Electronic Colloquium on Computational Complexity, Report No. 137 (2009)

Random CNFs require spacious Polynomial Calculus refutations

Massimo Lauria *
lauria@di.uniroma1.it

December 14, 2009

Abstract

We study the space required by Polynomial Calculus refutations of random k-CNFs. We are interested in how many monomials one needs to keep in memory to carry on a refutation. More precisely we show that for $k \geq 4$ a refutation of a random k-CNF of Δn clauses and n variables requires monomial space $\Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$ with high probability. For constant Δ we prove that monomial space complexity is $\Theta(n)$ with high probability. This solves a problem left open in Alekhnovich et al. (STOC, 2000) and in Ben-Sasson, Galesi (FOCS, 2001; Random Struct. Algorithms, 2003).

We study the *twofold matching game*: it is a prover-delayer game on a bipartite graph in which the prover wants to show that the left side has no pair of disjoint matching sets on the right side. The prover has a bounded amount of memory. We show that any delayer's winning strategy against such prover is also a strategy to satisfy all equations in a bounded memory polynomial calculus refutation.

We show that a random k-CNF with $k \geq 4$ has large enough expansion with high probability. This allows lower bounds on the memory of a winning prover in the corresponding twofold matching game. A lower bound on the monomial space required to refute the formula follows.

We claim without proof that our result also applies to pigeonhole principles on bipartite graphs.

*Dipartimento di Informatica, Sapienza - Università di Roma.

ISSN 1433-8092

1 Introduction

Proof complexity studies the computational feasibility of logical proofs. This study has strong theoretical motivations: it is easy to prove membership for an NP language; on the other hand proving that propositional formula is a tautology is a co-NP complete task. Cook and Reckhow [17] outline the following program to show NP≠co-NP (and P≠NP): show that increasingly strong proof systems require super polynomial size proofs for some tautologies. Eventually we may understand how to prove it for any proof system. Several lower bounds have been proved: contradictions based on Pigeonhole principle [21, 8, 25], randomly distributed CNF [15, 7, 13] and parity principles [28] require exponential size refutations in the famous Resolution proof system.

There are also practical motivations for proof complexity theory. SAT solvers and DPLL algorithms of widespread usage are in fact flavours of Resolution. With these applications in mind it is worth to study resources beyond proof length. How much memory is required to carry on the refutation? How complex proof lines must be? Notice that the latter question influences the efficiency of proof search techniques. In this paper we ignore the important topic of proof search, we just focus on proof existence.

In propositional proof complexity it is customary to study the dual of proving tautologies: we want to prove that a formula in conjunctive normal form (CNF) is unsatisfiable (i.e. we want to refute such a formula). In this setting the clauses of the CNF are added to the proof system as axioms. If we deduce an explicit contradiction it means that the CNF is unsatisfiable.

We study *Polynomial Calculus* (PC) proof system: proof lines are polynomial equations; a refutation in PC is a deduction of $1 = 0$ from a set of polynomials with no common roots. It is easy to encode CNF unsatisfiability in this framework. Polynomial Calculus has been introduced [6, 16] as a preliminary effort to study systems where proof lines are $AC^0[p]$ circuits. It is also worth to mention that Gröbner basis algorithms [18] for solving polynomial equations implicitly define PC refutations. So there are both theoretical and practical motivations for the study of this proof system.

Lower bounds for PC are mostly focused on the degree which must appear in a refutation of a given unsatisfiable CNF. Degree lower bounds have been proved for pigeonhole principles [24, 22, 4], random CNFs [10, 4] and ordering principles [20]. The study of degree has two merits: (1) the higher the degree requirement, the bigger the search space for Gröbner basis algorithms; (2) there is a relation between minimal size S of a PC refutation and its minimum degree D: $D \leq O(\sqrt{n \log S})$ where n is the number of variables [10, 3]. Several size lower bounds are obtained using this relation. Unfortunately this technique cannot be improved [20].

Space complexity is well studied in Resolution. It has been introduced in [19] and some lower bounds has been given for specific formulae [27, 2]. Ben-Sasson and Galesi [9] obtain lower bounds for random k-CNFs with $k \geq 3$. Their technique has been extended in [5] to show that width lower bounds imply space lower bounds. Recently the works [11, 12] clarified several other aspects of space in Resolution.

Very little is known about space complexity of Polynomial Calculus. There are some lower bounds for CNFs with many variables in each clause, like the pigeonhole principle [2]. This is unsatisfactory because the most interesting cases have small clauses. Our paper tries to increase the poor understanding of space in Polynomial Calculus.

1.1 Our results

We model the space complexity in Polynomial Calculus as the number of monomials in memory at each stage of a refutation. We show that for $k \geq 4$ a refutation of a random k-CNFs on n variables with Δn clauses requires monomial space $\Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$ with high probability. We get as corollary that for constant Δ the space complexity is $\Theta(n)$ with high probability. In order to prove such result we show a lower bound for the memory required to win the *twofold matching game*, and we show that the monomial space is at least such quantity. We claim without proof that our result also applies to the pigeonhole principle on bipartite graphs. We comment about a possible relation between monomial space complexity and degree complexity in polynomial calculus refutations.

1.2 Paper organization

In Section 2 we give basic definitions of Polynomial Calculus systems, random CNFs, matchings and bipartite expanders. In Section 3 we introduce the twofold matching game, a tool we use to prove the main theorem of the paper. In Section 4 we prove the monomial space lower bounds for random k-CNF with $k \geq 4$. In Section 5 we discuss the behaviour of degree and space in some alternative polynomial calculi for which our techniques do not seem to work. Some open problems are left in Section 6.

