Random CNFs are Hard for Cutting Planes

Noah Fleming, Denis Pankratov, Toniann Pitassi, Robert Robere

March 8, 2017

Abstract

The random k-SAT model is the most important and well-studied distribution over k-SAT instances. It is closely connected to statistical physics; it is used as a testbench for satisfiablity algorithms, and lastly average-case hardness over this distribution has also been linked to hardness of approximation via Feige's hypothesis. In this paper, we prove that any Cutting Planes refutation for random k-SAT requires exponential size, for k that is logarithmic in the number of variables, and in the interesting regime where the number of clauses guarantees that the formula is unsatisfiable with high probability.

1 Introduction

The Satisfiability (SAT) problem is perhaps the most famous problem in theoretical computer science, and significant effort has been devoted to understanding randomly generated SAT instances. The most well-studied random SAT distribution is the random d-SAT model, $\mathcal{F}(m, n, d)$, where a random d-CNF over n variables is chosen by uniformly and independently selecting m clauses from the set of all possible clauses on d distinct variables. The random d-SAT model is widely studied for several reasons. First, it is an intrinsically natural model analogous to the random graph model, and closely related to phase transitions and structural phenomena occurring in statistical physics. Second, the random d-SAT model gives us a testbench of empirically hard examples which are useful for comparing and analyzing SAT algorithms; in fact, some of the better practical ideas in use today originated from insights gained by studying the performance of algorithms on this distribution and the properties of typical random instances.

Third, and most relevant to the current work, the difficulty of solving random d-SAT instances above the threshold (in the regime where the formula is almost certainly unsatisfiable) has recently been connected to worst-case inapproximability [12]. Feige's hypothesis states that there is no efficient algorithm to certify unsatisfiability of random 3-SAT instances for certain parameter regimes of (m, n, d), and he shows that this hard-on-average assumption for 3-SAT implies worst-case inapproximability results for many NP-hard optimization problems. The hypothesis was generalized to d-SAT as well as to any CSP, thus exposing more links to central questions in approximation algorithms and the power of natural SDP algorithms [4]. The importance of understanding the difficulty of solving random d-SAT instances in turn

makes random d-SAT an important family of formulas for propositional proof complexity, since superpolynomial lower bounds for random d-SAT formulas in a particular proof system show that *any* complete and efficient algorithm based on the proof system will perform badly on random d-SAT instances. Furthermore, since the proof complexity lower bounds hold in the unsatisfiable regime, they are directly connected to Feige's hypothesis.

Remarkably, determining whether or not a random SAT instance from the distribution $\mathcal{F}(m, n, d)$ is satisfiable is controlled quite precisely by the ratio $\Delta = m/n$, which is called the *clause density*. A simple counting argument shows that $\mathcal{F}(m, n, d)$ is unsatisfiable with high probability for $\Delta > 2^d \ln 2$. The famous satisfiability threshold conjecture asserts that there is a constant c_d such that random d-SAT formulas of clause density Δ are almost certainly satisfiable for $\Delta < c_d$ and almost certainly unsatisfiable if $\Delta > c_d$, where c_d is roughly $2^d \ln 2$. In a major recent breakthrough, the conjecture was resolved for large values of d [11].

From the perspective of proof complexity, the density parameter Δ also plays an important role in the *difficulty* of refuting unsatisfiable CNF formulas. For instance, in Resolution, which is arguably the simplest proof system, the complexity of refuting random d-SAT formulas is now very well understood in terms of Δ. In a seminal paper, Chvatal and Szemeredi [10] showed that for any fixed Δ above the threshold there is a constant κ_Δ such that random d-SAT requires size $\exp(\kappa_\Delta n)$ Resolution refutations with high probability. In their proof, the drop-off in κ_Δ is doubly exponential in Δ, making the lower bound trivial when the number of clauses is larger than $n \log^{1/4} n$ (and thus does not hold when d is large.) Improved lower bounds [5, 7] proved that the drop-off in κ_Δ is at most polynomial in Δ. More precisely, they prove that a random d-SAT formula with at most $n^{(d+2)/4}$ clauses requires exponential size Resolution refutations. Thus for all values of d, even when the number of clauses is way above the threshold, Resolution refutations are exponentially long. They also give asymptotically matching upper bounds, showing that there are DLL refutations of size $\exp(n/\Delta^{1/(d-2)})$.

Superpolynomial lower bounds for random d-SAT formulas are also known for other weak proof systems such as the polynomial calculus and $\mathsf{Res}(k)$ [1, 6], and random d-SAT is also conjectured to be hard for stronger semi-algebraic proof systems. In particular, it is a relatively long-standing open problem to prove superpolynomial size lower bounds for Cutting Planes refutations of random d-SAT. As alluded to earlier, this potential hardness (and even more so for the semi-algebraic SOS proof system) has been linked to hardness of approximation.

