Electronic Colloquium on Computational Complexity, Report No. 79 (2001)

An Upper Bound on the Space Complexity of Random Formulae in Resolution*

M. Zito
Department of Computer Science
University of Liverpool
Chadwick Building, Peach Street Liverpool L69 7ZF
e-mail: michele@csc.liv.ac.uk

September 6, 2001

Abstract

We prove that, with high probability, the space complexity of refuting a random unsatisfiable boolean formula in k-CNF on n variables and $m = \Delta n$ clauses is $O(n \cdot \Delta^{\frac{1}{k-2}})$.

1 Introduction

The importance of studying the complexity of (propositional) proof systems comes from its close relationship with long-standing open problems in Complexity Theory such as NP =? Co-NP [5]. The complexity measure related to the classical notion of *time* is the *size* of a proof, viz. the number of *lines* used in the proof.

Recently Esteban and Torán [8] suggested a measure for the space complexity of refuting an unsatisfiable formula in a proof system called resolution (subsequent work [1] extended this notion to other proof systems). Although several results [1, 8, 13] are, by now, known on the space complexity of various classes of formulae, a quantitative analysis of the space needed to prove the unsatisfiability of random formulae has remained, until recently, somewhat elusive.

As a step towards the solution of this problem, we point out that a combination of a modification of the classical Davis-Putnam [7] algorithm and a polynomial time algorithm that produces a refutation for any given unsatisfiable 2-CNF formula, outputs refutations of any unsatisfiable random formula within the space bounds stated in the following Theorem.

Theorem 1.1 *Let ϕ be an unsatisfiable random k-CNF formula on n variables and $m = \Delta n$ clauses. There is an $a > 0$ such that, with probability approaching one as n goes to infinity, the space complexity of ϕ is at most $kan \cdot \Delta^{\frac{1}{k-2}} + O(1)$ for $\Delta \geq \Delta_a$.*

For sufficiently large $\Delta > 0$ this bound "almost" matches a lower bound proved in [2].

The remainder of this paper is organised as follows. In Section 2 we introduce all relevant notations and technical results; in Section 3 we give full details of the proof of Theorem 1.1 for the case $k = 3$ and in Section 4 we give all details needed to extend the proof to any fixed $k > 3$; Section 5 is devoted to final remarks and open problems.

*Partially supported by British Council grant ARC 4079. Part of this work was carried out while the author was visiting the Abteilung Theoretische Informatik, Fakultät für Informatik, University of Ulm.

ISSN 1433-8092

2 Preliminaries

Let a finite set of variables $X = \{x_1, \ldots, x_n\}$ be given. A *literal* is either x^0 or x^1 for any $x \in X$, although we will often follow the common practice and denote x^0 (resp. x^1) by $\neg x$ (x). We identify *clauses* with sets of literals, but we will often abuse the notation and write $x \in C$ to denote the fact that the variable x *occurs* in the clause C as either x^0 or x^1. A *formula* is a sequence of clauses $\phi = (C_1, \ldots, C_m)$. A formula is in *$k$-conjunctive normal form* (or k-CNF) if $|C_i| \leq k$ for all $i \in \{1, \ldots, m\}$. In all the subsequent treatment Δ will denote the *clause density* m/n of the given formula.

Let $\mathcal{C}^{k,n}$ denote the set of all clauses with exactly k literals of k distinct variables defined from X. A *random formula* is obtained by selecting uniformly at random, independently and with replacement m clauses from $\mathcal{C}^{k,n}$. Let $\mathcal{F}_m^{k,n}$ denote the resulting probability space on the set of all k-CNF formulae over n variables and m clauses. We will write $\phi \sim \mathcal{F}_m^{k,n}$ to signify that ϕ is obtained by the process outlined above. In all subsequent treatment we say that an event $\mathcal{E}$, depending on a parameter n, holds with high probability (w.h.p.) if it holds with probability approaching one as n tends to infinity.

