On the Limits of Sparsification*

Rahul Santhanam[1] and Srikanth Srinivasan[2]

[1] University of Edinburgh rsanthan@inf.ed.ac.uk**
[2] DIMACS, Rutgers University srikanth@dimacs.rutgers.edu***

Abstract. Impagliazzo, Paturi and Zane (JCSS 2001) proved a sparsification lemma for k-CNFs: every k-CNF is a sub-exponential size disjunction of k-CNFs with a linear number of clauses. This lemma has subsequently played a key role in the study of the exact complexity of the satisfiability problem. A natural question is whether an analogous structural result holds for CNFs or even for broader non-uniform classes such as constant-depth circuits or Boolean formulae. We prove a very strong negative result in this connection: For every superlinear function $f(n)$, there are CNFs of size $f(n)$ which cannot be written as a disjunction of $2^{n-\varepsilon n}$ CNFs each having a linear number of clauses for any $\varepsilon > 0$. We also give a hierarchy of such non-sparsifiable CNFs: For every k, there is a k' for which there are CNFs of size $n^{k'}$ which cannot be written as a sub-exponential size disjunction of CNFs of size n^k. Furthermore, our lower bounds hold not just against CNFs but against an *arbitrary* family of functions as long as the cardinality of the family is appropriately bounded.

As by-products of our result, we make progress both on questions about circuit lower bounds for depth-3 circuits and satisfiability algorithms for constant-depth circuits. Improving on a result of Impagliazzo, Paturi and Zane, for any $f(n) = \omega(n \log(n))$, we define a pseudo-random function generator with seed length $f(n)$ such that with high probability, a function in the output of this generator does not have depth-3 circuits of size $2^{n-o(n)}$ with bounded bottom fan-in. We show that if we could decrease the seed length of our generator below n, we would get an explicit function which does not have linear-size logarithmic-depth series-parallel circuits, solving a long-standing open question.

Motivated by the question of whether CNFs sparsify into bounded-depth circuits, we show a *simplification* result for bounded-depth circuits: any bounded-depth circuit of linear size can be written as a sub-exponential size disjunction of linear-size constant-width CNFs. As a corollary, we show that if there is an algorithm for CNF satisfiability which runs in time $O(2^{\alpha n})$ for some fixed $\alpha < 1$ on CNFs of linear size, then there is an algorithm for satisfiability of linear-size constant-depth circuits which runs in time $O(2^{(\alpha+o(1))n})$.

* This is an extended abstract with some proofs missing. The full version may be found at [11].

** Partially supported by ESPRC Grant EP/H05068X/1

*** Work partially done as a Member at the Institute of Advanced Study, Princeton.

1 Introduction

The Satisfiability (SAT) problem is of central importance in theoretical computer science. Since SAT is NP-complete, the NP vs P problem reduces to the question of whether SAT has polynomial-time algorithms. We do not believe that SAT has polynomial-time algorithms, however it is still a very interesting question which the best algorithms are for solving SAT in the worst case. Specifically, by how much can we improve over the "naive" brute-force search algorithm for SAT, which enumerates over all possible 2^n assignments for a SAT instance and checks whether any of them are satisfying? A very concrete motivation for this problem is that SAT instances need to be solved in the real world, in a variety of contexts such as verification, automated planning and testing [6].

From a complexity-theoretic point of view, the importance of improving over brute-force search has been illustrated by the recent results of Williams [14] [15]. He shows that even *marginal* improvements over brute-force search for satisfiability of Boolean circuits in a class $\mathcal{C}$ implies that NEXP does not have polynomial-size circuits in the class $\mathcal{C}$, for a range of natural classes $\mathcal{C}$ of circuits. He applies his methodology [15] to obtain a new circuit lower bound, namely that NEXP $\not\subseteq \mathsf{ACC}^0$, by designing an algorithm performing slightly better than brute-force search for ACC^0-SAT. In fact, there are connections between SAT algorithms and lower bounds in the opposite direction as well, as evidenced in recent work using lower bound techniques to design and analyze improved Satisfiability algorithms [10] [3]. This makes the question of understanding the complexity landscape of the SAT problem even more intriguing.

When trying to design an improved algorithm, a natural approach is to find general structural properties of the class of instances which can be exploited algorithmically. Some examples of such properties for SAT are the downward self-reducibility property used to reduce the search problem to the decision version, and the Satisfiability Coding Lemma of Paturi, Pudlak and Zane, which has been used to design and analyze better algorithms for k-SAT as well as to prove depth-3 circuit lower bounds for restricted classes of circuits [9] [8].

Perhaps the most influential such property is that of *sparsifiability.* The Sparsification Lemma of Impagliazzo, Paturi and Zane [5] plays a key role in the study of the exact complexity of SAT. It states that for any constants $\epsilon > 0$ and k a positive integer, any k-CNF on n variables can be written as the disjunction of $2^{\epsilon n}$ *linear-size* CNFs, where the constant factor in the size depends only on k and ϵ.