In this paper, we focus on the *Chvatal-Gomory Cutting Planes* proof system and some of its generalizations. A proof in this system begins with a set of unsatisfiable linear integral inequalities, and new integral inequalities are derived by (i) taking nonnegative linear combinations of previous lines, or (ii) dividing a previous inequality through by 2 (as long as all coefficients on the left-hand side are even) and then rounding up the constant term on the right-hand side. The goal is to derive the "false" inequality $0 \geq 1$ with as few derivation steps as possible. This system can be generalized in several natural ways. In *Semantic Cutting Planes*, there are no explicit rules – a new linear inequality can be derived from two previous ones as long as it follows soundly. A further generalization of both CP and Semantic CP is the CC-proof system, where now every line is only required to have low (deterministic or real) communication complexity; like Semantic CP, a new line can be derived from two previous ones as long as the derivation is sound.

The main result of this paper is a new proof method for obtaining Cutting Planes lower bounds, and we apply it to prove the first nontrivial lower bounds for the size of Cutting Planes refutations of random d-SAT instances. Specifically we prove that for $d = \Theta(\log n)$ and m in the unsatisfiable regime, with high probability random d-SAT requires exponential-size Cutting Planes refutations. Our main result holds for the other generalizations mentioned above (Semantic CP and CC-proofs).

We obtain the lower bound by establishing an *equivalence* between proving such lower bounds and proving a corresponding monotone circuit lower bound. Said a different way, we generalize the interpolation method so that it applies to *any* unsatisfiable family of formulas. Namely, we show that proving superpolynomial size lower bounds for any formula for Cutting Planes amounts to proving a monotone circuit lower bound for certain yes/no instances of the monotone CSP problem. Applying this equivalence to random d-SAT instances, we reduce the problem to that of proving a monotone circuit lower bound for a specific family of yes/no instances of the monotone CSP problem. We then apply the symmetric method of approximations in order to prove exponential monotone circuit lower bounds for our monotone CSP problem.

In recent private communication with Pavel Hrubes and Pavel Pudlák we have learned that they have independently proven a similar theorem.

1.1 Related Work

Exponential lower bounds on lengths of refutations are known for CP, Semantic CP, and low-weight CC-proofs) [9, 13, 19] These lower bounds were obtained using the method of interpolation [18]. A lower bound proof via interpolation begins with a special type of formula – *an interpolant*. Given two disjoint NP sets U and V an interpolant formula has the form $A(x, y) \wedge B(x, z)$ where the A-part asserts that $x \in U$, as verified by the NP-witness y, and the B-part asserts that $x \in V$, as verified by the NP-witness z. The prominent example in the literature is the clique/coclique formula where U is the set of all graphs with the clique number at least k, and V is the set of all $(k-1)$-colorable graphs. Feasible interpolation for a proof system amounts to showing that if an interpolant formula has a short proof then we can extract from the proof a small monotone circuit for separating U from V. Thus lower bounds follow from the celebrated monotone circuit lower bounds for clique [2, 20].

Despite the success of interpolation, it has been quite limited since it only applies to "split" formulas. In particular, the only family of formulas for which are known to be hard for (unrestricted) Cutting Planes are the clique-coclique formulas. In contrast, for Resolution we have a clean combinatorial characterization for when a formula does or doesn't admit a short Resolution refutation [3, 7]; we would similarly like to understand the strength of Cutting Planes with respect to arbitrary formulas and most notably for random d-SAT formulas and Tseitin formulas.

Our main equivalence is an adaptation of the earlier work combined with a key reduction between search problems and monotone functions established in [14]. With this reduction in hand, our main proof is very similar to both [9] and [21]. [9] proved this equivalence for the special case of the clique-coclique formulas. Namely they showed that low-weight CC-

proofs for this particular formula are equivalent to monotone circuits for the corresponding sets U, V. Our argument is essentially the same as theirs, only we realize that it holds much more generally for *any* unsatisfiable CNF and partition of the variables, and the corresponding set of Yes/No instances of CSP.

On the other hand, Razborov [21] proved the equivalence between PLS communication games (for KW games) and monotone circuits. The construction in our proof is essentially equivalent to his but bypasses PLS and proves a direct equivalence between monotone circuits and CC-proofs. We could have alternatively proven our equivalence via: (1) Razborov's equivalence between monotone circuits (for a monotone function) and PLS communication games (for the associated KW game), and then (2) an equivalence between PLS communication games (for a monotone KW game) and CC-proofs (for the search problem associated with the KW game). Inspired by [22], we give a direct argument which is (somewhat) simpler.