2 Preliminaries

We denote the set of integers $\{i : 1 \leq i \leq u\}$ as $[u]$. We denote variables as $x_1, x_2 \ldots$ and $y_1, y_2, \ldots$; variables are always assumed to have values in some algebraic representation of the booleans. Literals x and $\neg x$ (also denoted as $\bar{x}$) evaluate respectively to the value of x and to its negation. A clause is a disjunction of literals $l_1 \vee l_2 \ldots$ which is true if at least one of the literals is true. A *conjunctive normal form* (CNF) formula $\bigwedge_j C_j$ is a conjunction of clauses which is true when all of them are true. We usually index variables with i and clauses with j. A k-CNF is a CNF in which all clauses have at most k literals. Usually we consider propositional formulae on n variables and m clauses. In this paper we also deal with sequences of memory configurations: usually t denotes the length of the sequence and s is used as index.

2.1 Algebraic proof systems

In our proof systems we use algebraic reasoning to prove facts on propositional formulae. In this context any line in the proof is a polynomial equation. In a refutation the aim is to deduce the trivial algebraic contradiction $1 = 0$.

Polynomial Calculus (PC) has been defined in [16]. It is a refutational system based on the polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ for a fixed field $\mathbb{F}$. We always consider equations of the form $p = 0$, and we simply denote them as p. The equations are intended to hold on $\{0,1\}^n$ thus the system contains the following axioms:

$$x_i^2 - x_i, \quad i \in [n]$$

There are two inference rules. For any $\alpha, \beta \in \mathbb{F}$ and polynomials p, q:

$$\frac{p \qquad q}{\alpha p + \beta q} \quad \textit{Sum Rule} \qquad \frac{p}{pq} \quad \textit{Product Rule}$$

A PC proof of a polynomial g from a set of initial polynomials $f_1, \ldots, f_m$ (denoted by $f_1, \ldots, f_m \vdash g$) is a sequence of polynomials where each one is either an f_j, an axiom of the system or is obtained by inference rules from previous polynomials.

PC is sound and complete, in the sense that $f_1, \ldots, f_m \vdash g$ if and only if $g(\vec{x}) = 0$ for every $\vec{x} \in \{0,1\}^n$ which is also a common zero of $f_1, \ldots, f_m$. A PC refutation is a proof that $f_1, \ldots, f_m \vdash 1$, which is possible if and only if $f_1, \ldots, f_m$ have no common zeros. Completeness of PC comes as a corollary of Hilbert's Nullstellensatz (see [18]) or from complete algorithms based on Gröebner basis (see [16, 18]).

Consider a PC proof Π: the *degree* of Π, $deg(\Pi)$, is the maximal degree of a polynomial in the proof; the *size* of Π, $S(\Pi)$, is the number of monomials in the proof, the *length* of Π, $|\Pi|$, is the number of equations in the proof. The degree, size, length complexity of an unsatisfiable set ϕ of polynomial equations are respectively the smallest degree, size, length possible for a PC refutation of ϕ. In case of satisfiable ϕ we say that degree, size, length complexity are ∞.

Polynomial Calculus with Resolution (PCR) [3] is a refutational system which extends PC to polynomials in the ring $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$, where $\bar{x}_1, \ldots, \bar{x}_n$ are new formal variables. PCR includes the axioms and rules of PC plus a new set of axioms

$$1 - x_i - \bar{x}_i \quad i \in [n]$$

to force $\bar{x}$ variables to have the opposite values of x variables. We extend to PCR the definitions of proof, refutation, degree, size and length given for PC. Observe that using the linear transformation $\bar{x} \mapsto 1 - x$, any PCR refutation can be converted into a PC refutation without increasing the degree. Notice that such transformation could cause an exponential increase in size. Moreover any Resolution refutation can be easily transformed in a PCR refutation of degree equal to the width of the original one, and comparable size.

Polynomial Calculus in Fourier Basis is a variant of PC we want to briefly mention: the boolean values true and false are mapped respectively to -1 and $+1$. This proof system has the same rules of PC, but the axioms are

$$x_i^2 = 1 \quad i \in [n]$$

Notice that equations in Fourier basis can be converted in classic PC equations by the mapping $x \mapsto \frac{x+1}{2}$. This mapping can cause an exponential blow up in size, too.

Clause encoding. Our main interest is to refute CNFs. In PC and PCR the usual convention is to represent *true* with 0 and *false* with 1. Literals x and $\neg x$ are represented respectively as x and $(1 - x)$ and the disjunction is represented as multiplication. Notice that a clause with many negated literals has exponential size representation in PC. PCR has been introduced for this reason: $\neg x$ can be represented by the algebraic variable $\bar{x}$ and any clause can be represented as a monomial equation. In the paper we almost ignore Fourier basis, which has not even an efficient encoding for propositional clauses.

Multilinear Polynomial Calculus. In all algebraic interpretations of the booleans shown so far, any monomial is logically equivalent to a square free one. Such equivalence can be proved by using an axiom rule, a product rule and a sum rule, and it is possible to efficiently remove the squares from a proof as soon as they appear. In the standard basis this process is equivalent to map $x^c \mapsto x$ for any $c > 0$. In the Fourier base this is equivalent to map $x^{2n+b} \mapsto x^b$ for $b \in \{0,1\}$. This "multilinearization" process can be done explicitly with negligible increase in proof length, size and degree. We denote as η the appropriate multilinearization map for the proof system in hand and we substitute the product rule with the following

$$\frac{p}{\eta(pq)} \quad \textit{Multilinear Product Rule}$$

It is an easy fact that any refutation in our algebraic systems can be efficiently turned into a square free refutation with the aid of the multilinear product rule.

