIT, Haifa, Israel. amir.yehudayoff@gmail.com.
Horev fellow – supported by the Taub foundation. Research supported by ISF and BSF.

ISSN 1433-8092

than monotone circuits [9]. This is an analog of the gap between monotone and general boolean circuits [8]. In the aforementioned paper, Nisan asks whether such a separation also holds in the non-commutative setting. The expected answer is “yes.” A proof of such a statement should not be out of reach: while we do not know how to prove strong non-commutative circuit lower bounds, proving lower bounds for monotone circuits is fairly straightforward. In this text, we show that the answer is indeed positive.

Separations of this flavor have been obtained before. In [4], the authors gave a super-polynomial gap between non-commutative formulas and non-commutative monotone formulas. In [5], Li gave a similar separation for (some complexity measure related to) algebraic branching programs. Our result is stronger.

We prove an exponential separation between non-commutative *formulas* and non-commutative monotone *circuits*. Commutative or not, monotone or not, circuits are at least as powerful as algebraic branching programs, and those are in turn are at least as powerful as formulas. The separation proved here, therefore, implies similar separations for any of these classes. Hence, in this context, our result is as strong as possible.

Our construction is inspired by the separation between rank and non-negative rank given in Fiorini et al. [2], which in turn uses Razborov’s bound on the distributional communication complexity of disjointness [7]. See Section 4 for more details.

1.1 Notation

A *non-commutative polynomial* over a field is a polynomial in which variables are not assumed to multiplicatively commute. Two standard models of non-commutative computation we consider are non-commutative arithmetic circuits and formulas. These models were investigated, e.g., in [6, 3], where we refer the reader for exact definitions.

A non-commutative polynomial f over the field of real numbers is *monotone* if every coefficient in f is non-negative. Similarly a non-commutative arithmetic circuit is *monotone*, if it uses non-negative real numbers only. Since we are only dealing with non-commutative computation, we shall often drop the “non-commutative” adjective.

When considering boolean vectors, we use the following notation. Let $u = (u(1),\dots,u(n))$, $v = (v(1),\dots,v(n))$ be two vectors in $\{0,1\}^n$. Define $u \wedge v = (u(1)v(1),\dots,u(n)v(n)) \in \{0,1\}^n$ and $|u| = u(1)+\cdots+u(n) \in \mathbb{N}$. Interpreting u, v as subsets of $\{1,\dots,n\}$, $u \wedge v$ is the intersection of u and v, and $|u|$ is the size of u. We also write $i \in u$ instead of $u(i) = 1$. The concatenation of two general vectors u, v is denoted uv.

We consider polynomials in only two variables, x_0 and x_1. For $u \in \{0,1\}^n$, let x_u be the monomial

$$x_u = x_{u(1)}x_{u(2)}\cdots x_{u(n)}\,.$$

For a polynomial f and a monomial x_u, write

$$x_u \in f$$

if the coefficient of x_u in f is non-zero. A polynomial f is *homogeneous of degree* n, if $u \in \{0,1\}^n$ for every monomial $x_u \in f$.

1.2 Statement of results

The polynomial D_n is the monotone homogeneous degree-$2n$ polynomial defined by

$$D_n = \sum_{u,v \in \{0,1\}^n} (|u \wedge v| - 1)^2 x_{uv} \, .$$

The following two theorems summarize the main statements proved in this paper.

Theorem 1. *The polynomial D_n can be computed by a non-commutative formula of size $O(n^3)$.*

Theorem 2. *Every monotone non-commutative circuit computing D_n has size at least $2^{\Omega(n)}$.*

The rest of the text is mainly devoted to proving two theorems. In Section 2 we prove the upper bound on formula complexity, and in Section 3 we prove the lower bound on monotone circuit complexity.

2 Formula complexity of D_n

Proof of Theorem 1. Write

$$D_n = \sum_{u,v} |u \wedge v|^2 x_{uv} - 2 \sum_{u,v} |u \wedge v| x_{uv} + \sum_{u,v} x_{uv} \, ,$$

where u, v range over $\{0,1\}^n$. Let

$$f_1 = \sum_{u,v} |u \wedge v|^2 x_{uv}, \quad f_2 = \sum_{u,v} |u \wedge v| x_{uv} \quad \text{and} \quad f_3 = \sum_{u,v} x_{uv} \, .$$

It is sufficient to show that each of f_1, f_2, f_3 can be computed by a formula of cubic size.