2   Definitions and Preliminaries

If $x, y \in \{0, 1\}^n$ then we write $x \leq y$ if $x_i \leq y_i$ for all i. A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is *monotone* if $f(x) \leq f(y)$ whenever $x \leq y$. If f is monotone then an input $x \in \{0, 1\}^n$ is a *maxterm* of f if $f(x) = 0$ but $f(x') = 1$ for any x' obtained from x by flipping a single bit from 0 to 1; dually, x is a *minterm* if $f(x) = 1$ but $f(x') = 0$ for any x' obtained by flipping a single bit of x from 1 to 0. More generally, if $f(x) = 1$ we call x an *accepting instance* or a *yes instance*, while if $f(x) = 0$ then we call x a *rejecting instance* or a *no instance*. If x is any yes instance of f and y is any no instance of f then there exists an index $i \in [n]$ such that $x_i = 1, y_i = 0$, as otherwise we would have $x \leq y$, contradicting the fact that f is monotone. If $f, g, h : \{0, 1\}^n \rightarrow \{0, 1\}$ are boolean functions on the same domain then $f, g \vDash h$ if for all $x \in \{0, 1\}^n$ we have $f(x) \land g(x) \implies h(x)$.

A *monotone circuit* is a circuit in which the only gates are $\land$ or $\lor$ gates. A *real monotone circuit* is a circuit in which each internal gate has two inputs and computes any function $\phi(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ which is monotone nondecreasing in its arguments.

Definition 2.1. A *linear integral inequality* in variables $x = (x_1, \ldots, x_n)$ with coefficients $a = (a_1, \ldots, a_n) \in \mathbb{Z}^n$ and constant term $b \in \mathbb{Z}$ is an expression

$$a^T x \geq b.$$

Definition 2.2. Given a system of linear integral inequalities $Ax \geq b$, where $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, a *cutting planes proof* of an inequality $a^T x \geq c$ is a sequence of inequalities

$$a_1{}^T x \geq c_1, a_2{}^T x \geq c_2, \ldots, a_\ell{}^T x \geq c_\ell,$$

such that $a_\ell = a$, $c_\ell = c$ and every inequality $i \in [\ell]$ satisfies either

- $a_i{}^T x \geq c_i$ appears in $Ax \geq b$,

- $a_i{}^T x \geq c_i$ is a Boolean axiom, i.e., $x_j \geq 0$ or $-x_j \geq -1$ for some j,

- there exists $j, k < i$ such that $a_i^T x \geq c_i$ is the sum of the linear inequalities $a_j^T x \geq c_j$ and $a_k^T x \geq c_k$,

- there exists $j < i$ and a positive integer d dividing every coefficient in a_j such that $a_i = a_j/d$ and $c_i = \lceil c_j/d \rceil$.

The *length* of the proof is ℓ, the number of lines. If all coefficients and constant terms appearing in the cutting planes proof are bounded by $O(\mathsf{poly}(n))$, then the proof is said to be of *low weight*.

Let $\mathcal{F} = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable CNF formula over variables $z_1, \ldots, z_n$. For any clause C let C^- denote the set of variables appearing negated in the clause and let C^+ denote variables occurring positively in the clause. Each clause C in $\mathcal{F}$ can be encoded as a linear integral inequality as

$$\sum_{z_i \in C^+} z_i + \sum_{z_i \in C^-} (1 - z_i) \geq 1.$$

Thus each unsatisfiable CNF can be translated into a system of linear integral inequalities $Az \geq b$ with no $0/1$ solutions. A *cutting planes (CP) refutation* of this system is a cutting planes proof of the inequality $0 \geq 1$ from $Ax \geq b$.

Definition 2.3. Let $\mathcal{F} = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable k-CNF on n variables. A *semantic refutation* of $\mathcal{F}$ is a sequence

$$L_1, L_2, \ldots, L_\ell$$

of boolean functions $L_i : \{0,1\}^n \rightarrow \{0,1\}$ such that

1. $L_i = C_i$ for all $i = 1, 2, \ldots, m$.

2. $L_\ell = 0$, the constant 0 function.

3. For all $i > m$ there exists $j, k < i$ such that $L_j, L_k \vDash L_i$.

The *length* of the refutation is ℓ.

We will be particularly interested in semantic refutations where the boolean functions can be computed by short communication protocols.