2.2 Monomial Space complexity of a refutation

To study the memory requirements for a refutation we study **the number of distinct monomials contained in memory** at any step of the computation. We are ignoring the fact that there could be several copies of the same monomial in memory. Since we are interested in lower bounds this makes our results stronger. We view a refutation Π as a sequence of configurations $M_0, M_1, \ldots M_t$ which are sets of polynomials; $M_0 = \emptyset$ and every M_{s+1} follows from M_s by one of the following rules:

DOWNLOAD $M_{s+1} = M_s \cup \{p\}$ where p is either an axiom of the proof system or the algebraic encoding of a clause in the CNF to refute.

ERASURE $M_{s+1} = M_s - \{p\}$ for some $p \in M_i$.

INFERENCE $M_{s+1} = M_s \cup \{p\}$ where p is obtained with an application of either multilinear product rule or sum rule to some polynomials **contained in** M_s.

We say that a sequence of memory configurations proves p if $p \in M_t$. We define $Space(M_s)$ as the monomial space complexity of a configuration M_s, which is the number of distinct monomials contained in all polynomials in M_s. The monomial space complexity of a refutation $\Pi = (M_0, \ldots, M_t)$ is $Space(\Pi) = \max_{s=1}^{t} Space(M_s)$. The monomial space complexity of a CNF ϕ, denoted as $Space(\phi)$, is the smallest monomial space complexity of a PCR refutation of ϕ. It ϕ is satisfiable we say it is ∞. In this paper we are mostly interested in studying the value of $Space(\phi)$ for random ϕ. We restrict our attention to square free refutations. Notice that the use of multilinear product rule instead of standard product rule reduces at most by half the space requirements, so this is not an issue for our results. Also consider that no reasonable implementation of PC or PCR would implement multilinearization as an explicit inference.

2.3 Random k-CNFs

We consider CNFs over boolean variables $x_1 \ldots x_n$. For any CNF there is a natural bipartite *underlying graph* which expresses the variables involved in each constraint. We define a distribution on random formulae and the corresponding distribution on the underlying bipartite graphs.

Definition 1. *Fix $k > 0$ and $\Delta > 0$. We define $\mathcal{F}^k_{\Delta n,n}$ to be the distribution over k-CNFs $C_1 \wedge \ldots \wedge C_{\Delta n}$ where each C_j is distributed according to the following probabilistic process: choose uniformly at random a subset of k variables $\{y_1 \ldots y_k\} \subseteq \{x_1 \ldots x_n\}$. For $1 \leq i \leq k$ fix l_i to be either y_i or $\neg y_i$ with uniform probability. Set $C_j := l_1 \vee \ldots \vee l_k$.*

Definition 2. *Fix $k > 0$ and $\Delta > 0$. We define a distribution $\mathcal{G}^k_{\Delta n,n}$ on bipartite graphs $G = (L, R, E)$ with $L = [\Delta n]$ and $R = [n]$. Each vertex $j \in L$ has a set chosen uniformly at random of k distinct neighbours in R.*

It is clear that the underlying graph of a CNF distributed according to $\mathcal{F}^k_{\Delta n,n}$ is itself distributed according to $\mathcal{G}^k_{\Delta n,n}$. Such distribution of formulae is a well studied model (see [15, 7, 13, 10, 9, 26, 1] among many others)

2.4 Matching and Expansion

We define important structural concepts about bipartite graphs. A *partial matching* of a bipartite graph $G = (L, R, E)$ is a subset $E' \subseteq E$ such that $G' = (L, R, E')$ has degree at most one. A *perfect matching* is a partial matching with no isolated vertices. We say that some set

$U \subseteq L \cup R$ can be matched if there is a partial matching in which all elements of U have degree one. For $U \subseteq L \cup R$ we define its neighbourhood to be $N(U) = \{v : v \notin U, \{u, v\} \in E\}$. For a single vertex $u \in L \cup R$ we set $N(u) := N(\{u\})$. The most important and useful theorem in the context of matching is

Hall's Matching Theorem. *For a bipartite graph $G = (L, R, E)$, a set of vertices $L' \subseteq L$ can be matched if and only if for any $L'' \subseteq L'$ it holds that $|N(L'')| \geq |L''|$.*

The following is an easy generalization of Hall's Theorem.

k-fold Matching Theorem. *Fix a bipartite graph $G = (L, R, E)$, and a set of vertices $L' \subseteq L$. A k-fold matching for L' is a subset $E' \subseteq E$ such that in $G' = (L, R, E')$ all vertices in L' have degree k and vertices in R have degree at most 1. L' can be k-fold matched if and only if for any $L'' \subseteq L'$ it holds that $|N(L'')| \geq k|L''|$.*

Proof. The condition is clearly necessary. For the sufficiency consider a graph H with k copies of the left side $L_1 \cup L_2 \cup \ldots L_k$ and one copy of R. Consider the k copies of L'. Consider any subset $U \subseteq L'_1 \cup L'_2 \cup \ldots \cup L'_k$, and assume wlog that $|U \cap L'_1| \geq \frac{|U|}{k}$. Then we have

$$|N(U)| = |N(U \cap L'_1) \cup \ldots \cup N(U \cap L'_k)| \geq |N(U \cap L'_1)| \geq k|U \cap L'_1| \geq |U|$$

In the second inequality we used the condition on the size of the neighbourhood. By Hall's Matching Theorem $L'_1 \cup \ldots \cup L'_k$ has a matching in H. This means that L' has a k-fold matching in G. □

The expansion on bipartite graphs is also deeply related with matchings.