A *truth-assignment* is a mapping α that assigns "false" or "true" (usually denoted by 0 or 1) to each variable in its domain $\mathrm{Dom}(\alpha)$. We write $|\alpha|$ for $|\mathrm{Dom}(\alpha)|$. Given a clause C, a variable x, and a value $\nu \in \{0, 1\}$, the *restriction of C to $x = \nu$*, $C\big|_{x=\nu}$, is C if $x \notin C$, is one if $x^\nu \in C$ and it is $C \setminus \{x^{1-\nu}\}$ otherwise. If ϕ is a formula, then $\phi\big|_{x=\nu}$ is the sequence $\left(C_1\big|_{x=\nu}, \ldots, C_{m'}\big|_{x=\nu}\right)$ for all $C \in \phi$ such that $C\big|_{x=\nu} \neq 1$. If α is a truth-assignment with domain $\mathrm{Dom}(\alpha) = \{x_{j_1}, \ldots, x_{j_t}\}$ then $C\big|_\alpha$ denotes the clause

$$(\cdots(C\big|_{x_{j_1}=\alpha(x_{j_1})})\big|_{x_{j_2}=\alpha(x_{j_2})}\cdots).$$

The meaning of $\phi\big|_\alpha$ is defined similarly. For $i \in \{0, \ldots, k\}$, let $C_i(\phi, \alpha)$ denote the set of clauses of size i in $\phi\big|_\alpha$. We say that a formula ϕ is *true* (*false*) *under the assignment* α if $\phi\big|_\alpha$ is empty ($\{\} \in \phi\big|_\alpha$). A formula is *satisfiable* if there exists a truth-assignment α (also known as *satisfying* assignment) such that ϕ is true under α.

A (*resolution*) *derivation of a clause C from* (resp. a *refutation of*) *a formula ϕ* is a sequence of clauses $\pi = (D_1, \ldots, D_t)$ such that $D_t = C$ (resp. $D_t = \{\}$) and for all $i \in \{1, \ldots, t-1\}$ either $D_i = C_j$ for some $j \in \{1, \ldots, m\}$ or D_i is obtained from D_j and D_k (with $j, k < i$) by the following *resolution rule*:

$$\{x, l_1, l_2, \ldots\}, \{\neg x, t_1, t_2, \ldots\} \to \{l_1, l_2, \ldots\} \cup \{t_1, t_2, \ldots\}$$

The two clauses to the left of "$\to$" are called *premises*, the clause to the right is called *resolvent*. The *size* of derivation π, $|\pi|$ is t. Clearly, ϕ is unsatisfiable if and only if there exists a refutation of ϕ.

Given a formula ϕ and a derivation $\pi = (D_1, \ldots, D_t)$ of some clause D_t from ϕ, a sequence of clauses $\kappa = (E_1, \ldots, E_s)$ is a *sub-derivation* of π if κ is a derivation of E_s from ϕ and for each $i \in \{1, \ldots, s\}$ there is a $j \in \{1, \ldots, t\}$ such that $E_i = D_j$.

2.1 Space complexity of derivations

Generalising from [8], a clause C has a derivation from a k-CNF formula ϕ bounded by space s if there exists a sequence of formulae $\phi_1, \ldots, \phi_t$ with

1. $\phi_1 \subseteq \phi$;
2. $|\phi_i| \leq s$, for all $i \in \{1, \ldots, t\}$;
3. ϕ_{i+1} is obtained from ϕ_i by deleting (if wished) some clauses, adding the resolvent of two clauses in ϕ_i, and adding (if wished) some clauses of ϕ.
4. $C \in \phi_t$.

$\mathrm{space}(\phi, C)$ is the minimum s for which there is a derivation of C from ϕ bounded by space s. We write $\mathrm{space}(\phi)$ instead of $\mathrm{space}(\phi, \{\})$.

Any derivation π of a clause C from a k-CNF formula ϕ can be represented as a directed acyclic graph (dag) $G_{\phi,\pi}$ in a standard way: clauses in π correspond to nodes in $G_{\phi,\pi}$, with the clauses of ϕ associated

with $G_{\phi,\pi}$'s source nodes, C associated with the (only) sink of $G_{\phi,\pi}$, and each application of the resolution rule corresponding to an internal node of $G_{\phi,\pi}$ associated with some clause D with two incoming edges leaving the nodes associated with $D_1 \cup \{x\}$ and $D_2 \cup \{\neg x\}$ respectively, with $D = D_1 \cup D_2$. A derivation is *tree-like* if its underlying dag is in fact a tree. Unless ambiguity arises, from now on π will refer to either a derivation from ϕ or its corresponding dag.

There is a nice relationship between the space needed to derive a clause C from ϕ and the number of pebbles needed in a particular pebbling game $\mathcal{G}$ played on any dag $G_{\phi,\pi}$.