Definition 2.4. Let $\mathcal{F} = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable CNF on $n = n_1 + n_2$ variables, and let $X = \{x_1, x_2, \ldots, x_{n_1}\}, Y = \{y_1, \ldots, y_{n_2}\}$ be a partition of the variables. A CC_k-refutation of $\mathcal{F}$ with respect to the partition (X, Y) is a semantic refutation

$$L_1, \ldots, L_\ell$$

of $\mathcal{F}$ such that each function L_i in the proof can be computed by a k-bit communication protocol with respect to the partition (X, Y).

Since any linear integral inequality $ax + by \geq c$ with polynomially bounded weights can be evaluated by a trivial $O(\log n)$-bit communication protocol (just by having Alice evaluating ax and sending the result to Bob), it follows that low-weight cutting planes proofs are also $\mathsf{CC}_{O(\log n)}$-proofs. We can similarly define a proof system which can simulate any cutting planes proof by strengthening the type of communication protocol.

Definition 2.5. A k-round *real communication protocol* is communication protocol between two players, Alice and Bob, where Alice receives an input $x \in \mathcal{X}$ and Bob receives $y \in \mathcal{Y}$. In each round, Alice and Bob each send real numbers α, β to a "referee", who responds with a single bit b which is 1 if $\alpha \geq \beta$ and 0 otherwise. After k rounds of communication, the players output a bit b. The protocol computes a function $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ the protocol outputs $F(x, y)$.

Definition 2.6. Let $\mathcal{F} = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable CNF on $n = n_1 + n_2$ variables $X = \{x_1, \ldots, x_{n_1}\}$ and $Y = \{y_1, \ldots y_{n_2}\}$. An RCC_k-refutation of $\mathcal{F}$ is a semantic refutation

$$L_1, L_2, \ldots, L_\ell$$

in which each function L_i can be computed by a k-round real communication protocol with respect to the variable partition X, Y.

It is clear that *any* linear integral inequality $ax + by \geq c$ can be evaluated by a 1-round real communication protocol, and so it follows that a cutting planes refutation of $\mathcal{F}$ is also an RCC_1-refutation of $\mathcal{F}$. We record each of these observations in the next proposition.

Proposition 2.7. *Let $\mathcal{F}$ be an unsatisfiable CNF on variables $z_1, z_2, \ldots, z_n$, and let X, Y be any partition of the variables into two sets. Any length-ℓ low-weight cutting planes refutation of $\mathcal{F}$ is a length-ℓ $\mathsf{CC}_{O(\log n)}$-refutation of $\mathcal{F}$. Similarly, any length-ℓ cutting planes refutation of $\mathcal{F}$ is a length-ℓ RCC_1-refutation of $\mathcal{F}$.*

2.1 Total Search Problems and Monotone CSP-SAT

In this section we review the equivalence between the search problem associated with an unsatisfiable CNF formula, and the Karchmer-Wigderson (KW) search problem for a related (partial) monotone function.

Definition 2.8. Let n_1, n_2, m be positive integers, and let $\mathcal{X}, \mathcal{Y}$ be finite sets. A *total search problem* is a relation $\mathcal{R} \subseteq \mathcal{X}^{n_1} \times \mathcal{Y}^{n_2} \times [m]$ where for each $(x, y) \in \mathcal{X}^{n_1} \times \mathcal{Y}^{n_2}$, there is an $i \in [m]$ such that $\mathcal{R}(x, y, i) = 1$. We refer to $x \in \mathcal{X}^{n_1}$ as Alice's input and $y \in \mathcal{Y}^{n_2}$ as Bob's input. The search problem is *d-local* if for each $i \in [m]$ we have that $\mathcal{R}(*, *, i)$ depends on a fixed set of at most d coordinates of x (it may depend on any number of y coordinates).

A standard example of a d-local search problem is the search problem associated with unsatisfiable d-CNFs.

Definition 2.9. Let $\mathcal{F}$ be an unsatisfiable d-CNF formula with m clauses and n variables $z_1, \ldots, z_n$. Consider any partition of $z_1, z_2, \ldots, z_n$ into two sets $x_1, x_2, \ldots, x_{n_1}$ and $y_1, y_2, \ldots, y_{n_2}$. The search problem Search($\mathcal{F}$) with respect to this partition takes as input an assignment $x \in \{0,1\}^{n_1}$ and $y \in \{0,1\}^{n_2}$ and outputs the index $i \in [m]$ of a violated clause under this assignment.

This problem is clearly d-local since each clause can contain at most d variables from $x_1, x_2, \ldots, x_{n_1}$. Associated with this search problem is the following monotone variant of the constraint satisfaction problem.