Definition 3. *We say that a bipartite graph $G = (L, R, E)$ is an (r, c)-expander when for any set $L' \subseteq L$ of size at most r it holds that $|N(L')| \geq c|L'|$.*

Hall's Matching Theorem pairs with expander graph definition: any set of at most r left vertices has a matching if and only if the graph is an $(r, 1)$-expander, and has a k-fold matching if and only if the graph is an (r, k)-expander.

3 Twofold Matching game

We describe a game we are going to use to prove the monomial space lower bound for random k-CNF refutations in PCR. **Diana (the delayer)** claims that a formula is satisfiable and **Paula (the prover)** challenges such claim. Paula asks questions to Diana and she must answer consistently. Paula wins the game if she eventually catches Diana in contradiction. Diana wins if she has a strategy for playing infinitely long without being caught.

To study space complexity in proof systems we enforce that Paula can remember a bounded amount of previous answers. Diana is caught only if she is inconsistent with answers which Paula remembers. We show a relation between the amount on memory Paula needs to win the game and the number of monomials needed to be kept in memory in a PCR refutation.

Twofold Matching Game. Consider a bipartite graph $G = (L, R, E)$: the status of the game at step s is encoded by a graph $G_s = (L, R, E_s)$ where E_s is a subset of E. We set $E_0 = \emptyset$. Paula has two options at each step s:

- Paula asks some $l \in L$ which is **isolated** in G_{s-1}. Diana must answer with two distinct vertices $r_1, r_2 \in R$ which are both neighbours of l in G but are isolated in G_{s-1}. If no such vertices exist then Diana **loses** the game, otherwise G_s is updated by setting $E_s := E_{s-1} \cup \{\{l, r_1\}, \{l, r_2\}\}$.

- Paula removes some edges from G_{s-1}. More precisely she chooses $E_s \subseteq E_{s-1}$ to be the edge set of G_s.

Original Matching Game. The game is defined in [9]. It is similar to the twofold matching game. The only difference is that Diana has to answer with just one neighbour to each of Paula's requests.

A gameplay of t moves on a graph G characterizes a sequence of subgraphs $G_0, \ldots, G_t$ of G. The *matching complexity* of this gameplay is $\max_{s=0}^{t} |E_s|$. The matching complexity of G is the smallest complexity of a winning gameplay for Paula on G: we denote it as $MSpace(G)$ for the original matching game and $TMSpace(G)$ for the twofold matching game. There are graphs (e.g. the bipartite complete $K_{n,2n}$) for which Diana can play consistently for infinitely long and there is no winning gameplay for Paula. In such cases we say that the (twofold) matching complexity is infinite.

Lemma 1. *For any $k \geq 4$ and $0 < \epsilon \leq 1/2$ a random graph G distributed according to $\mathcal{G}^k_{\Delta n,n}$ requires twofold matching space complexity $\Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$ with probability $1 - o(1)$.*

Proof. We are interested in the expansion properties of the graph G. Thus we use the following Expansion Lemma which is a variant of Lemma 11 in [7] and Lemma 5.1 in [9].

> **Expansion Lemma** (See proof in Appendix)**.** *For each $k \geq 4$ and $0 < \epsilon \leq 1/2$, there exist a constant c depending only on k and ϵ such that the following holds: a random graph distributed according to $\mathcal{G}^k_{\Delta n,n}$ is a $\left(cn\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}}, 2+\epsilon\right)$-expander.*

From now on we fix $r = cn\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}}$ as in the Expansion Lemma. We assume G to be an $(r, 2+\epsilon)$-expander. We are going to prove that

$$TMSpace(G) \geq \frac{\epsilon}{4+\epsilon} r$$

Define $G' = (L_1 \cup L_2, R, E_1 \cup E_2)$ where $L_i = \{u_i : u \in L\}$ and $E_i = \{\{u_i, v\} : u \in L, \{u, v\} \in E\}$ for $i \in \{1, 2\}$. G' is essentially a bipartite graph with two copies of L and one of R.

We show G' is an $(r, 1 + \frac{\epsilon}{2})$-expander: consider any sets $A_1 \subseteq L_1$ and $A_2 \subseteq L_2$ such that $|A_1 \cup A_2| \leq r$. We may assume without loss of generality that $u_2 \in A_2$ implies $u_1 \in A_1$ because u_1 and u_2 can be exchanged. It follows that $|A_2| \leq |A_1|$ and $N(A_2) \subseteq N(A_1)$. So we have that

$$|N(A_1 \cup A_2)| = |N(A_1)| \geq (2+\epsilon)|A_1| \geq (1 + \frac{\epsilon}{2})|A_1 \cup A_2|$$

We now apply the following theorem:

> **Theorem** (see [9])**.** *Let be G' and $(r, 1+\alpha)$-expander, then $MSpace(G') \geq \frac{\alpha}{2+\alpha} r$.*

The twofold matching game on G is essentially a special case of the original matching game on G' in which Paula always asks u_1 and u_2 consecutively. This means that a winning strategy for Diana for original matching game on G' is also a winning strategy for twofold matching game for G. So we get

$$TMSpace(G) \geq MSpace(G') \geq \frac{\epsilon/2}{2+\epsilon/2} r = \frac{\epsilon}{4+\epsilon} r$$

□

4 Space complexity for Random k-CNFs

The main theorem of this section and of the paper is a bound on the number of monomials that any PCR refutation of random k-CNFs must keep in memory simultaneously.