The Lemma has found many different applications in both algorithmic and lower bound contexts. Impagliazzo, Paturi and Zane [5] used a constructive version of it in their study of sub-exponential reducibilities between NP-complete problems. Their results indicate that the Exponential-Time Hypothesis (ETH), which states that 3-SAT is not solvable in time $2^{o(n)}$, can be used as a unifying hypothesis in the study of exact complexity of NP-hard problems. They prove that, for various problems such as k-SAT (where $k \geq 3$ is a positive integer), k-Colourability, Clique, Vertex Cover, Satisfiability of linear-size Boolean circuits etc., existence of a $2^{o(n)}$ time algorithm is equivalent to ETH. The Lemma has also been used to undertake more refined studies of the complexity of SAT in terms of various parameters such as clause width and clause density [4] [2]. From the point of view of lower bounds, the Lemma has been used to construct a small pseudorandom family of functions such that with high probability, a function in this family does not have depth-3 circuits of size $2^{n-o(n)}$ and bounded bottom fan-in. This is closely related to classical questions about lower bounds for linear-size logarithmic-depth circuits [13].

It is natural to ask whether a similar sparsifiability property holds for broader classes of formulae or circuits, such as CNFs or even constant-depth circuits. Such a result would be useful in getting better algorithmic results and deriving new lower bounds. For example, while k-SAT is solvable in time $2^{n-\Omega(n)}$ for $m = poly(n)$ and constant k, the best known algorithm for SAT on general CNFs runs in time $2^{n-\Omega(n/\log(m/n))}$. A sparsification lemma for CNFs would be an important step towards a $2^{n-\Omega(n)}$ time algorithm for SAT on polynomial-size formulae. Indeed, this has explicitly been posed as an open question by Calabro, Impagliazzo and Paturi [2].

In this paper, we show a strong *negative* answer to the question of whether CNFs (and hence also more general classes of circuits) can be sparsified.

Theorem 1. *Let $f : \mathbb{N} \to \mathbb{N}$ be any function such that $f(n) = \omega(n)$. Then there is a sequence of CNFs $\{\phi_n\}$, where for each n ϕ_n has n variables and has size at most $f(n)$, such that for any constants $\varepsilon \in (0, 1]$ and $c > 0$, for all large enough n ϕ_n cannot be written as the OR of $2^{n-\varepsilon n}$ CNFs of size at most cn. In particular, CNFs are not sparsifiable.*

In fact, what we show is significantly stronger - for any sequence $\{F_n\}$ of families of Boolean functions such that $|F_n| = n^{O(n)}$, there is a sequence of CNFs which are not expressible as a $2^{n-\Omega(n)}$ size disjunction of functions in F_n. Also, the CNFs for which we show this are very natural. The functions they represent are the solution sets of sparse linear equations.

Theorem 1 only rules out "sparsifying" superlinear-size CNFs to linear-size CNFs. It could potentially still be the case that n^3-size CNFs are sparsifiable into n^2-size CNFs. It turns out that the counter-examples of Theorem 1 cannot establish this stronger statement, however by using a different set of counter-examples and a similar argument, we derive a hierarchy of non-sparsifiable CNFs.

Theorem 2. *Let k and $k' > 2k$ be any fixed constants. There is a fixed $\epsilon > 0$ and a sequence of CNFs $\{\phi_n\}$ where ϕ_n has n variables and $|\phi_n| \leq n^{k'}$ such that for large enough n, ϕ_n cannot be written as the OR of $2^{\epsilon n}$ CNFs of size at most n^k.*

The hard CNFs are again natural - they are simply *random* CNFs of a specified width and size. Thus, in a sense, the proof of Theorem 2 shows that CNFs cannot be sparsified even *on average*.

We motivated the question about sparsification by describing the possible applications of a positive result. It turns out that our negative results have a couple of interesting byproducts as well. By itself, the results give some indication of the obstacles to designing better SAT algorithms, as well as what kinds of instances are likely to be hard. For example it is known that in certain contexts, such as for Resolution-based algorithms, instances encoding subspaces or random instances are hard. Our results are in a similar spirit.

More concretely, motivated by Theorem 1, we construct a simple new sub-exponential time reduction from satisfiability on linear-size constant-depth circuits to k-SAT. The motivation is to apply Theorem 1 to show that CNFs cannot in general be sparsified into linear-size constant depth circuits. We cannot simply use the stronger form of Theorem 1 for arbitrary families of functions of small enough cardinality here, as we are unable to bound the number of functions computed by unbounded fan-in linear-size constant-depth circuits by $n^{O(n)}$. Instead, we show a *positive* result that any linear-size constant-depth circuit can be written as an OR of $2^{\epsilon n}$ k-CNFs for any $\epsilon > 0$ and k depending only on ϵ. This decomposition can actually be done constructively, and this gives us the reduction we mentioned before. The decomposition also implies that superlinear-size CNFs cannot be sparsified into linear-size constant-depth circuits.

Theorem 3. *Let $\{f_n\}$ be a sequence of Boolean functions on n bits, such that f_n is computed by linear-size constant-depth circuits. For any constant $\epsilon > 0$, there is a constant k such that f_n is the disjunction of $2^{\epsilon n}$ functions each of which is computed by a k-CNF of linear size.*

Theorem 1 also has an application to circuit lower bounds. Here we are concerned with lower bounds for depth-3 circuits where there is a bound on the bottom fan-in. If we could show that there is an explicit function which does not have size $2^{n/2}$ depth-3 circuits with bottom fan-in $O(1)$, this would be a lower bound breakthrough, as using a connection due to Valiant[13] it would imply a superlinear-size lower bound against logarithmic-depth series-parallel circuits. Valiant argues that the series-parallel restriction on the structure of the circuit is interesting because the best-known circuits for many problems are series-parallel. Impagliazzo, Paturi and Zane [5] make progress on this question by constructing an explicit pseudo-random family of $2^{O(n^2)}$ functions such that most functions in the family do not have size $2^{n-\Omega(n)}$ depth-3 circuits with bottom fan-in $O(1)$. We improve their result by reducing the size of the function family down to $n^{f(n)}$ for any $f(n) = \omega(n)$. We also argue that a further improvement of the family size to 2^{cn} for $c < 1$ would actually imply a breakthrough lower bound for an explicit function.