Definition 2.10. Let $H = (L \cup R, E)$ be a bipartite graph such that each vertex $v \in L$ has degree at most d, and let $m = |L|$ and $n = |R|$. Let Σ be a finite alphabet. A *constraint satisfaction problem* (CSP) $\mathcal{H}$ with topology H and alphabet Σ is defined as follows. The vertices in L are thought of as the set of *constraints*, and the vertices in R are thought of as a set of *variables*; thus for each vertex $i \in L$ we let vars(i) denote the neighbourhood of i. For each vertex $i \in L$ the CSP has an associated boolean function $\mathrm{TT}_u : \Sigma^{\mathsf{vars}(i)} \to \{0,1\}$ called the *truth table* of i that encodes the set of "satisfying" assignments to the constraint associated with i. An assignment $\alpha \in \Sigma^n$, thought of as a Σ-valued assignment to the variables R, *satisfies* the CSP $\mathcal{H}$ if for each $i \in L$ we have $\mathrm{TT}_i(\alpha \restriction \mathsf{vars}(i)) = 1$, otherwise the assignment *falsifies* the CSP.

For each $i \in [m]$ and $\alpha \in \Sigma^{\mathsf{vars}(i)}$ we abuse notation and let $\mathrm{TT}_i(\alpha)$ represent the boolean variable corresponding to this entry of the truth table for the constraint i.

Definition 2.11. Let $H = (L \cup R, E)$ be a bipartite graph such that each vertex $i \in L$ has degree at most d, and let $m = |L|$ and $n = |R|$. We think of H as encoding the topology of a constraint satisfaction problem, where each vertex $i \in L$ represents a *constraint* of the CSP and each $i \in R$ represents a *variable* of the CSP. Let Σ be a finite alphabet, and let $N = \sum_{i=1}^{m} |\Sigma|^{\mathsf{vars}(i)} \leq m|\Sigma|^d$. The monotone function $\mathrm{CSP\text{-}SAT}_{H,\Sigma} : \{0,1\}^N \to \{0,1\}$ is defined as follows. An input $x \in \{0,1\}^N$ encodes a CSP $\mathcal{H}(x)$ by specifying for each vertex $u \in L$ its truth table

$$\mathrm{TT}_u^x : \Sigma^{\mathsf{vars}(u)} \to \{0,1\}.$$

Given an assignment $x \in \{0,1\}^N$ the function $\mathrm{CSP\text{-}SAT}_{H,\Sigma}(x) = 1$ if and only if the CSP $\mathcal{H}(x)$ is satisfiable. This function is clearly monotone since for any $x, y \in \{0,1\}^N$ with $x \leq y$, any satisfying assignment for the CSP $\mathcal{H}(x)$ is also a satisfying assignment for the CSP $\mathcal{H}(y)$.

Next we show how to relate d-local total search problems and the CSP-SAT problem. Let $\mathcal{R} \subseteq \mathcal{X}^{n_1} \times \mathcal{Y}^{n_2} \times [m]$ be a d-local total search problem. Associated with $\mathcal{R}$ is a bipartite *constraint graph* $H_\mathcal{R}$ encoding for each $i \in [m]$ the coordinates in $\mathcal{X}^{n_1}$ on which $\mathcal{R}(*, *, i)$ depends. Formally, the constraint graph is the bipartite graph $H_\mathcal{R} = (L \cup R, E)$ with $L = [m]$, $|R| = [n_1]$, and for each pair $(i, j) \in L \times R$ we add the edge if $\mathcal{R}(*, *, i)$ depends on the variable x_j. Note that each vertex $u \in L$ has degree at most d, since the original search problem is d-local.

Given $\mathcal{R}$ and its corresponding constraint graph we can give a natural way to construct accepting and rejecting instances of $\mathrm{CSP\text{-}SAT}_{H_\mathcal{R}, \mathcal{X}}$ from $\mathcal{X}^{n_1}$ and $\mathcal{Y}^{n_2}$. To reduce clutter,

given a d-local total search problem $\mathcal{R}$ we abuse notation and write

$$\text{CSP-SAT}_{\mathcal{R}} := \text{CSP-SAT}_{H_{\mathcal{R}},\mathcal{X}}.$$

Accepting Instances $\mathcal{U}$. For any $x \in \mathcal{X}^{n_1}$ we construct an accepting input $\mathcal{U}(x)$ of CSP-SAT$_{\mathcal{R}}$ as follows. For each vertex $i \in L$ we define the corresponding truth table TT_i by setting $\text{TT}_i(\alpha) = 1$ if $x \upharpoonright \text{vars}(i) = \alpha$ and $\text{TT}_i(\alpha) = 0$ otherwise.