Theorem 1. *For any $k \geq 4$ and $\Delta > 0$, a random k-CNF distributed according to $\mathcal{F}^k_{\Delta n,n}$ has monomial space complexity $\Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$ with probability $1 - o(1)$.*

Theorem 1 easily follows from the next lemma which we prove later.

Lemma 2. *For any $F = \bigwedge_{j=1}^{m} C_j$ in conjunctive normal form on variables $\{x_1, \ldots, x_n\}$, consider the bipartite graph $G = ([m], [n], E)$ where $\{j, i\} \in E$ if and only if variable x_i occurs in clause C_j. Then*

$$Space(F) \geq \frac{TMSpace(G)}{4}$$

Proof of Theorem 1. Fix $m = \Delta n$, and pick a random formula

$$F = \bigwedge_{1 \leq j \leq m} C_j$$

distributed according to $\mathcal{F}^k_{m,n}$. Consider the bipartite graph $G = ([m], [n], E)$ where $E = \{\{j, i\} : x_i \text{ occurs in } C_j\}$. Notice that G is distributed according to $\mathcal{G}^k_{\Delta n,n}$ thus

$$TMSpace(G) = \Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$$

holds with probability $1-o(1)$ because of Lemma 1. Then the lower bound for $Space(F)$ follows because of Lemma 2. □

Corollary 1. *Fix $k \geq 4$ and constant $\Delta > 0$. A random k-CNF distributed according to $\mathcal{F}^k_{\Delta n,n}$ has monomial space complexity $\Theta(n)$ with probability $1 - o(1)$.*

Proof. The lower bound is a direct application of Theorem 1. The upper bound is a standard tree like refutation of size 2^n. For any variable x we will denote the literals x and $\bar{x}$ as x^1 and x^0 respectively. In this notation $\prod_{i=1}^{n} x_i^{\alpha_i} = 1$ if and only $\alpha \in \{0,1\}^n$ is assigned to $\{x_1 \ldots x_n\}$.

We now show that for any $0 \leq l \leq n$ and for any fixed α we can deduce $\prod_{i=1}^{l} x_i^{\alpha_i} = 0$ with at most $n - l + 6$ monomials simultaneously in memory. The equation corresponding to $l = 0$ is intended to be $1 = 0$.

If $l = n$ then the monomial follows by an application of the product rule to some clause of F falsified by α. This require two monomials in memory. If $l < n$ then by assume we can deduce $x_1^{\alpha_1} \cdots x_l^{\alpha_l} x_{l+1} = 0$ and $x_1^{\alpha_1} \cdots x_l^{\alpha_l} \bar{x}_{l+1} = 0$ with $n-l+5$ monomials in memory each. We deduce the former and we keep it in memory, erasing any intermediate step. We then deduce the latter in the same space. The whole process requires space $n-l+6$. Now we clear the memory to have only 2 monomials. By using axiom $1 - x_{l+1} - \bar{x}_{l+1}$ and never more that 6 distinct monomials in memory, we can deduce $x_1^{\alpha_1} \cdots x_l^{\alpha_l} = 0$ from those. By backward induction the statement is true for $l = 0$. □

The rest of the section will be devoted to prove the Lemma 2.

Proof of Lemma 2. Let G be the graph underlying F as in the statement. Fix $b := TMSpace(G)$. Consider a sequence of memory configurations $M_0, \ldots, M_t$ deducible in PCR from F, such that $|M_s| < \frac{b}{4}$ for all $0 \leq s \leq t$. We will define a gameplay $G_0, \ldots, G_t$ for the twofold matching game on G and a sequences of 2-CNFs $A_0, \ldots, A_t$ such that:

- The number of clauses in A_s are less than or equal to $2|M_s|$.
- The variables in A_s are the right side vertices in G_s with degree 1.
- Any variable occurs at most once in A_s (i.e. clauses are variable disjoint).
- A_s logically implies all polynomial equations in M_s.

This will conclude the proof: all such 2-CNFs are satisfiable thus there is no empty clause in any of the M_s. We set $M_0 = \emptyset$, G_0 to have no edges and A_0 to be the empty CNF which is trivially satisfiable. From G_s and A_s with the desired properties we define A_{s+1} conditioning on the rule used to go from M_s to M_{s+1}.

ERASURE/INFERENCE: A_s implies the truth of all equations in M_s and in case of an equation erasure or of an inference rule the configuration M_{s+1} is a semantic consequences of M_s. We know that A_s already implies M_{s+1} but it could be larger than $2|M_{s+1}|$ in case of an erasure, so we need a smaller 2-CNF. We use the following Locality Lemma shown in [2].

Locality Lemma (rephrased from Lemma 4.14 in [2], see proof in Appendix)**.** *Let F be a 2-CNF with no common variables among two clauses. Let E be a set of polynomial equations containing h distinct monomials. Assume that F logically implies E. Then there exists a 2-CNF F' such that:*

- *F' has at most $2h$ clauses;*
- *variable of F' are all contained in F.*
- *no two clauses in F' share a variable;*
- *F' logically implies E;*

We choose A_{s+1} to be the 2-CNF given by the previous lemma. The variables appearing in A_{s+1} are a subset of the variables appearing in A_s, thus the removal of the edges corresponding to such variables from G_s is a legitimate move in the twofold matching game which defines G_{s+1}.