> **Pebbling Game $\mathcal{G}$.** *Given a connected dag with one sink the aim of the game is to put a pebble on the sink of the graph (the only node with no outgoing edge) according to the following rules:*
>
> 1. *a pebble can be placed on any initial node;*
> 2. *a pebble can be removed from any node at any time;*
> 3. *a pebble can be placed on any internal node provided there is a pebble on all its parents.*

The following Lemma is an immediate consequence of the definitions.

Lemma 1 [8] *For any formula ϕ and clause C,* space(ϕ, C) *coincides with the minimum number of pebbles needed to win $\mathcal{G}$ on a graph $G_{\phi,\pi}$, where π is a derivation of C from ϕ.*

Using this result it is possible to analyse the space needed for refuting a formula through techniques used for bounding the number of pebbles used/needed to play $\mathcal{G}$. The following result is a consequence of Lemma 1.

Theorem 2.1 [8] *If ϕ has a tree-like refutation of size S, then* space$(\phi) \leq \lceil \log S \rceil + 1$.

From this it follows immediately that space$(\phi) \leq n + 1$ for any formula ϕ over n variables.

Although refutations of unsatisfiable k-CNF formulae for $k \geq 3$ may require non constant space, unsatisfiable 2-CNF formulae can always be refuted in constant space.

Theorem 2.2 [8] space$(\phi) = O(1)$, *for any unsatisfiable* 2-CNF *formula ϕ.*

2.2 A technical result

A fundamental conjecture about $\mathcal{F}_m^{k,n}$ states that there is a θ_k independent of n, the *unsatisfiability threshold*, such that a $\phi \sim \mathcal{F}_m^{k,n}$ is satisfiable (resp. unsatisfiable) w.h.p. if $m/n < \theta_k$ ($m/n > \theta_k$). $\theta_2 = 1$ [9] but only upper an lower bounds are known for $k \geq 3$ (see for instance [11]). The techniques used to derive these bounds turn out to be useful to prove the following result which will be used in the proof of Theorem 1.1.

Lemma 2 *Let $\phi \sim \mathcal{F}_{cn}^{2,n}$, with $c > 1$. For any $\gamma \in (0,1)$, the probability that ϕ be satisfiable is at most $(\max_{(x,y)\in D_\gamma} f_c(x,y))^n$, where $D_\gamma = \{(x,y) : \gamma \leq y \leq 1-\gamma, 1+\gamma \leq x \leq \frac{c}{y} - \gamma\}$,*

$$f_c(x,y) = \left(\frac{\mathrm{e}^{r_0}-1}{y}\left(\frac{x}{\mathrm{e}r_0}\right)^x\right)^y \left(\frac{1}{1-y}\right)^{1-y} \left(\frac{c}{2x}\right)^{xy} \left(\frac{c(3-2y)}{4(c-xy)}\right)^{c-xy},$$

and r_0 is the solution of $r\mathrm{e}^r = x(\mathrm{e}^r - 1)$.

Proof. Following [11] it is possible to bound the probability that the given random k-CNF be satisfiable by the expected number of satisfying assignments that are *maximal* in the sense that it is not possible to flip the value of any single variable set to 1 and preserve satisfiability. This number can be computed resorting to the coupon collector's probabilities, as described in [14]. The resulting upper bound on the probability that ϕ be satisfiable is

$$\sum_{s=0}^{n} \sum_{j=s}^{cn} \binom{n}{s}\binom{cn}{j} \left(\frac{s\binom{n-1}{k-1}}{2^k\binom{n}{k}}\right)^j \left(\frac{(2^k-1)\binom{n}{k} - s\binom{n-1}{k-1}}{2^k\binom{n}{k}}\right)^{cn-j} \mathrm{coupon}(j,s),$$

where coupon(j,s) is the probability that a coupon collector will pick s different coupons in j trials [12]. For $k=2$ this simplifies to

$$\sum_{s=0}^{n}\sum_{j=s}^{cn}\binom{n}{s}\binom{cn}{j}\left(\frac{s}{2n}\right)^{j}\left(\frac{3}{4}-\frac{s}{2n}\right)^{cn-j}\text{coupon}(j,s).$$