Rejecting Instances $\mathcal{V}$. For any $y \in \mathcal{Y}^{n_2}$ we construct a rejecting input $\mathcal{V}(y)$ of CSP-SAT$_{\mathcal{R}}$ as follows. For each vertex $i \in L$ and each $\alpha \in \Sigma^{\text{vars}(i)}$ we set

$$\text{TT}_i(\alpha) = 0 \iff \mathcal{R}(\alpha, y, i) \text{ holds.}$$

Given $x \in \mathcal{X}^{n_1}$ it is easy to see that $\mathcal{U}(x)$ is a satisfying assignment for CSP-SAT$_{\mathcal{R}}$ since x is a satisfying assignment for the corresponding CSP. The rejecting instances require a bit more thought. Let $y \in \mathcal{Y}^{n_2}$ and consider the rejecting instance $\mathcal{V}(y)$ as defined above. Suppose by way of contradiction that the corresponding CSP $\mathcal{H}_{\mathcal{R}}(\mathcal{V}(y))$ is satisfiable, and let $x \in \mathcal{X}^{n_1}$ be the satisfying assignment for the CSP. It follows by definition of the rejecting instances that $\mathcal{R}(x, y, u)$ does not hold for any u, implying that $\mathcal{R}$ is not total.

3 Relating Proofs and Circuits

In this section we relate CC_d-proofs and monotone circuits, as well as RCC_1-proofs and *real* monotone circuits.

Theorem 3.1. *Let $\mathcal{F}$ be an unsatisfiable CNF formula on n variables and let $X = \{x_1, \ldots, x_{n_1}\}$, $Y = \{y_1, \ldots, y_{n_2}\}$ be any partition of the variables. Let k be a positive integer. If there is a CC_k refutation of $\mathcal{F}$ with respect to the partition (X, Y) of length ℓ, then there is a monotone circuit separating the accepting and rejecting instances $\mathcal{U}(\{0, 1\}^{n_1}), \mathcal{V}(\{0, 1\}^{n_2})$ of $\text{CSP-SAT}_{\mathsf{Search}(\mathcal{F})}$ of size $O(2^k \ell)$.*

Proof. Let $\mathcal{F} = C_1 \wedge \ldots \wedge C_m$ over variables $x_1, \ldots, x_{n_1}, y_1, \ldots, y_{n_2}$. Let P be a CC_k-proof for $\mathcal{F}$ with ℓ lines. Order the lines in P as $L_1, L_2, \ldots, L_\ell$, where the final line L_ℓ is the identically false formula, and each earlier line is either a clause, or follows semantically from two earlier lines.

We build the circuit for $\text{CSP-SAT}_{\mathsf{Search}(\mathcal{F})}$ that separates $\mathcal{U}, \mathcal{V}$ by induction on ℓ. For each line L in the proof, there are 2^k possible histories h, each with an associated monochromatic rectangle $R_L(h)$. A rectangle h is *good* for L if it is 0-monochromatic. For every line L and each good history h for L, we will build a circuit $\mathcal{C}_h^L$ that correctly "separates" x and y for each $(x, y) \in R_L(h)$. By this, we mean that the circuit $\mathcal{C}_h^L$ outputs 1 on $\mathcal{U}(x)$ (the 1-input associated with x) and outputs 0 on $\mathcal{V}(y)$ (the 0-input associated with y).

For each leaf in the proof, the associated line L is a clause C_i of $\mathcal{F}$. The communication protocol for C_i is a two-bit protocol where Alice/Bob each send 0 iff their inputs are α, β such that $C_i(\alpha, \beta) = 0$. Thus there is only one good (0-monochromatic) rectangle with history

$h = 00$. This pair α, β corresponds to the variable $\mathrm{TT}_i(\alpha)$, and we define the circuit $\mathcal{C}_h^L$ corresponding to line $L = C_i$ and good history $h = 00$ to be the variable $\mathrm{TT}_i(\alpha)$.