AXIOM DOWNLOAD: Consider M_{s+1} to be equal to $M_s \cup \{C_j\}$ for some j. If A_s already implies C_j we are done. Otherwise we consider all the edges appearing G_s. Those are exactly $2|A_s| \leq 4|M_s| \leq b - 4$, thus in the twofold matching game position G_s Diana is able to answer to the left side vertex j with two isolated right side vertices i_1 and i_2. The edges of G_{s+1} are $E_s \cup \{\{j, i_1\}, \{j, i_2\}\}$. Let l_1 and l_2 be the corresponding literals in C_j, then $A_{s+1} := A_s \wedge (l_1 \vee l_2)$. Clearly A_{s+1} has at most $2|M_{s+1}|$ clauses and it logically implies M_{s+1} since A_s implies M_s and $l_1 \vee l_2$ implies C_j.

□

4.1 Space bound for Pigeonhole principles on graphs

For the lack of space we omit the discussion about pigeonhole principles on bipartite graph. It is easy to see that the same reasoning used to develop our space lower bound for k-CNF can be used for pigeonhole principles with Δn pigeons and n holes on a random bipartite with minimal degree 4. The reduction from a twofold matching strategy to a monomial space lower bound is identical to the one of Theorem 1.

5 Space complexity of PC compared to Resolution

The main motivations for studying PC/PCR are that PCR is strictly stronger than Resolution yet it is not difficult to implement. Furthermore several intuitions behind lower bounds in Resolution are also useful in variants of polynomial calculus. In particular there is a resemblance between Resolution width[1] and PC degree. There seems to be a general "meta theorem": if expansion properties of formulae imply high width [15, 13] then something similar holds for PC degree [10, 4]. In particular this is true for random CNF. There is a similar trade-off between Size vs Width in Resolution [13, 14] and Size vs Degree in PC/PCR [16, 3, 20].

Notice that the Resolution **space** lower bound for random CNFs has been proved using expansion properties [9]. This is not coincidence because later it has been proved in [5] that space is always bigger than width in Resolution. Then we have this "meta-relation" in resolution

$$\text{Expansion}_R \lessapprox \text{Width}_R \leq \text{Space}_R$$

It is natural to ask if such correspondence between degree and Resolution width also extends to the relation with space complexity. In [10] it is shown that $\text{Expansion}_{PC} \lessapprox \text{Width}_{PC}$ and Theorem 1 itself hints of a possible relation $\text{Expansion}_{PC} \lessapprox \text{Space}_{PC}$. It is natural to ask whether something along the lines of $\text{Degree}_{PC} \lessapprox \text{Space}_{PC}$ holds.

5.1 Barriers to a Degree $\leq$ Space relation

In this subsection we argue that any Degree vs Space relation cannot be very simple or general. The key point of our space lower bound is the Locality Lemma which is used to compress a small memory configuration with a low complexity formula. This essentially means that monomials in memory do not convey a lot of information. This is not the case in general for PCR: an high degree monomial equation can encode a full assignment in a single unit of space. In Resolution a partial assignment requires a unit of space for each assigned variable. Let us consider an example:

Fact 1. *Fix constant $k \geq 3$, and any constant $\Delta > 0$, let G be a bipartite graph distributed according to $\mathcal{G}^k_{\Delta n,n}$. Let $b = (b_1 \ldots, b_{\Delta n}) \in \{0,1\}^{\Delta n}$ be distributed uniformly. Consider the linear system*

$$\sum_{u \in N(j)} u = b_j \pmod 2$$

It holds with high probability that

- *There exists a degree one refutations in PC over any field $\mathbb{F}$ with characteristic 2.*
- *Any such refutation requires linear space.*
- *There exists a constant space in PC over any field in* **the Fourier basis**.
- *Any such refutation requires linear degree.*

This fact follows from [10]: in characteristic 2 you just sum a subset of equations which gives $1 = 0$; the lower bounds comes because of expansion. For the Fourier basis just notice that sum modulo 2 can be simulated by multiplication. Such cases suggest a trade off between degree and space. Maybe there are formulae with refutations either of large degree and small

[1] the smallest number of literals appearing in the largest clause of a refutation

space or of small degree and large space, but not small space and degree. It is a useful remark[2] that any PC/PCR refutation which is achievable in space s and degree d can be simulated in resolution with width less than $O(sd)$.

6 Open problems

We leave several interesting open problems:

1. In Section 5 we argued that the relation between degree and space in PC is not clear. We would like to know Space $\geq$ Degree holds. There may be formulae requiring high degree but small space, and viceversa small degree and high space like black-white pebbling tautologies from [23, 11]. There may even be a formula that can be refuted both in small space and small degree, but not simultaneously. Even better there could be a smooth trade-off between those measures (i.e. the conjunction of two formulae with different space and degree complexity does not answer the question). The example given in Section 5.1 does not answer completely, since the two refutations are given in different flavours of PC.

2. We have shown space hardness for k-CNF with $k \geq 4$. We weren't able to prove the result for the most interesting case: $k = 3$. We suspect it to be true. Our technique seems insufficient because it requires $2 + \epsilon$ expansions to prove $\Omega(n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}})$ monomial space. In resolution $1 + \epsilon$ expansion is sufficient to prove $\Omega(n\Delta^{-\frac{1+\epsilon}{k-2-\epsilon}})$ clause space, thus it works for random 3-CNF. It is natural to ask whether our result can be improved to match resolution one.