Without loss of generality we assume $s=\Theta(n)$ and $j=\Theta(n)$. Using the obvious asymptotics for the binomial coefficients, the double sum is asymptotic to

$$n^{O(1)}\left(\frac{n}{s}\right)^{s}\left(\frac{n}{n-s}\right)^{n-s}\left(\frac{cn}{j}\right)^{j}\left(\frac{cn}{cn-j}\right)^{cn-j}\left(\frac{s}{2ns}\right)^{j}\left(\frac{3n-2s}{4n}\right)^{cn-j}\text{coupon}(j,s).$$

To get the result set $y=\frac{s}{n}$ and $x=\frac{j}{s}$ and use the relevant (see for instance [4]) asymptotic for coupon(j,s). □

The function $f_c(x,y)$ can actually be maximised in D_γ by looking at the partial derivatives of $\ln f_c(x,y)$. Letting $g_1(x)=(\mathrm{e}^{r_0}-1)\left(\frac{x}{\mathrm{e}r_0}\right)^{x}$, and using the definition of r_0, we have

$$\tfrac{\mathrm{d}}{\mathrm{d}x}\ln g_1(x)=\ln\tfrac{x}{r_0}$$

Hence

$$G_y(x)=_{df}\tfrac{\partial}{\partial x}\ln f_c(x,y)=y\ln\tfrac{2(c-xy)}{r_0(3-2y)}$$

For each fixed y in the given domain, this function is defined and continuous. Moreover its sign changes for $x\in[1+\gamma,\frac{c}{y}-\gamma]$ Therefore there is[1] an $x^*=x^*(y)$ such that $G_y(x^*)=0$, and equivalently

$$\frac{2(c-x^*y)}{x^*(3-2y)}=\frac{r_0}{x^*} \tag{1}$$

Note that, from the definition of r_0, x^* must satisfy

$$x^*=x^*(r_0)=\frac{r_0\mathrm{e}^{r_0}}{\mathrm{e}^{r_0}-1}. \tag{2}$$

Let $F(y)=_{df}\ln f_c(x^*(y),y)$. The maximum of $f_c(x,y)$ will be among the critical points of $F(y)$. Notice that

$$\mathrm{d}F(y)=G_y(x^*(y))\mathrm{d}x+\tfrac{\partial}{\partial y}\ln f_c(x,y)\mathrm{d}y$$

and, since $G_y(x^*)=0$, we have $\mathrm{d}F(y)y=0$ if and only if

$$\tfrac{\partial}{\partial y}\ln f_c(x,y)=\ln\tfrac{1-y}{y}+x^*\ln\tfrac{2(c-x^*y)}{x^*(3-2y)}-\tfrac{2c-3x^*}{3-2y}+\ln g_1(x^*)=0.$$

We can rewrite $\frac{2c-3x^*}{3-2y}$ as $\frac{2(c-x^*y)}{3-2y}-x^*$ and therefore, using (1), the equation above simplifies to

$$\ln\tfrac{1-y}{y}+x^*\ln\tfrac{r_0}{x^*}-r+x+\ln(e^{r_0}-1)-x^*\ln\tfrac{r_0}{x^*}-x=0,$$

which is satisfied for

$$y^*=y^*(r_0)=\frac{e^{r_0}-1}{2\mathrm{e}^{r_0}-1}. \tag{3}$$

Hence, for a given c, one needs to find the r_0 which satisfies (1) in which x and y are replaced by expressions (2) and (3). Then the maximum of $f_c(x,y)$ is achieved at

$$x^*(r_0)=\tfrac{r_0\mathrm{e}^{r_0}}{\mathrm{e}^{r_0}-1}\quad y^*(r_0)=\tfrac{e^{r_0}-1}{2\mathrm{e}^{r_0}-1}.$$

[1] Indeed $G_y(x)$ is strictly decreasing in the given domain, so there is only one such x^*.

2.3 An algorithm

Let $Y = \{x_{j_1}, x_{j_2}, \ldots, x_{j_t}\} \subseteq X$ with t to be fixed later. For any integer $b \geq 1$ the set $Y_b = \{x_{j_b}, \ldots, x_{j_t}\}$ is called a *final segment* of Y, with the convention that $Y_1 = Y$ and $Y_b = \{\}$ if $b > t$. Consider the following modification of the classical Davis-Logemann-Loveland [6] resolution algorithm:

Function RoughDLL (ϕ: k-CNF; Y: set of variables; α: truth-assignment): boolean
  if $\phi = \{\}$ **return** true
  else if $\{\} \in \phi$ **return** false
  else if $Y = \{\}$
    return 2SAT-solver$(C_2(\phi, \alpha))$
  else
    Let x be the smallest index variable in Y;
    $Y \leftarrow Y \setminus \{x\}$;
    return RoughDLL$(\phi\big|_{x=0}, Y, \alpha \cup \{x = 0\}) \vee$
      RoughDLL$(\phi\big|_{x=1}, Y, \alpha \cup \{x = 1\})$;

where 2SAT-solver$(\ldots)$ is a function deciding 2-SAT.