Now suppose that L is derived from L_1 and L_2, and inductively we have circuits $\mathcal{C}_{h'}^{L_1}$, $\mathcal{C}_{h''}^{L_2}$ for each history h' good for L_1 and h'' good for L_2. Given a good history h for L, we will show how to build the circuit $\mathcal{C}_h^L$. It will use all of the the circuits that were built for L_1 and L_2 ($\{\mathcal{C}_{h'}^{L_1}, \mathcal{C}_{h''}^{L_2}\}$ for all good h') and an additional 2^k gates. To build $\mathcal{C}_h^L$ we will construct a *stacked* protocol tree for L, corresponding to first running the communication protocol for L_1 and then running the communication protocol for L_2. This will give us a height $2k$ (full) binary tree, T, where the top part is the communication protocol tree for L_1, with protocol trees for L_2 hanging off of each of the leaves. We label each of the leaves of this stacked tree with a circuit from $\{\mathcal{C}_{h'}^{L_1}, \mathcal{C}_{h''}^{L_2}\}$ as follows. Consider a path labelled $h_1 h_2$ in T, where h_1 is the history from running L_1 and h_2 is the history from running L_2. By soundness, either the rectangle $R_L(h) \cap R_{L_1}(h_1)$ is 0-monochromatic, or the rectangle $R_L(h) \cap R_{L_2}(h_2)$ is 0-monochromatic. In the first case, we will label this leaf with $\mathcal{C}_{h_1}^{L_1}$ and otherwise we will label this leaf with $\mathcal{C}_{h_2}^{L_2}$. Now we will label the internal vertices of the stacked tree with a gate: if a node corresponds to Alice speaking, then we label the node with an $\vee$ gate, and otherwise if the node corresponds to Bob speaking, then we label the node with an $\wedge$ gate. The resulting circuit has size 2^k plus the sizes of the subcircuits, and thus the total circuit size is $2^k \ell$. The theorem is therefore immediately implied by the following claim.

Claim. The circuit resulting from the above construction satisfies: for each line L in P, and for each good history h for L, $\mathcal{C}_h^L$ will be correct for all $(x, y) \in R_L(h)$.

Proof of Claim. If L is an axiom, then L is a clause, C_i. The communication protocol for C_i is a two-bit protocol where Alice and Bob each send 0 iff their part of C_i evalutes to 0. There is only one good (0-monochromatic) history, $h = 00$. If $(x, y) \in R_L(h)$ then $C_i(x, y) = 0$ by definition. Let $\alpha = x \upharpoonright \mathsf{vars}(C_i)$. In our construction the circuit corresponding to $\mathcal{C}_h^L$ is labelled by the variable $\mathrm{TT}_i(\alpha)$, and it is easy to check that $\tilde{x}$ sets $\mathrm{TT}_i(\alpha)$ to true, and $\tilde{y}$ sets $\mathrm{TT}_i(\alpha)$ to false.

If L is not an axiom, then we will prove the lemma by proving the following stronger statement by induction: For each line L (derived from previous lines L_1 and L_2), and for each node v in the stacked protocol tree for L, with corresponding (sub)history $h' = h_1 h_2$, the subcircuit $\mathcal{C}_{h'}^L$ associated with vertex v is correct on all $(x, y) \in R_L(h) \cap R_{L_1}(h_1) \cap R_{L_2}(h_2)$.

Fix a line L that is not an axiom. For the base case, suppose that v is a leaf of the stacked protocol tree for L with history $h' = h_1 h_2$. Then by soundness either (i) $R_L(h) \cap R_{L_1}(h_1) = 0$ or (ii) $R_L(h) \cap R_{L_1}(h_2) = 0$. In case (i) we labelled v by $\mathcal{C}_{h_1}^{L_1}$. Since $R_L(h) \cap R_{L_1}(h_1) = 0$, $R_{L_1}(h_1) = 0$ and therefore $\mathcal{C}_{h_1}^{L_1}$ is defined and is correct on all $(x, y) \in R_{L_1}(h_1)$, so it is correct on all $(x, y) \in R_L(h) \cap R_{L_1}(h_1) \cap R_{L_2}(h_2)$. A similar argument holds in case (ii).

For the inductive step, let v be a nonleaf node in the protocol tree with history h' and assume that Alice owns v. The rectangle $R_L(h) \cap R_{L_1}(h_1) \cap R_{L_2}(h_2) = A \times B$ is partitioned into $A_0 \times B$ and $A_1 \times B$, where

1. $A = A_0 \cup A_1$,

9

2. $A_0 \times B$ is the rectangle with history $h'0$,

3. $A_1 \times B$ is the rectangle with history $h'1$.

Given $(x, y) \in R_L(h) \cap R_{L_1}(h_1) \cap R_{L_2}(h_2)$, since $\mathcal{C}_{h'0}^L$ is correct on all $(x, y) \in A_0 \times B$ and $\mathcal{C}_{h'1}^L$ is correct on all $(x, y) \in A_1 \times B$, it follows that $\mathcal{C}_h^L = \mathcal{C}_{h'0}^L \vee \mathcal{C}_{h'1}^L$ is correct on all $(x, y) \in A \times B$. To see this, observe that if $x \in A_0$, then $\mathcal{C}_{h'0}^L(\mathcal{U}(x)) = 1$ and therefore

$$\mathcal{C}_h^L(\mathcal{U}(x)) = \mathcal{C}_{h'0}^L(\mathcal{U}(x)) \vee \mathcal{C}_{h'1}^L(\mathcal{U}(x)) = 1.$$