In the theorem below, a Σ_3 circuit is an unbounded fan-in depth 3 circuit where the top gate is an OR. Note that when trying to prove a lower bound for an explicit function, we can assume wlog that the top gate is an OR.

Theorem 4. *For each $f(n) = \omega(n)$, there is a sequence $\{F_n\}$ of families of Boolean functions on n bits, where F_n has size at most $n^{f(n)}$, such that with probability $1 - o(1)$, a random function from F_n does not have Σ_3^k circuits of size $2^{n-\Omega(n)}$ with bottom fan-in $O(1)$. Moreover, given $i \in [1, n^{f(n)}]$ in binary and $x \in \{0,1\}^n$, there is a polynomial-time algorithm for evaluating the i'th function in F_n on x.*

2 Preliminaries

2.1 Basic complexity notions

We assume a basic knowledge of complexity theory. Standard references for this include the book by Arora and Barak [1] and the Complexity Zoo[3].

When discussing sparsification, we find it convenient to talk of non-uniform complexity measures. A non-uniform complexity measure $\mathcal{CSIZE}$ associates with each integer n and size bound s, a class of Boolean functions $\mathcal{CSIZE}(s(n))$ on n bits, such that for any $s' \geq s, \mathcal{CSIZE}(s(n))) \subseteq \mathcal{CSIZE}(s'(n))$. We will be concerned mainly with measures which correspond directly to standard models of computation, such as CNFs, CNFs of constant width (referred to as $O(1)$-CNFs), constant-depth unbounded fan-in circuits (AC^0), Boolean formulae and Boolean circuits.

By the size of a CNF, we will typically mean the number of clauses. If we mean the total number of literal occurrences, we will make this explicit.

As we will be studying lower bounds for depth-3 circuits, we require some notation for such circuits. Define Σ_d^k to be the set of depth d circuits with top gate OR such that each bottom gate has fan-in at most k. It is known that any Σ_3^k circuit for the Parity function or the Majority function requires $\Omega(2^{n/k})$ gates, and such bounds are tight for $k = O(\sqrt{n})$. For $k = 2$, a $2^{n-o(n)}$ size lower bound is known for an explicit function in P, however not even an $\Omega(2^{n/2})$ size lower bound is known for an explicit function for any $k > 2$. Using a connection due to Valiant [13], this question can be related to classical lower bound questions about linear-size logarithmic-depth Boolean circuits. Valiant's results imply that linear-size logarithmic-depth Boolean circuits with bounded fan-in can be computed by depth-3 unbounded fan-in circuits of size $O(2^{n/\log \log n})$ with bottom fan-in limited by n^{ε} for arbitrarily small ε. If in addition, the graph of connections of the circuit is restricted to be series-parallel, the simulation can be modified to give size $2^{n/2}$ and fan-in $O(1)$.

Given functions $f, g : \mathbb{N} \rightarrow \mathbb{R}^{>0}$, we occasionally use $f \ll g$ to denote $f(n) = o(g(n))$. This notation makes the transitivity of the $o(\cdot)$ relation more transparent.

2.2 Sparsification and simplification

Definition 1. *Given non-uniform complexity measures $\mathcal{CSIZE}$ and $\mathcal{C}'SIZE$, and functions $s, s' : \mathbb{N} \rightarrow \mathbb{N}$, we say that there is a $(\mathcal{C}, s, \mathcal{C}', s')$-sparsification if for any constant $\epsilon > 0$ and any function $f \in \mathcal{CSIZE}(O(s))$, f is the OR of at most $2^{\epsilon n}$ functions each belonging to $\mathcal{CSIZE}(O(s'))$. We say that $\mathcal{C}$ is sparsifiable to $\mathcal{C}'$ if there is a $(\mathcal{C}, n^k, \mathcal{C}', n)$-sparsification for each k, and we say simply that $\mathcal{C}$ is sparsifiable if $\mathcal{C}$ is sparsifiable to $\mathcal{C}$.*

Definition 2. *Given non-uniform complexity measures $\mathcal{CSIZE}$ and $\mathcal{C}'SIZE$, and function $s : \mathbb{N} \rightarrow \mathbb{N}$, we say that there is an OR-simplification of $\mathcal{C}$ to $\mathcal{C}'$*

[3] `http://qwiki.stanford.edu/index.php/Complexity_Zoo`

at size s if there is a $(\mathcal{C}, s, \mathcal{C}', s)$-sparsification. We say that there is an OR-simplification of $\mathcal{C}$ to $\mathcal{C}'$ if there is an OR-simplification of $\mathcal{C}$ to $\mathcal{C}'$ at size n.

The following proposition is immediate since sub-exponential size ORs are closed under composition.