If the input formula ϕ is unsatisfiable then a call to RoughDLL$(\phi, Y, \{\})$ will return the correct "false" answer provided either $\{\} \in \phi\big|_\alpha$ or $C_2(\phi, \alpha)$ is unsatisfiable, for every α with $\mathrm{Dom}(\alpha) = Y$. Furthermore, the recursive calls to RoughDLL naturally induce a rooted binary tree, $T_{\phi,Y}$, whose internal nodes are labelled by the variables that are set at a particular step, with the out-edges of a node labelled by the two possible assignments to its associated variable. Each path from the root in $T_{\phi,Y}$ corresponds to a partial assignment α with $\mathrm{Dom}(\alpha) \subseteq Y$. Each leaf is labelled by either a clause of ϕ that becomes empty or by the set $C_2(\phi, \alpha)$. We will prove that if all the formulae $C_2(\phi, \alpha)$ are unsatisfiable, $T_{\phi,Y}$ can be transformed into a refutation of ϕ, by working from the clauses labelling the leaves and the refutations obtained for each $C_2(\phi, \alpha)$ towards the root of $T_{\phi,Y}$. Theorem 1.1 will then be proved by showing that w.h.p. the refutations defined in this way can be pebbled with (relatively) few pebbles. We first need to prove a property of the formulae $C_2(\phi, \alpha)$. For any $b \geq 2$, let α_b be a truth-assignment with $\mathrm{Dom}(\alpha_b) = Y \setminus Y_b$. Let $C_{\alpha_b} = \{x_{j_1}^{1-\alpha_b(x_{j_1})}, \ldots, x_{j_{b-1}}^{1-\alpha_b(x_{j_{b-1}})}\}$. Clearly $C_{\alpha_b}\big|_{\alpha_b} = \{\}$.

Lemma 3 *Let ϕ be an unsatisfiable k-CNF formula and α a truth-assignment whose domain is included in the set of variables occuring in ϕ. Let π be a tree-like refutation of $C_2(\phi, \alpha)$. Any sub-derivation $\kappa = (E_1, \ldots, E_s)$ of π can be transformed into a tree-like derivation of a clause $C \cup E_s$ from ϕ where $C \subseteq C_\alpha$.*

Proof. Let π be a tree-like refutation of $C_2(\phi, \alpha)$. The result is proved by induction on the depth of κ, the maximum length of a path from a leaf to the root of $G_{C_2(\phi,\alpha),\kappa}$. If $\kappa \equiv C \in C_2(\phi, \alpha)$ then $\kappa' \equiv C'$ where $C' \in \phi$ and $C = C'\big|_\alpha$. Clearly C' will contain a number of literals belonging to C_α (all the literals in $C' \setminus C$).

At the inductive step, let κ'_1 and κ'_2 be the derivations associated by the induction hypothesis with the left and right subtrees κ_1 and κ_2 of κ. The derivation κ' associated with κ is obtained from κ'_1 and κ'_2 by applying the resolution rule to the clauses labelling the roots of κ'_1 and κ'_2 (these clauses will still contain the pair of complementary literals that is "removed" in the application of the resolution rule that generates the clause labelling the root of κ). □

The following result is an immediate consequence of Lemma 3 obtained by choosing $\kappa = \pi$.

Corollary 1 *Let ϕ be an unsatisfiable k-CNF formula and α a truth-assignment whose domain is included in the set of variables occuring in ϕ. Any tree-like refutation of $C_2(\phi, \alpha)$ can be trasformed into a tree-like derivation of a clause $C \subseteq C_\alpha$ from ϕ.*

The final result of this section completes the description of the refutations that, under certain conditions, can be associated with the execution of RoughDLL on any unsatisfiable formula ϕ.