First, $f_3 = (x_0 + x_1)^{2n}$ has a formula of size $O(n)$. Second, we claim that

$$f_2 = \sum_{i \in \{1,\dots,n\}} g_i g_i$$

where

$$g_i = (x_0 + x_1)^{i-1} x_1 (x_0 + x_1)^{n-i}.$$

This is because, for every u, v, the coefficient of x_{uv} in $g_i g_i$ is one if $i \in u \wedge v$ and it is zero otherwise. Hence the coefficient of x_{uv} in $\sum_i g_i g_i$ is exactly $|u \wedge v|$. This gives an $O(n^2)$-size formula for f_2. Third, we claim that

$$f_1 = \sum_{i,j \in \{1,\ldots,n\}} g_{ij} g_{ij}$$

where

$$g_{ij} = \begin{cases} (x_0 + x_1)^{i-1} x_1 (x_0 + x_1)^{j-i-1} x_1 (x_0 + x_1)^{n-j} & i < j, \\ g_{ij} = g_{ji} & j > i, \\ g_{ij} = g_i & j = i. \end{cases}$$

Again, the coefficient of x_{uv} in g_{ij} is one if $i, j \in u \wedge v$, and it is zero otherwise. Hence the coefficient of x_{uv} in $\sum_{i,j} g_{ij} g_{ij}$ is the number of pairs $i, j \in u \wedge v$. This is exactly $|u \wedge v|^2$. We thus obtained a formula of size $O(n^3)$ computing f_1. $\square$

3 Monotone circuit complexity of D_n

We now show that D_n requires monotone circuits of exponential size. The lower bound uses only the two following properties of D_n: for every $u, v \in \{0,1\}^n$,

(i). if $|u \wedge v| = 0$ then $x_{uv} \in D_n$, and

(ii). if $x_{uv} \in D_n$ then $|u \wedge v| \neq 1$.

The proof consists of two steps summarized by Lemmas 4 and 5 below. The first lemma is a known structural representation of monotone non-commutative circuits. The second lemma heavily relies on Razborov's lower bound on the distributional communication complexity of disjointness [7]. We now phrase (a weaker version of) his result. For $A, B \subseteq \{0,1\}^n$, let

$$\mu(A \times B) = \Big| \big\{ (u,v) \in A \times B : |u| = |v| = n/4,\ |u \wedge v| = 0 \big\} \Big|.$$

Let $\mu(n) = \mu(\{0,1\}^n \times \{0,1\}^n)$.

Lemma 3 (Razborov). *Assume that $A, B \subseteq \{0,1\}^n$ are such that for every $u \in A, v \in B$, it holds that $|u \wedge v| \neq 1$. Then $\mu(A \times B) \leq 2^{-\Omega(n)} \cdot \mu(n)$.*

3.1 The two lemmas

The first lemma requires a definition. We call a homogeneous polynomial f of degree k *central*, if there exist non-negative integers p_1, q, p_2 with $p_1 + q + p_2 = k$ and $k/3 < q \leq 2k/3$ so that

$$f = g_1 h \bar{g}_1 + \cdots + g_m h \bar{g}_m$$

where h, $g_1, \ldots, g_m$, $\bar{g}_1, \ldots, \bar{g}_m$ are homogeneous polynomials of degrees $\deg h = q$, $\deg g_1 = \ldots = \deg g_m = p_1$ and $\deg \bar{g}_1 = \ldots = \deg \bar{g}_m = p_2$. No bound on m is assumed.

Lemma 4. *Assume that a homogeneous polynomial f of degree $k \geq 2$ can be computed by a monotone circuit of size s. Then there exists $t = O(ks)$ and monotone central polynomials $f_1, \ldots, f_t$ such that $f = f_1 + \cdots + f_t$.*

Proof. The proof is a straightforward adaptation of Proposition 3.2 in [3]. We get $O(ks)$ instead of $O(k^3 s)$ because we do not need to homogenize a monotone circuit. $\square$

The second lemma requires definitions too. Let $\Lambda_p(n)$ denote the set of vectors $u_1 u_2 \ldots u_p v_1 v_2 \ldots v_p \in \{0,1\}^{2pn}$ such that for every $i \in \{1, \ldots, p\}$,

$$u_i, v_i \in \{0,1\}^n, \quad |u_i| = |v_i| = n/4 \quad \text{and} \quad |u_i \wedge v_i| = 0\,.$$

Clearly,

$$|\Lambda_p(n)| = \mu(n)^p. \tag{1}$$

For a homogeneous polynomial f of degree $2pn$, define

$$\Lambda_p(f) = \{u \in \Lambda_p(n) : x_u \in f\}\,.$$

Observe

$$\Lambda_p(n) = \Lambda_p(D_{pn}), \tag{2}$$

that is, $\Lambda_p(D_{pn})$ is largest possible.