Proposition 1. *If $\mathcal{C}$ is sparsifiable to $\mathcal{C}'$ and there is an OR-simplification of $\mathcal{C}'$ to $\mathcal{C}$, then $\mathcal{C}$ is sparsifiable.*

There are many interesting positive results on sparsification and simplification. Impagliazzo, Paturi and Zane [5] showed that k-CNFs are sparsifiable for any constant k. Improved parameters were obtained by [2].

Lemma 1 (Sparsification Lemma). *[5] [2] Let $k > 0$ be any integer. For any constant $\epsilon > 0$, there exists a constant $c(k, \epsilon)$ such that for large enough n, any k-CNF over n variables can be expressed as the OR of $2^{\epsilon n}$ k-CNFs each of size at most $c(k, \epsilon)n$.*

The original proof of Lemma 1 [5] yielded c doubly exponential in k but this was subsequently improved to singly exponential in k. Using results of Miltersen, Radhakrishnan and Wegener [7], it can be shown that an exponential dependence on k is necessary.

Schuler [12] showed that there is an OR-simplification of CNFs to $O(1)$-CNFs. This follows from the following more general lemma, the proof of which is similar and is deferred to the full version.

Lemma 2. *For any constant $\varepsilon \in (0, 1]$ and function $c : \mathbb{N} \to \mathbb{N}$, every CNF φ with at most cn clauses can be written as the OR of at most $2^{\varepsilon n}$ many k-CNFs with at most cn clauses, where $k = O(\frac{1}{\varepsilon} \log(\frac{c}{\varepsilon}))$.*

Note that when c is a constant in Lemma 2, k is a constant as well.

Corollary 1. *There is an OR-simplification of CNFs to $O(1)$-CNFs.*

3 The Limits of sparsification

3.1 Non-sparsifiability of CNFs

We will show that there are CNFs of slightly superlinear size that cannot be written as a subexponential OR of CNFs of linear size.

Given $\ell, r \in \mathbb{N}$, let $\mathcal{S}_{\ell,r}$ denote the collection of all r-tuples of subsets of $[n]$ of size ℓ. Given $\overline{S} = (S_1, \ldots, S_r) \in \mathcal{S}_{\ell,r}$, let $\varphi_{\overline{S}}$ denote some CNF for the following function:

$$G_{\overline{S}} = \bigwedge_{i=1}^{r} \neg \bigoplus_{j \in S_i} x_j$$

Though the above function has not been written in CNF form, it is easy to see that for any $\overline{S}$ as above, $\varphi_{\overline{S}}$ can be chosen to be CNFs of size at most $r2^{\ell}$.

Lemma 3. *Fix any $\ell, r : \mathbb{N} \to \mathbb{N}$. Then we have that for any $\overline{S} \in \mathcal{S}_{\ell,r}$, the CNF $\varphi_{\overline{S}}$ has at least 2^{n-r} satisfying assignments.*

Proof. This follows from the fact that any homogeneous system of r linear equations has at least 2^{n-r} solutions over $\mathbb{F}_2$. $\square$

Now we proceed to the proof of the main lemma. Given a CNF formula φ, let $\mathrm{Sat}(\varphi)$ denote the set of satisfying assignments of φ.

Fix a $T \subseteq [n]$ and assume that $S \in \binom{[n]}{\ell}$ is chosen uniformly at random. Given $\eta \in [0,1]$, we call S $(1-\eta)$-*balanced w.r.t.* T if $|S \cap T| \geq (1-\eta)\,\mathbf{E}_S[|S \cap T|]$. We call S *balanced w.r.t.* T if S is 1/2-balanced w.r.t. T. Given $\overline{S} \in \mathcal{S}_{\ell,r}$, we say that $\overline{S}$ is $(1-\eta)$-*balanced w.r.t.* T (*balanced w.r.t.* T) if at least half the S_i are $(1-\eta)$-balanced w.r.t. T (respectively, balanced w.r.t. T).

We need the following technical lemma regarding balance.

Lemma 4. *Let $\varepsilon, \eta \in (0,1)$ be constants. Fix $\ell = \ell(n), r = r(n)$ such that $1 \ll \ell(n)$ and $n/\ell \ll r(n)$. Assume $T \subseteq [n]$ such that $|T| \geq \varepsilon n$. Then for a randomly chosen $\overline{S} \in \mathcal{S}_{\ell,r}$, we have $\Pr_{\overline{S}}[\overline{S}$ is not $(1-\eta)$-balanced w.r.t. $T] = \frac{1}{2^{\omega(n)}}$.*

Proof. A simple concentration equality tells us that for any $i \in [r]$, $\Pr_{S_i}[S_i \text{ not } (1-\eta)\text{-balanced}] \leq 2^{-\Omega(\ell)}$. Hence, given a set of $r/2$ many S_i, the probability that *none* of them are balanced w.r.t. T is bounded by $2^{-\Omega(\ell r)} = 2^{-\omega(n+r)}$, where the last equality follows from the fact that $r \gg n/\ell$. By a union bound, it follows that the probability that there *exists* a subset of $\overline{S}$ of size $r/2$ all of whose elements are not $(1-\eta)$-balanced w.r.t T is at most $\binom{r}{r/2}2^{-\omega(n)} \leq 2^r 2^{-\omega(n+r)} \leq 2^{-\omega(n)}$. The lemma now follows since this event corresponds precisely to $\overline{S}$ not being balanced w.r.t T. $\square$