Theorem 2.3 *Let ϕ be an unsatisfiable k-CNF formula whose variables belong to the set $X = \{x_1, \ldots, x_n\}$. Let $Y = \{x_{j_1}, x_{j_2}, \ldots, x_{j_t}\} \subseteq X$ for some $t \in \{1, \ldots, n\}$ and let Y_b be some final segment of Y, for a given $b \in \{1, \ldots, t\}$. Let $T_{\phi|_{\alpha_b}, Y_b}$ be the execution tree associated with* RoughDLL$(\phi|_{\alpha_b}, Y_b, \{\})$*, where α_b is some truth-assignment with* $\mathrm{Dom}(\alpha_b) = Y \setminus Y_b$*. Then $T_{\phi|_{\alpha_b}, Y_b}$ can be transformed into a tree-like derivation from ϕ of a clause $C \subseteq C_{\alpha_b}$, provided all formulae $C_2(\phi|_{\alpha_b}, \alpha')$ labelling the leaves of $T_{\phi|_{\alpha_b}, Y_b}$ are unsatisfiable, where α' is any truth assignment with* $\mathrm{Dom}(\alpha') = Y_b$*.*

Proof. The result can be proved by induction on the product between the number of nodes in $T_{\phi|_{\alpha_b}, Y_b}$ minus one and $t - b + 1$. If this product is null then $T_{\phi|_{\alpha_b}, Y_b}$ contains a single node v. If v is labelled by a clause $C \in \phi$ then the process will return C. Otherwise if v is labelled by an unsatisfiable $C_2(\phi, \alpha)$. Corollary 1 asserts that any refutation of $C_2(\phi, \alpha)$ can be transformed into a derivation π' of a clause $C \subseteq C_\alpha$ from ϕ: the process will return π'.

At the inductive step let v be the root of $T_{\phi|_{\alpha_b}, Y_b}$ and assume v is labelled by a variable $x \in Y_b$. By induction hypothesis the Theorem holds for the trees

$$T_{\phi|_{\alpha_b, x=0}, Y_b \setminus \{x\}} \text{ and } T_{\phi|_{\alpha_b, x=1}, Y_b \setminus \{x\}}.$$

Therefore there exist two derivations π_1 and π_2 of clauses $C_1 \subseteq C_{\alpha_b \cup \{x=0\}}$ and $C_2 \subseteq C_{\alpha_b \cup \{x=1\}}$. If $x \in C_1$ and $\neg x \in C_2$, a further application of the resolution rule to the clauses C_1 and C_2, resolving on the variable x generates a tree-like derivation π' of a clause $C \subseteq C_{\alpha_b}$ from ϕ. Otherwise , let π' be π_l where l is the smallest $i \in \{1, 2\}$ such that $x \notin C_i$. □

3 Analysis for $k = 3$

We analyze the space complexity of the refutations associated with RoughDLL by first considering the case $k = 3$. Since $\Delta > \theta_3$ the set of satisfiable formulae in $\mathcal{F}^{3,n}_{\Delta n}$ is very small. In the following we assume ϕ to be unsatisfiable. We will prove that it is possible to choose t so that RoughDLL ends with a "false" answer w.h.p. If this is the case, the refutation built using the algorithm in Section 2.3 is formed by joining the refutations for $C_2(\phi, \alpha)$ to the complete binary tree of depth t corresponding to the branching of RoughDLL. By Theorem 2.1 this tree can be pebbled using $t + 1$ pebbles. The result then follows from Theorem 2.2 applied to $C_2(\phi, \alpha)$ for each α.

To complete the proof note that, conditioned on the fact that $|C_2(\phi, \alpha)| = \Omega(n)$ for each α on Y, the event

"RoughDLL does not end with a 'false' answer"

is implied by the event

"there is an α with $\mathrm{Dom}(\alpha) = Y$, such that $C_2(\phi, \alpha)$ is satisfiable"

and the probability of the latter is at most

$$\sum_{\alpha : \mathrm{Dom}(\alpha) = Y} \Pr[C_2(\phi, \alpha) \in \mathrm{SAT}].$$