Lemma 5. *Assume that f is a central polynomial of degree $6n$ such that for every $u, v \in \{0,1\}^{3n}$, if $x_{uv} \in f$ then $|u \wedge v| \neq 1$. Then $|\Lambda_3(f)| \leq 2^{-\Omega(n)} |\Lambda_3(n)|$.*

The lemma is proved below. We first show that Lemmas 4 and 5 give Theorem 2.

Proof of Theorem 2. Assume[1] that n is divisible by twelve. Consider D_{3n} under the assumption that n is divisible by four. Assume that D_{3n} can be computed by a monotone circuit of size s. By Lemma 4, we have $t = O(ns)$ and monotone central polynomials $f_1, \ldots, f_t$ such that $D_{3n} = f_1 + \cdots + f_t$. Since $\Lambda_3(\sum_i f_i) \subseteq \bigcup_i \Lambda_3(f_i)$,

$$|\Lambda_3(D_{3n})| \leq |\Lambda_3(f_1)| + \cdots + |\Lambda_3(f_t)|.$$

By (2), we know $|\Lambda_3(D_{3n})| = \Lambda_3(n)$. Hence,

$$t \max_i |\Lambda_3(f_i)| \geq |\Lambda_3(n)|.$$

For every $u, v \in \{0,1\}^{3n}$, if $x_{uv} \in D_{3n}$ then $|u \wedge v| \neq 1$. Since $f_1, \ldots, f_t$ are monotone, the same must hold for every f_i. Hence, by Lemma 5, $\max_i |\Lambda_3(f_i)| \leq 2^{-\Omega(n)} |\Lambda_3(n)|$. Since n is divisible by four, $|\Lambda_3(n)| \neq 0$. So, $t \geq 2^{\Omega(n)}$ and consequently $s \geq 2^{\Omega(n)}$. $\square$

[1] This assumption is without loss of generality. When n is not divisible by twelve, we can restrict a few variables in the polynomial and the circuit to be zero and obtain a polynomial in $12\lfloor n/12 \rfloor$ variables. This is not completely obvious, since we are working with only two variables, so restricting the circuit requires some thought. There are several ways to handle this, e.g., one can "order" the circuit as in [3]. This may cause a (negligible) increase in size by a factor of order n.

3.2 Proof of Lemma 5

Before entering the proof, let us give few more definitions. A vector $\sigma \in \{0,1,\star\}^k$ is called a *restriction*. We are interested in restrictions of a specific form, namely

$$\sigma = w_1 \star^{p_1} w_2 \star^{p_2} w_3 \,,$$

where $w_i \in \{0,1\}^{q_i}$, $\star^p$ is a vector of p stars, and $p_1+p_2+q_1+q_2+q_3 = k$. The *degree* of σ is p_1+p_2. Such a σ *acts* on a degree-k homogeneous polynomial g: If g is written as

$$g = \sum_{v_1,u_1,v_2,u_2,v_3} a(v_1,u_1,v_2,u_2,v_3) x_{v_1 u_1 v_2 u_2 v_3} \,,$$

where $a(v_1,u_1,v_2,u_2,v_3) \in \mathbb{R}$, and the summation ranges over all $v_i \in \{0,1\}^{q_i}$ $u_j \in \{0,1\}^{p_j}$, then

$$\sigma(g) = \sum_{u_1,u_2} a(w_1,u_1,w_2,u_2,w_3) x_{u_1 u_2},$$

where $u_j \in \{0,1\}^{p_j}$. One can thing of σ as picking out the monomials in g that are compatible with σ, and shrinking them.

We can see that

(i). $\sigma(g)$ is a homogeneous polynomial of degree equal to the degree of σ (this includes the case $\sigma(g) = 0$),

(ii). if g_1, g_2 are homogeneous of degree k then $\sigma(g_1+g_2) = \sigma(g_1)+\sigma(g_2)$, and

(iii). if g_1, g_2 are homogeneous of degree ℓ and $k-\ell$ respectively, then $\sigma(g_1 g_2) = \sigma_1(g_1)\sigma_2(g_2)$, where $\sigma_1 \in \{0,1,\star\}^\ell$ and $\sigma_2 \in \{0,1,\star\}^{k-\ell}$ are the (unique) restrictions such that $\sigma = \sigma_1\sigma_2$.