Lemma 5. *Fix constants $c, \varepsilon > 0$. Let $\ell = \ell(n), r = r(n)$ be parameters such that $1 \ll \ell = O(\log n)$, $n/\ell \ll r \ll n$. Fix any collection $\mathcal{A}$ of subsets of $\{0,1\}^n$ of size at most n^{cn} such that each $A \in \mathcal{A}$ has size at least $2^{\varepsilon n}$. Then, for a random $\overline{S} \in \mathcal{S}_{\ell,r}$, we have $\Pr_{\overline{S}}[\exists A \in \mathcal{A} : \ A \subseteq \mathrm{Sat}(\varphi_{\overline{S}})] = o(1)$.*

Proof. Fix any $A \in \mathcal{A}$. Since $\mathrm{Sat}(\varphi)$ is a subspace of $\mathbb{F}_2^n$, we see that $A \subseteq \mathrm{Sat}(\varphi)$ iff $\mathrm{Span}(A) \subseteq \mathrm{Sat}(\varphi)$, where $\mathrm{Span}(A)$ is the span of A in $\mathbb{F}_2^n$. Hence, we assume wlog that every $A \in \mathcal{A}$ is actually a subspace of dimension at least εn. Fix such a subspace A. Let $d \geq \varepsilon n$ denote the dimension of A.

By Gaussian elimination, we can choose a $d \times n$ matrix $M(A)$ such that the rows of $M(A)$ generate A and after some column permutations, $M(A) = [I_d\ M']$ where I_d denotes the $d \times d$ identity matrix. Let the variables indexed by the first d columns of $M(A)$ be denoted $S(A)$.

Consider a uniformly random $\overline{S} = (S_1, \ldots, S_r) \in \mathcal{S}_{\ell,r}$. For $i \in [r]$ let χ_i denote the characteristic vector of S_i. It is easily seen that $A \subseteq \mathrm{Sat}(\varphi_{\overline{S}})$ iff each $\chi_i \in A^\perp$, where $A^\perp$ denotes the dual space of A.

We now consider the probability that $\chi_i \in A^\perp$ for any fixed i. This happens iff $M(A)\chi_i = 0$. Note that this event can occur with probability at least $\frac{1}{2^{O(\ell)}}$ if, for example, $M' = 0$ and it happens that $S_i \subseteq [n] \setminus S(A)$. We now show that this probability is much lower if we condition on the event that S_i is balanced w.r.t. $S(A)$.

Say we condition on $|S_i \cap S(A)| = q$, where $q \in [\ell]$. Note that picking a random S_i conditioned on this event is equivalent to picking a random subset

S_i' of $S(A)$ of size q and a random subset S_i'' of $\overline{S(A)}$ of size $\ell - q$ and setting $S_i = S_i' \cup S_i''$. Let χ_i' and χ_i'' denote the characteristic vectors of S_i' and S_i'' respectively. Then, $M(A)\chi_i = 0$ iff $I_d\chi_i' + M'\chi_i'' = 0$ iff $\chi_i' = M'\chi_i''$. For any fixed choice of χ_i'', the probability over the choice of χ_i' that this occurs is at most $1/\binom{d}{q} \le (q/\varepsilon n)^q \le \frac{1}{(\varepsilon n)^{\Omega(q)}}$. Hence, conditioned on S_i being balanced w.r.t. $S(A)$, we see that the probability that $M(A)\chi_i = 0$ is at most $\frac{1}{(\varepsilon n)^{\Omega(\varepsilon\ell)}} \le \frac{1}{n^{\Omega(\ell)}}$. Using the fact that $r = \omega(n/\ell)$, this implies that $\Pr_{\overline{S}}[\forall i \in [r] : \ M(A)\chi_i = 0 \mid \overline{S} \text{ balanced w.r.t. } S(A)] \le \left(\frac{1}{n^{\Omega(\ell)}}\right)^{r/2} = \frac{1}{n^{\omega(n)}}$. $(*)$

We are now ready to bound the probability that there exists a subspace $A \in \mathcal{A}$ that is contained in $\mathrm{Sat}(\varphi_{\overline{S}})$. Let $E_1(A)$ denote the event that $A \subseteq \mathrm{Sat}(\varphi_{\overline{S}})$. Given $T \subseteq [n]$ s.t. $|T| \ge \varepsilon n$, let $E_2(T)$ denote the event that $\overline{S}$ is not balanced w.r.t. T. We have

$$\begin{aligned}
\Pr_{\overline{S}}[\bigvee_A E_1(A)] &\le \Pr_{\overline{S}}[\bigvee_A E_1(A) \vee \bigvee_{T\subseteq[n]:|T|\ge\varepsilon n} E_2(T)] \\
&= \Pr_{\overline{S}}[\bigvee_T E_2(T)] + \Pr_{\overline{S}}[\bigvee_A E_1(A) \wedge \neg \bigvee_T E_2(T)] \\
&\le \sum_T \Pr_{\overline{S}}[E_2(T)] + \sum_A \Pr_{\overline{S}}[E_1(A) \wedge \neg E_2(S(A))] \\
&\le \sum_T \Pr_{\overline{S}}[E_2(T)] + \sum_A \Pr_{\overline{S}}[E_1(A) \mid \neg E_2(S(A))] \\
&\le 2^n \cdot \frac{1}{2^{\omega(n)}} + n^{cn} \cdot \frac{1}{n^{\omega(n)}} = o(1)
\end{aligned}$$

where the last inequality follows from Lemma 4 and $(*)$. This concludes the proof of the lemma. □