Let "$\psi \in$SAT" denote the event "the formula ψ is satisfiable". For each α we can compute $\Pr[C_2(\phi, \alpha) \in \mathrm{SAT}]$ conditioning on the size of $C_2(\phi, \alpha)$:

$$\Pr[C_2(\phi, \alpha) \in \mathrm{SAT}] \leq \Pr[|C_2(\phi, \alpha)| < dn] + \Pr[C_2(\phi, \alpha) \in \mathrm{SAT} \mid |C_2(\phi, \alpha)| \geq dn]$$

where $d > 0$ is some constant to be fixed later. Since clauses in ϕ are selected independently and with replacement from $\mathcal{C}^{3,n}$, given Y and α, in each of the $m = \Delta n$ trials there is a fixed probability of selecting a clause C such that $C|_\alpha$ is a 2-clause. This is exactly the probability of choosing one variable from Y with a sign fixed by the assignment α and the remaining two arbitrarily on the set $X \setminus Y$. Hence

$$\Pr[C|_\alpha \text{ is a 2-clause}] = \frac{4t\binom{n-t}{2}}{8\binom{n}{3}} = \frac{3t}{2n}\left(1 - \frac{t-1}{n-1}\right)\left(1 - \frac{t-1}{n-2}\right) > \frac{3t}{2n}\left(1 - \frac{t}{n}\right)^2$$

(where the last inequality holds as long as $t < \frac{n}{2}$). In particular if we set $\frac{t}{n} = \frac{3a}{\Delta}$ the lower bound on $\Pr[C\big|_\alpha$ is a 2-clause] becomes $\frac{9a}{2\Delta}\left(1-\frac{3a}{\Delta}\right)^2$.

These calculations imply that given Y, for each α with $\mathrm{Dom}(\alpha) = Y$, the random variable $|C_2(\phi,\alpha)|$ has a binomial distribution with parameters m and $p =_{df} \Pr[C\big|_\alpha$ is a 2-clause]. Hence, for any $\epsilon > 0$, using standard Chernoff-type bounds [10],

$$\Pr\left[|C_2(\phi,\alpha)| < (1-\epsilon)\tfrac{9an}{2}\left(1-\tfrac{3a}{\Delta}\right)^2\right] \le \mathrm{e}^{-\frac{\epsilon^2 mp}{2}} \le \mathrm{e}^{-\frac{\epsilon^2 n}{2}\left[\frac{9a}{2}\left(1-\frac{3a}{\Delta}\right)^2\right]}$$

The second probability in the expression for $\Pr[C_2(\phi,\alpha) \in \mathrm{SAT}]$ is bounded using Lemma 2, since, conditioned on $|C_2(\phi,\alpha)|$ to have some known value z, clauses in $C_2(\phi,\alpha)$ are distributed according to $\mathcal{F}_z^{2,n-t}$. More precisely, since the probability of satisfying a formula decreases as the number of clauses in the formula increases, the sought probability is at most:

$$n^{O(1)}\Pr\left[C_2(\phi,\alpha) \in \mathrm{SAT} \mid |C_2(\phi,\alpha)| = (1-\epsilon)\tfrac{9a}{2}\left(\tfrac{\Delta-3a}{\Delta}\right)^2(n-t)\right]$$

Then, by Lemma 2, using the definition of t and ignoring the polynomial factor, this is at most

$$(\max_{(x,y)\in D_\gamma} f_d(x,y))^{\frac{(\Delta-3a)n}{\Delta}} \times \mathrm{e}^{\frac{3an}{\Delta}\ln 2}$$

where $d = (1-\epsilon)\frac{9a}{2}\left(1-\frac{3a}{\Delta}\right)^2$. Finally, putting everything together, the overall error probability is at most

$$\exp\left\{-n\left[\tfrac{9a\epsilon^2}{4}\left(1-\tfrac{3a}{\Delta}\right)^2 - \tfrac{3a\ln 2}{\Delta}\right]\right\} + n^{O(1)}(\max_{(x,y)\in D_\gamma} f_d(x,y))^{\frac{(\Delta-3a)n}{\Delta}} \cdot \mathrm{e}^{\frac{3an}{\Delta}\ln 2}$$

For $a = 1.1$ there exists an $\epsilon \in (0,1)$ such that the result holds if $\Delta > 20$.