Proof of Lemma 5. Since f is central, write

$$f = \sum_{i=1}^{m} g_i h \bar{g}_i,$$

with $k = 6n$. Assume, w.l.o.g., that $p_1 \geq p_2$ (if $p_2 > p_1$ the argument is symmetric).

We use the following simple claim. For two integers a, b, denote by $(a, b]$ the half-closed interval $(a, b] = \{c \in \mathbb{Z} : a < c \leq b\}$.

Claim. *There exists* $e \in \{0,1\}$ *such that* $(en, (e+1)n] \subseteq (0, p_1]$ *and* $((e+3)n, (e+4)n] \subseteq (p_1, p_1+q]$.

Proof. It is basically a case analysis. We have $p_1+q+p_2 = 6n$ and $2n < q \leq 4n$. Since $p_1 \geq p_2$,

$$p_1 \geq (6n-q)/2 \geq (6n-4n)/2 = n, \quad p_1 \leq 6n - q \leq 4n$$

and

$$p_1 + q \geq (6n-q)/2 + q \geq 4n.$$

- If $p_1 \leq 3n$, set $e = 0$. In this case, the above inequalities imply $(0, n] \subseteq (0, p_1]$ and $(3n, 4n] \subseteq (p_1, p_1 + q]$.
- If $p_1 > 3n$, set $e = 1$. In this case, $p_1 + q \geq 3n + 2n$, and the above inequalities imply $(n, 2n] \subseteq (0, p_1]$ and $(4n, 5n] \subseteq (p_1, p_1 + q]$.

$\square$

In the rest of the proof, we fix a particular $e \in \{0, 1\}$ that satisfies the Claim. For $z = u_1u_2v_1v_2 \in \Lambda_2(n)$, let $\sigma_z \in \{0, 1, \star\}^{6n}$ be the restriction

$$\sigma_z = \begin{cases} \star^n u_1 u_2 \star^n v_1 v_2 & e = 0, \\ u_1 \star^n u_2 v_1 \star^n v_2 & e = 1. \end{cases}$$

Let

$$f(z) = \sigma_z(f).$$

Since σ_z has degree $2n$, $f(z)$ is a homogeneous polynomial of degree $2n$. For every $u_3, v_3 \in \{0, 1\}^n$,

$$x_{u_3v_3} \in f(z) \quad \text{iff} \quad \begin{cases} x_{u_3u_1u_2v_3v_1v_2} \in f & e = 0, \\ x_{u_1u_3u_2v_1v_3v_2} \in f & e = 1. \end{cases}$$

If $x_{uv} \in f$, then by assumption $|u \wedge v| \neq 1$. Since $z = u_1u_2v_1v_2 \in \Lambda_2(n)$, we have $|u_1 \wedge v_1| = |u_2 \wedge v_2| = 0$. Hence, no matter what e is,

$$\text{if } x_{u_3v_3} \in f(z) \text{ then } |u_3 \wedge v_3| \neq 1. \tag{3}$$

Similarly,

$$|\Lambda_3(f)| = \sum_{z \in \Lambda_2(n)} |\Lambda_1(f(z))| \leq |\Lambda_2(n)| \max_{z \in \Lambda_2(n)} |\Lambda_1(f(z))| \,. \tag{4}$$

We now want to estimate $\max_{z \in \Lambda_2(n)} |\Lambda_1(f(z))|$. Fix $z = u_1u_2v_1v_2 \in \Lambda_2(n)$. First, we claim that there exist homogeneous polynomials h_1, h_2, each of degree n, such that

$$f(z) = h_1 h_2 \,. \tag{5}$$

Since f is central,

$$\sigma_z(f) = \sum_i \sigma_1(g_i)\sigma_2(h)\sigma_3(\bar{g}_i),$$

where $\sigma_1 \in \{0, 1, \star\}^{p_1}$, $\sigma_2 \in \{0, 1, \star\}^q$ and $\sigma_3 \in \{0, 1, \star\}^{p_2}$ are restrictions such that $\sigma_z = \sigma_1\sigma_2\sigma_3$. By definition of σ_z and choice of e, the restrictions $\sigma_1, \sigma_2, \sigma_3$ have degrees $n, n, 0$ respectively. Hence, $\sigma_1(g_1), \ldots, \sigma_1(g_m)$ and $\sigma_2(h)$ are homogeneous polynomials of degree n, and $\sigma_3(\bar{g}_1), \ldots, \sigma_3(\bar{g}_m)$ are constant polynomials. So,