3. In [7, 29] there are upper bounds for resolution clause space for a random formulae. In particular Zito [29] shows that Resolution clause space $O(n/\Delta^{\frac{1}{k-2}}$ is sufficient for random k-CNF. In the case of resolution this is somehow tight. It is natural to ask if for PCR either $O(n/\Delta^{\frac{1}{k-3}})$ is sufficient (at least when $k \geq 4$) or not. This is the dual of the previous problem.

4. PCR has been defined in [2] because PC is too inefficient for clause representation. Is it possible to prove a separation between monomial space in PC and in PCR? Maybe an exponential one? Of course the interesting cases are CNFs of constant width.

Acknowledgements

The author thanks Jakob Nordström for an introduction to the problem. He also want to thanks Eli Ben-Sasson, Nicola Galesi and Jakob Nordström for insightful discussions.

References

[1] Michael Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 251–256, 2005.

[2] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. In *STOC*, pages 358–367, 2000.

[2] It has been mentioned to the author by Eli Ben-Sasson during the "Ramsey Theory in Logic, Combinatorics and Complexity" in Bertinoro, 2009.

[3] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88, 2004.

[4] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science*, pages 190–199, 2001.

[5] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[6] P. Beame, R. Impagliazzo, J. Krajícek, T. Pitassi, and P. Pudlák. Lower bounds on hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 73:1–26, 1996.

[7] Paul Beame, Richard M. Karp, Toniann Pitassi, and Michael E. Saks. The efficiency of resolution and davis–putnam procedures. *SIAM J. Comput.*, 31(4):1048–1075, 2002.

[8] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science*, pages 274–282. IEEE, 1996.

[9] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Struct. Algorithms*, 23(1):92–109, 2003.

[10] Eli Ben-Sasson and Russell Impagliazzo. Random CNFs are hard for the polynomial calculus. In *40th Annual Symposium on Foundations of Computer Science*, pages 415–421, 1999.

[11] Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *FOCS*, pages 709–718, 2008.

[12] Eli Ben-Sasson and Jakob Nordström. Understanding space in resolution: Optimal lower bounds and exponential trade-offs. *Electronic Colloquium on Computational Complexity (ECCC)*, 16(034), 2009.

[13] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 517–526, 1999.

[14] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001.

[15] Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988.

[16] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Gröebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 174–183, 1996.

[17] Stephen A. Cook, Robert, and A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[18] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms : An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3rd edition.* Springer, 2007.

[19] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. In *STACS*, pages 551–560, 1999.

[20] Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 2010. To appear.

[21] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[22] Russell Impagliazzo, Pavel Pudlák, and Jirí Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[23] Jakob Nordström. Narrow proofs may be spacious: separating space and width in resolution. In *STOC*, pages 507–516, 2006.

[24] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, 1998.

[25] Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci.*, 1(303):233–243, 2003.

[26] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM J. Comput.*, 33(5):1171–1200, 2004.

[27] Jacobo Torán. Lower bounds for space in resolution. In *CSL*, pages 362–373, 1999.

[28] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987.

[29] Michele Zito. An upper bound on the space complexity of random formulae in resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(079), 2001.

Appendix

The following proofs have appeared elsewhere with some minor differences. We put them here for completeness.

Locality Lemma (see [2])**.** *Let F be a 2-CNF with no common variables between two clauses. Let E be a set of polynomial equations containing h distinct monomials. Assume that F logically implies E. Then there exists a 2-CNF F' such that:*

- *F' has at most $2h$ clauses;*
- *variable of F' are all contained in F.*
- *no two clauses in F' share a variable;*
- *F' logically implies E;*

Proof. We denote as $|F|$ the number of clauses in a CNF F. Consider E as a boolean function on its monomials $m_1, \ldots, m_h$. Consider the bipartite graph H where the set L of left vertices is $\{m_1, \ldots, m_h\}$ and the set R of right vertices is the set $\{C_1, \ldots, C_{|F|}\}$ of clauses of F. H contains the edge $\{m_i, C_j\}$ if and only if they share at least a variable.

Let be L_N a maximal subset of L such that $|N(L_N)| < 2|L_N|$, and $L_W = L - L_N$. We also set $R_N = N(L_N)$ and $R_W = N(L_W) - N(L_N)$. We define H_N to be the subgraph of H induced

by $L_N \cup R_N$ and H_W to be the one $L_W \cup R_W$. Observe that H_W is an $(|L_W|, 2)$-expander, because for any $|L' \subseteq L_W|$ it holds that $|N(L') - N(L_N)| \geq 2|L'|$, otherwise L_N would not be maximal. We will define two 2-CNFs F_N and F_W such that $F_N \wedge F_W$ implies E and the conditions required by the theorem.

For F_W we consider a twofold matching from L_W to R_W: it exists because of the expansion of H_W and the 2-fold Matching Theorem in Section 2.4.

Any monomial m in L_W is matched with two clauses A_m, B_m, with corresponding variables u_m and v_m both occurring in m. We consider a new clause C_m on u_m and v_m: negations on the two variables are set in such a way that C_m implies $m = 0$. We also name the two remaining variables as a_m and b_m occurring respectively in A_m and B_m. We fix

$$F_W := \bigwedge_{m \in L_W} C_m \qquad F_N := \bigwedge_{C \in R_N} C \qquad F' := F_N \wedge F_W$$

By construction we have that no variable occurs more than once in F', and all variables in F' also occur in F. So we have $|F'| = |F_N| + |F_W| = |R_N| + |L_W| = |N(L_N)| + |L_W| \leq 2|L_N| + |L_W| \leq 2h$.