4 Analysis for $k > 3$

For $k > 3$, let $\frac{t}{n} = ka\left(\frac{n}{m}\right)^{\frac{1}{k-2}}$. The probability that a fixed restriction defined on Y reduces a clause C to a 2-clause is

$$p = \frac{4\binom{n-t}{2}\binom{t}{k-2}}{2^k\binom{n}{k}} = 2^{2-k}\frac{(n-t)(n-t-1)}{2}\frac{t!}{(k-2)!\,(t-k+2)!}\frac{k!\,(n-k)!}{n!} = 2^{2-k}\binom{k}{2}(n-t)(n-t-1)\frac{t!}{(t-k+2)!}\frac{(n-k)!}{n!}$$

which can be rewritten as

$$p = 2^{2-k}\binom{k}{2}\frac{t}{n}\left(1-\tfrac{t-1}{n-1}\right)\left(1-\tfrac{t-1}{n-2}\right)\prod_{i=1}^{k-3}\tfrac{t-i}{n-i-2}$$

and this is at least

$$2\binom{k}{2}\left(1-\tfrac{t}{n}\right)^2\left(\tfrac{t}{4n}\right)^{k-2}$$

for sufficiently large n. Using once more the definition of t,

$$p > \tfrac{2}{\Delta}\binom{k}{2}\left(\tfrac{ka}{4}\right)^{k-2}\left(1-ka\Delta^{-\frac{1}{k-2}}\right)^2 =_{df} p_k$$

The proof can be completed exactly like in the case $k = 3$, by estimating the probability that $|C_2(\phi,\alpha)| < dn$ and the probability that $C_2(\phi,\alpha)$ be satisfiable conditioned on $|C_2(\phi,\alpha)| \ge dn$, for $d = (1-\epsilon)p_k\Delta$.

5 Final remarks and open questions

In this paper we presented a class of refutations which can be associated with high probability with any given unsatisfiable random k-CNF formula on n variables and $m = \Delta n$ clauses. A pebbling game can be played on the directed acyclic graphs corresponding to these refutations and relatively few pebbles are sufficient to win such game. As a consequence of this an upper bound can be obtained on the space complexity of refuting unsatisfiable random k-CNF formulae in resolution.

The analysis presented might be tightened by using refined bounds [3] on the probability that a random 2-CNF formula on n variables and $m > n$ clauses be satisfiable. Moreover, "more efficient" refutations might exist. However, Ben-Sasson and Galesi [2] recently proved an $\Omega\left(n\cdot\Delta^{-\frac{1+\epsilon}{k-2-\epsilon}}\right)$ lower bound (for any $\epsilon \in (0,1/2)$) which rules out the possibility of a significant improvement on the result presented.

Acknowledgements. The author wishes to thank Jacobo Torán for many helpful discussions, and Nicola Galesi for informing him of reference [2].

References

[1] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson. Space complexity in propositional calculus. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, Portland, OR, May 2000.

[2] E. Ben-Sasson and N. Galesi. Space complexity of random formulae in resolution. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*. IEEE Computer Society, June 2001.

[3] B. Bollobás, C. Borgs, J. T. Chayes, J. H. Kim, and D. B. Wilson. The scaling window of the 2-SAT transition. Technical Report MSR-TR-99-41, Microsoft, 1999. To appear in Random Structures and Algorithms.

[4] V. Chvátal. Almost all graphs with $1.44n$ edges are 3-colourable. *Random Structures and Algorithms*, 2:11–28, 1991.

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

[6] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.

[7] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the Association for Computing Machinery*, 7:201–215, 1960.

[8] J. Esteban, J. L. Toran. Space bounds for resolution. In C. Meinel and S. Tison, editors, *STACS 99: 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 1999. Proceedings*, volume 1563 of *Lecture Notes in Computer Science*, pages 551–560. Springer-Verlag, 1999.

[9] A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53:469–486, 1996.

[10] T. Hagerup and C. R¨ub. A guided tour of Chernoff bounds. *Information Processing Letters*, 33(6):305–308, February 1989-90.

[11] L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Structures and Algorithms*, 12(3):253–269, 1998.

[12] F. G. Maunsell. A problem in cartophily. *Mathematical Gazette*, 22(251):328–331, October 1938.

[13] J. Toran. Lower bounds for the space used in resolution. In J. Flum and M. Rodriguez-Artalejo, editors, *Computer Science Logic: 13th International Workshop, CSL'99, 8th Annual Conference of the EACSL, Madrid, Spain, September 20-25, 1999, Proceedings*, volume 1683 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

[14] M. Zito. *Randomised Techniques in Combinatorial Algorithmics.* PhD thesis, Department of Computer Science, University of Warwick, 1999.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'