Theorem 5. *Fix any constants $c > 0$ and $\varepsilon \in (0,1]$. Say $\overline{S}$ is chosen uniformly at random from $\mathcal{S}_{\ell,r}$, where ℓ, r are as in the statement of Lemma 5. Then, the probability that $\varphi_{\overline{S}}$ can be written as a union of at most $2^{n-\varepsilon n}$ many CNFs of size at most cn is $o(1)$.*

Proof. Assume that for some $\overline{S}$, $\varphi_{\overline{S}}$ can be written as an OR of at most $2^{n-\varepsilon n}$ many CNFs of size at most cn. By Lemma 2, each such CNF can be written as a union of at most $2^{\varepsilon n/2}$ many k-CNFs of size at most cn, where $k = k(c,\varepsilon)$ is a constant. Moreover, Lemma 3 implies that $|\mathrm{Sat}(\varphi_{\overline{S}})| \ge 2^{n-r} = 2^{n-o(n)}$. Hence, it must be the case that there is some k-CNF ψ of size at most cn such that $|\mathrm{Sat}(\psi)| \ge 2^{\varepsilon n/4}$ and $\mathrm{Sat}(\psi) \subseteq \mathrm{Sat}(\varphi_{\overline{S}})$. Let $\mathcal{A} = \{\mathrm{Sat}(\psi) \mid \psi \text{ a } k\text{-CNF}, \mathrm{Size}(\psi) \le cn, \text{ and } |\mathrm{Sat}(\psi)| \ge 2^{\varepsilon n/4}\}$; clearly, $|\mathcal{A}| \le \binom{(2n)^k}{cn} \le n^{kcn}$. We have seen above that if $\varphi_{\overline{S}}$ can be written as an OR of at most $2^{n-\varepsilon n}$ many CNFs of size at most cn, then there must be an $A \in \mathcal{A}$ such that $A \subseteq \mathrm{Sat}(\varphi_{\overline{S}})$. By Lemma 5, the probability that this happens is $o(1)$. Hence, the theorem follows. □

The above easily yields Theorem 1 by choosing $\ell = \omega(1)$ small enough and $r = n/\sqrt{\ell}$ so that $f(n) \ge n2^\ell/\sqrt{\ell}$, and then using Theorem 5 to yield existence of CNFs of the desired size which are non-sparsifiable.

3.2 A Hierarchy Theorem for Non-Sparsifiability

Theorem 5 shows the existence of CNFs of slightly super-linear size which cannot be sparsified into linear-size CNFs. A natural question is whether there is a hierarchy of such non-sparsifiable CNFs: is it true that for each k, there is an $k' > k$ such that there are CNFs of size $n^{k'}$ which cannot be sparsified into CNFs of size n^k.

First note that the hard CNFs we're looking for cannot be of the form $\varphi_{\overline{S}}$ for some $\overline{S} \in \mathcal{S}_{\ell,r}$. This is because the corresponding function $G_{\overline{S}}$ trivially has formulae of size $o(n\log(n))$ over the basis $\{\wedge, \vee, \oplus\}$, and so also is sparsifiable into formulae of the same size over this basis. Lemma 5 shows non-sparsifiability into *any* class of functions of small enough cardinality, so we cannot hope to strengthen Lemma 5 to get the desired result for $k > 1$.

Instead, we use a random CNF ψ with a prescribed width and clause density. Fix $n \in \mathbb{N}$ and $\ell : \mathbb{N} \to \mathbb{N}$. We denote by $\Psi_{n,\ell(n)}$ the collection of all CNF formulas on n boolean variables of width exactly $\ell(n)$ with $2^{\ell(n)}$ many clauses (with possible repetitions). To sample a random ψ from $\Psi_{n,\ell(n)}$, we simply sample $2^{\ell(n)}$ random clauses of width $\ell(n)$. We establish the following theorem, whose proof is omitted in this version.

Theorem 6. *Fix constants $c \geq 1, \eta > 0$. Assume $\ell = \ell(n) = (2c + \eta)\log n$. Then, then there exists a fixed $\delta = \delta(\eta, c) > 0$ such that the probability that a random ψ sampled from $\Psi_{n,\ell}$ can be written as an OR of at most $2^{\delta n}$ many CNFs of size at most $O(n^c)$ is at most $3/4 + o(1)$. In particular, there is no (CNF, $n^{2c+\eta}$, CNF, n^c)-sparsification.*

Theorem 6 straightaway implies Theorem 2.

4 Simplifying AC^0 to CNFs

In this section, all AC^0 circuits considered will have AND gates as their output gates. Note that any AC^0 circuit can be converted to this form by adding an additional AND gate at the output, hence increasing the size and depth by 1.