$$\sigma(f) = \sum_i \sigma_1(g_i)\sigma_2(h)\sigma_3(\bar{g}_i) = \left(\sum_i \sigma_1(g_i)\sigma_3(\bar{g}_i)\right)\sigma_2(h).$$

Equation (5) follows.

Let

$$A = \{u \in \{0,1\}^n : x_u \in h_1\} \text{ and } B = \{v \in \{0,1\}^n : x_v \in h_2\}.$$

Equation (5) means that $x_{uv} \in f(z)$ iff $u \in A$ and $v \in B$. Therefore,

$$|\Lambda_1(f(z))| = \mu(A \times B).$$

From (3), $|u \wedge v| \neq 1$ for every $u \in A, v \in B$. Lemma 3, hence, implies $\mu(A \times B) \leq 2^{-\Omega(n)}\mu(n)$ and so

$$|\Lambda_1(f(z))| \leq 2^{-\Omega(n)}\mu(n)\,.$$

Finally, by (4) and (1), we obtain

$$|\Lambda_3(f)| \leq |\Lambda_2(n)| 2^{-\Omega(n)}\mu(n) = 2^{-\Omega(n)}\mu(n)^3 = 2^{-\Omega(n)}|\Lambda_3(n)|\,.$$

□

4 A comment about non-negative rank

In [6], Nisan has pointed out that in order to separate non-commutative monotone and general branching programs, it is sufficient to separate the rank and the *non-negative rank* of a matrix. The non-negative rank of a $m \times n$ non-negative real matrix M is the smallest k so that M can be written as $M = AB$, where A and B are non-negative matrices of dimension $m \times k$ and $k \times n$. This concept was introduced by Yannakakis in [10], where it was related to the complexity of linear programming, and it has several other interesting applications.

The question how much can the rank and the non-negative rank differ is quite intriguing. It is relatively straightforward (see [1]) to construct an $n \times n$ matrix M whose rank is 3 but the non-negative rank is $\Omega(\log n)$. While this separation is very strong when comparing just the values of the two ranks (constant versus non-constant), it is much less so when taking into account the dimension of the matrix. For example, it is not known whether there exists M whose rank is constant but the non-negative rank is linear (in its dimension). A better separation was obtained by Fiorini et al. in [2], which we now outline. Let M be the $2^n \times 2^n$ matrix whose rows and columns are labelled with vectors in $\{0,1\}^n$ and

$$M_{u,v} = (|u \wedge v| - 1)^2\,, \text{ for all } u, v \in \{0,1\}^n\,.$$

The authors of [2] showed that the rank of M is $O(n^2)$, whereas its non-negative rank is $2^{\Omega(n)}$. In their argument too, the lower bound follows from Razborov's result about disjointness [7].

This rank separation gives, almost immediately, an exponential gap between general and monotone branching programs computing the polynomial D_n. It cannot, however, be directly extended to a monotone circuit lower bound. Our separation requires a deeper study of the structure of non-commutative circuits. Though, ultimately, the combinatorial essence is Razborov's result, our use of it is, indeed, more elaborate than in [2].

References

[1] L. Beasley and T. Laffey, Homogeneous formulas and symmetric polynomials. *Linear Algebra and its Applications*, 431(12):2330–2335, 2009.

[2] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *CoRR*, abs/1111.0837, 2011.

[3] P. Hrubeš, A. Wigderson, and A. Yehudayoff. Non-commutative circuits and the sum of squares problem. *J. Amer. Math. Soc.*, 24:871–898, 2011.

[4] P. Hrubeš and A. Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011.

[5] Y. D. Li. Applications of monotone rank to complexity theory. ECCC, TR11-025, 2012.

[6] N. Nisan. Lower bounds for non-commutative computation. In *Proceeding of the 23th STOC*: 410–418, 1991.

[7] A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

[8] E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[9] L. G. Valiant. Negation can be exponentially powerful. *Theoretical Computer Science*, 12:303–314, 1980.

[10] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de