We need to show that F' implies E. Fix an assignment α such that $F'(\alpha)$ is true. We now define a new assignment α' by modifying α. Consider the clauses in F which are not in F': either they have no variable in common with E (reassign such variables to satisfies them) or they come in a pair A_m, B_m for some monomial $m \in L_W$. We fix a_m and b_m to satisfy them. The new assignment α' satisfies F and thus it also satisfies E. We observe that none of these reassigned variables occurs in a monomial in L_N and any monomial $m \in L_W$ evaluates to zero in both assignments because C_m remains true. Then we get $E(\alpha) = E(\alpha')$ (i.e. satisfied). □

Expansion Lemma (see [9]). *For each $k \geq 4$ and $0 < \epsilon \leq 1/2$, there exist a constant c depending only on k and ϵ such that the following holds: a random graph distributed according to $\mathcal{G}^k_{\Delta n,n}$ is a $\left(cn\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}}, 2+\epsilon\right)$-expander.*

Proof. The probability of a set U of size i to have a neighbourhood inside a set V of size $(2+\epsilon)i$ is

$$p_i = \left(\frac{\binom{(2+\epsilon)i}{k}}{\binom{n}{k}}\right)^i \leq \left(\frac{(2+\epsilon)i}{n}\right)^{ki}$$

We upper bound the probability of the existence of such a set (i.e. the event "Fail") by union bound over all possible choices of U and V. We use that $\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b$.

$$\begin{aligned}
\mathbf{Pr}[\text{Fail}] &\leq \sum_{i=1}^{r} \binom{\Delta n}{i}\binom{n}{(2+\epsilon)i} p_i \\
&\leq \sum_{i=1}^{r} \left(\frac{e\Delta n}{i}\right)^i \left(\frac{en}{(2+\epsilon)i}\right)^{(2+\epsilon)i} \left(\frac{(2+\epsilon)i}{n}\right)^{ki} \\
&\leq \sum_{i=1}^{r} \left(a \cdot \Delta \cdot (\frac{i}{n})^{k-3-\epsilon}\right)^i \qquad \text{fix } a = e^{3+\epsilon}(2+\epsilon)^{k-2-\epsilon} \qquad (*)
\end{aligned}$$

Fix $r = cn\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}}$ and $c = \left(\frac{1}{2a}\right)^{\frac{1}{k-3-\epsilon}}$. We divide the analysis in cases:

Case 1: $\Delta \geq n^{1/8}$

$$\begin{aligned}
\mathbf{Pr}[\text{Fail}] &\leq \sum_{i=1}^{r} \left(a \cdot \Delta \cdot (\frac{i}{n})^{k-3-\epsilon} \right)^i \\
&\leq \sum_{i=1}^{r} \left(a \cdot \Delta \cdot (\frac{r}{n})^{k-3-\epsilon} \right)^i \\
&\leq \sum_{i=1}^{r} \left(ac^{k-3-\epsilon} \cdot \Delta \cdot (\frac{n\Delta^{-\frac{1+\epsilon}{k-3-\epsilon}}}{n})^{k-3-\epsilon} \right)^i \\
&\leq \sum_{i=1}^{r} \left(\frac{\Delta^{-\epsilon}}{2} \right)^i \\
&\leq \sum_{i=1}^{r} \left(\frac{1}{2n^{\epsilon/8}} \right)^i \underset{n\to\infty}{\longrightarrow} 0
\end{aligned}$$

Case 2: $\Delta < n^{1/8}$ We have that for $k \geq 4$ and $\epsilon \leq 1/2$, we get $r \geq n^{5/8}$. We consider $\mathbf{Pr}[\text{Fail}] \leq X + Y$ where X is the sum of the first $\sqrt{n}$ terms of equation (*). Y is the sum of the terms for $\sqrt{n} < i \leq r$.

$$\begin{aligned}
X &\leq \sum_{i=1}^{\sqrt{n}} \left(a \cdot \Delta \cdot (\frac{i}{n})^{k-3-\epsilon} \right)^i \\
&\leq \sum_{i=1}^{\sqrt{n}} \left(a \cdot n^{1/8} \cdot n^{-\frac{k-3-\epsilon}{2}} \right)^i \\
&\leq \sum_{i=1}^{\sqrt{n}} \left(a \cdot n^{1/8} \cdot n^{-1/4} \right)^i \\
&\leq \sum_{i=1}^{\sqrt{n}} \left(a \cdot n^{-1/8} \right)^i \underset{n\to\infty}{\longrightarrow} 0
\end{aligned}$$

$$\begin{aligned}
Y &\leq \sum_{i=\sqrt{n}+1}^{r} \left(a \cdot \Delta \cdot (\frac{i}{n})^{k-3-\epsilon} \right)^i \\
&\leq \sum_{i=\sqrt{n}+1}^{r} \left(\frac{a \cdot \Delta}{2a \cdot \Delta^{1+\epsilon}} \right)^i \\
&\leq n \left(\frac{a \cdot \Delta}{2a \cdot \Delta^{1+\epsilon}} \right)^{\sqrt{n}+1} \underset{n\to\infty}{\longrightarrow} 0
\end{aligned}$$

□

ECCC ISSN 1433-8092

http://eccc.hpi-web.de