Definition 3. *Given $s, d, k \in \mathbb{N}$, an AC^0 circuit C with an AND gate as its output gate is said to be* (s, d, k)-bounded *if it has size at most s, depth at most d, and all of its gates except the output gate have fanin bounded by k.*

Fact 7 *For constants $d, k \in \mathbb{N}$ and any $s \in \mathbb{N}$, any (s, d, k)-bounded AC^0 circuit can be written as a CNF of size $O(s)$ and width k^d.*

Definition 4. *Given $N, s, k \in \mathbb{N}$, a set $\mathcal{C}$ of at most N (s, d, k)-bounded AC^0 circuits is said to be an* (N, s, d, k)-disjoint system *if the set of satisfying assignments of each pair of distinct circuits $C_1 \neq C_2$ from $\mathcal{C}$ are disjoint. The function computed by $\mathcal{C}$ is defined to be $\bigvee_{C\in\mathcal{C}} C$.*

Lemma 6. *Fix constants $c, d \in \mathbb{N}$ such that $d \geq 2$ and $\varepsilon \in (0, 1]$. There exists a $k = k(c, d, \varepsilon)$ and a $c' = c'(c, d, \varepsilon)$ such that for any AC^0 circuit C of depth d and size at most cn on n variables, there is an $(2^{\varepsilon n}, c'n, d, k)$-disjoint system $\mathcal{C}$ that computes the same function as C.*

Proof. The proof is by induction on d. We need a small variant of Lemma 2, which gives us the base case of $d = 2$:

Claim. For any $c \in \mathbb{N}$ and $\varepsilon \in (0, 1]$, there exists a $k = k(c, \varepsilon) \in \mathbb{N}$ such that for any collection $\mathcal{S}$ of at most cn many clauses (respectively, terms), there is a partition of $\{0, 1\}^n$ into at most $2^{\varepsilon n}$ many parts such that in each part, each clause (resp. term) in $\mathcal{S}$ has size at most k. Moreover, each element of the partition is specified by a k-CNF with at most $(c + 1)n$ clauses.

Proof. We prove the result in the case of clauses; the proof for terms is almost identical. Let k be a parameter that we will choose later. As long as there is a clause of width at least k, choose k literals from the clause and split the remainder of the space into two parts depending on whether the disjunction of these literals is satisfied or not. Call the branch where the literals are *not* satisfied the *good* branch. Along the good branch, we can set k variables to some boolean values; along the other branch, we still end up satisfying the clause.

Note that there can be only $cn + n/k$ many steps overall, since every step either satisfies a clause or sets k variables. Moreover, there can be at most n/k many good steps along any branch. This means that the total number of branches is bounded by $\binom{cn+n/k}{n/k} \leq \binom{(c+1)n}{n/k} \leq (ek(c + 1))^{n/k} \leq 2^{O(\log(kc)n/k)} \leq 2^{\varepsilon n}$ for large enough k depending on c and ε.

Note, moreover, that inputs corresponding to each branch is given by a k-CNF, where k with at most $cn + n/k \cdot k = (c + 1)n$ many clauses. □

The above claim easily implies that for any CNF φ with at most cn clauses, there is a $(2^{\varepsilon n}, (2c + 1)n, 2, k)$-disjoint system computing the same function as φ, where k is as defined in Claim 4.

Now consider a circuit of depth $d > 2$. Let $C_{<d}$ be the circuit C up to layer $d - 1$, with the layer of height 1 gates being replaced by a new set of variables $y_1, \ldots, y_m$, where $m \leq cn$. By applying the induction hypothesis to $C_{<d}$ with $\varepsilon = \varepsilon/(2c)$, we see that there exist $c_1, k_1 \in \mathbb{N}$ and a $(2^{\varepsilon n/2}, c_1 n, d - 1, k_1)$-disjoint system $\mathcal{C}$ that computes the same function as $C_{<d}$ on inputs coming from $\{0, 1\}^m$.

Moreover, by applying Claim 4 to the AND and OR gates at height 1, there exists $k_2 \in \mathbb{N}$ and a partition $\mathcal{P}$ of $\{0, 1\}^n$ into at most $2^{\varepsilon n/2}$ parts, each of which is specified by a k_2-CNF of size at most $(c+1)n$, such that in each partition, each gate at height 1 depends on at most k_2 variables. For each $P \in \mathcal{P}$, let φ_P denote the k_2-CNF of size at most $(c + 1)n$ that accepts exactly the inputs in P; given any circuit $C' \in \mathcal{C}$, let C_P denote the circuit $C'' \wedge \varphi_P$, where C'' is obtained by substituting for each y_i the corresponding term or clause of width at most k_2 that agrees with the corresponding gate on inputs from the set P of inputs. The set of all such circuits C_P gives us a $(2^{\varepsilon n}, (c_1 + c + 1)n, d, \max\{k_1, k_2\})$-disjoint system that computes the same function as the circuit C. □

Corollary 2. *There is an OR-simplification of* AC^0 *to* $O(1)$*-CNFs. In particular, we have:*

1. *For any function* $f(n) = \omega(n)$ *and constants* $c, \varepsilon > 0$*, there is a sequence of CNFs* $\{\varphi_n\}$*, where* φ_n *has* n *variables and size at most* $f(n)$ *such that* φ_n

cannot be written as an OR of at most $2^{n-\varepsilon n}$ *many* AC^0 *circuits of depth* d *and size at most* cn*.*

2. *If satisfiability of linear-size CNFs can be tested in time* $2^{\alpha n}$ *for some fixed* $\alpha < 1$*, then satisfiability of linear-size* AC^0 *circuits can also be tested in time* $2^{(\alpha+\varepsilon)n}$*, for any fixed* $\varepsilon > 0$*.*

Proof. That there is an OR-simplification of AC^0 to $O(1)$-CNFs follows directly from Lemma 6 and Fact 7. Item 1 then follows from Theorem 1. Item 2 follows trivially. □

Theorem 3 follows from Corollary 2.

5 Circuit lower bounds for depth-3 circuits

Impagliazzo, Paturi and Zane [5] showed that non-sparsifiability is closely connected to lower bounds for depth-3 circuits with bounded bottom fan-in. It is a long-standing open problem to find an explicit Boolean function which requires Σ_3^k circuits of size $2^{\omega(n/k)}$, where k is the bottom fan-in.

It is implicit in [5] that there is no $(\mathrm{AC}^0[\oplus], n^2, \mathcal{C}, n)$-sparsification for any complexity measure $\mathcal{CSIZE}$ such that there are at most $n^{O(n)}$ Boolean functions in $\mathcal{CSIZE}(O(n))$. They use this to construct an explicit family of $2^{O(n^2)}$ Boolean functions such that with probability close to 1, a random function from this family does not have Σ_3^k circuits of size $2^{n-o(n)}$ for $k = o(\log\log(n))$. Note that such a lower bound holds for a *purely random* Boolean function using a straightforward counting argument; what their result gives is a pseudo-random function family of significantly smaller size for which the lower bound still holds with high probability. Their result relies on the sparsification lemma first proved in the same paper. Using our result, we can prove Theorem 4, which reduces the size of the family down to $n^{f(n)}$ for any $f(n) = \omega(n)$, which, as we show, is "close" to getting the lower bound for an explicit function.

Proof (of Theorem 4). The function family $\{F_n\}$ we use is simply the set $\{G_{\overline{S}}\}$, where $\overline{S} \in \mathcal{S}_{\ell,r}$, with ℓ and r chosen as in the proof of Theorem 1. The bound on the cardinality of F_n and the polynomial-time evaluability of functions in F_n are clear. We will show that if a function f cannot be written as an OR of $2^{n-\epsilon n}$ CNFs of linear size for any $\epsilon > 0$, then it does not have Σ_3^k circuits of size $2^{n-o(n)}$ with bottom fan-in $O(1)$. Thus the theorem follows using Theorem 5.

Suppose, on the contrary, that there is a constant $c < 1$ such that f has Σ_3^k circuits of size 2^{cn} with bottom fan-in $k = O(1)$. Consider the gates with output wires feeding in to the top OR gate. Each such gate computes an $O(1)$-CNF. By Lemma 1, for any $\epsilon > 0$ each such gate can be written as the OR of $2^{\epsilon n}$ $O(1)$-CNFs of size $O(n)$. By choosing ϵ such that $\epsilon + c < 1$, we get that f is the OR of $2^{c'n}$ functions, each of which has CNFs of size $O(n)$ for some $c' < 1$. This contradicts the assumption on f, hence we are done. □

Theorem 8. *Suppose there is a sequence* $\{F_n\}$ *of families of Boolean functions on* n *bits, where* F_n *has size at most* $2^{n-\Omega(n)}$*, such that for large enough* n*, there* *exists a function $f_n \in F_n$ such that f_n does not have Σ_3^k circuits of size $2^{n-o(n)}$ with bottom fan-in $k(n) = O(1)$ (resp. $n^{o(1)}$). Also assume that given $i \in [1, |F_n|]$ in binary and $x \in \{0,1\}^n$, there is a polynomial-time algorithm for evaluating the i'th function in F_n on x. Then there is a Boolean function $g \in P$ such that g does not have linear-size logarithmic-depth series-parallel circuits (resp. linear-size logarithmic-depth circuits).*

The proof is omitted in this version.

References

1. Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
2. Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *Proceedings of IEEE Conference on Computational Complexity*, pages 252–260, 2006.
3. Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for AC0. In *Proceedings of Symposium on Discrete Algorithms*, page To appear, 2012.
4. Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
5. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 62(4):512–530, 2001.
6. Sharad Malik and Lintao Zhang. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.
7. Peter Bro Miltersen, Jaikumar Radhakrishnan, and Ingo Wegener. On converting cnf to dnf. *Theoretical Computer Science*, 347(1–2):325–335, 2005.
8. Ramamohan Paturi, Pavel Pudlak, Mike Saks, and Francis Zane. An improved exponential-time algorithm for k-sat. In *Proceedings of 39th International Symposium on Foundations of Computer Sciece (FOCS)*, pages 628–637, 1998.
9. Ramamohan Paturi, Pavel Pudlak, and Francis Zane. Satisfiability coding lemma. In *Proceedings of 38th International Symposium on Foundations of Computer Science (FOCS)*, pages 566–574, 1997.
10. Rahul Santhanam. Fighting perebor: New and improved algorithms for formula and QBF satisfiability. In *Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 183–192, 2010.
11. Rahul Santhanam and Srikanth Srinivasan. On the limits of sparsification. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:131, 2011.
12. Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005.
13. L. G. Valiant. Graph-theoretic arguments in low-level complexity. In J. Gruska, editor, *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science*, volume 53 of *LNCS*, pages 162–176, Tatranská Lomnica, Czechoslovakia, September 1977. Springer.
14. Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 231–240, 2010.
15. Ryan Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of 26th Annual IEEE Conference on Computational Complexity*, pages 115–125, 2011.