Similarly, if $x \in A_1$, then $\mathcal{C}_{h'1}^L(\mathcal{U}(x)) = 1$ and therefore

$$\mathcal{C}_h^L(\mathcal{U}(x)) = \mathcal{C}_{h'0}^L(\mathcal{U}(x)) \vee \mathcal{C}_{h'1}^L(\mathcal{U}(x)) = 1.$$

Finally if $y \in B$ then both $\mathcal{C}_{h'0}^L(\mathcal{V}(y)) = \mathcal{C}_{h'1}^L(\mathcal{V}(y)) = 0$ and therefore

$$\mathcal{C}_h^L(\mathcal{V}(y)) = \mathcal{C}_{h'0}^L(\mathcal{V}(y)) \vee \mathcal{C}_{h'1}^L(\mathcal{V}(y)) = 0.$$

A similar argument holds if v is an internal node in the protocol tree that Bob owns (and is therefore labelled by an AND gate). $\square$

The converse direction is much easier.

Theorem 3.2. *If there is a monotone circuit separating these inputs of* $\text{CSP-SAT}_{\text{Search}(\mathcal{F})}$ *of size ℓ, then there is a CC_2-refutation of $\mathcal{F}$ of length ℓ with respect to this partition of the variables.*

Proof. In the other direction, we show that from a small monotone circuit $\mathcal{C}$ for $\text{CSP-SAT}_{\text{Search}(\mathcal{C})}$ that separates $\mathcal{U}(\{0, 1\}^{n_1})$ and $\mathcal{V}(\{0, 1\}^{n_2})$, we can construct a small CC_2-proof for $\mathcal{F}$, where Alice gets $x \in \{0, 1\}^{n_1}$ and Bob gets $y \in \{0, 1\}^{n_2}$. The lines/vertices of the refutation will be in 1-1 correspondence with the gates of $\mathcal{C}$. The protocol is constructed inductively from the leaves of $\mathcal{C}$ to the root. For a gate g of $\mathcal{C}$, let U_g be those inputs $u \in \mathcal{U}(\{0, 1\}^{n_1})$ such that $g(u) = 1$, and let V_g be those inputs $v \in \mathcal{V}(\{0, 1\}^{n_2})$ such that $g(v) = 0$. At each gate g we will prove that for every pair $(u, v) \in U_g \times V_g$ and for every (x, y) such that $u = \mathcal{U}(x), v = \mathcal{V}(y)$, the protocol R_g on input (x, y) will output 0. Since the output gate of $\mathcal{C}$ is correct for all pairs, this will achieve our desired protocol.

At a leaf ℓ labelled by some variable $\text{TT}_j(\alpha)$, the pairs associated with this leaf must have $\text{TT}_j(\alpha) = 1$ in u and 0 in v, and thus we can define $R_\ell(x, y)$ to be 0 if and only if x is consistent with α and the clause C_j evaluates to false on (x, y). This is a 2-bit protocol, and by definition of the accepting and rejecting instances we have for all (x, y) satisfying $u = \mathcal{U}(x), v = \mathcal{V}(y)$ that $x \restriction \text{vars}(j) = \alpha$ and $\mathcal{R}(\alpha, y, j)$ holds.

Now suppose that g is a OR gate of $\mathcal{C}$, with inputs g_1, g_2. The protocol R_g on (x, y) is as follows. Alice privately simulates $\mathcal{C}_{g_1}(\mathcal{U}(x))$ and $\mathcal{C}_{g_2}(\mathcal{U}(x))$, and Bob simulates $\mathcal{C}_{g_1}(\mathcal{V}(y))$ and $\mathcal{C}_{g_2}(\mathcal{V}(y))$. If (i) either $\mathcal{C}_{g_1}(\mathcal{U}(x)) = 1$ or $\mathcal{C}_{g_2}(\mathcal{U}(x)) = 1$ and (ii) both $\mathcal{C}_{g_1}(\mathcal{V}(y)) = 0$ and $\mathcal{C}_{g_2}(\mathcal{V}(y)) = 0$, then they output 0, and otherwise they output 1. This is a 2-bit protocol, with Alice sending one bit to report whether or not condition (i) is satisfied, and Bob sending one bit to report if (ii) is satisfied.

Now, we want to show that for all (x, y) such that $\mathcal{C}_g(\mathcal{U}(x)) = 1$ and $\mathcal{C}_g(\mathcal{V}(y)) = 0$ we have that $R_g(x, y) = 0$. This is easy — since $g = g_1 \vee g_2$ we have that $\mathcal{C}_g(\mathcal{U}(x)) = 1$ and $\mathcal{C}_g(\mathcal{V}(y) = 0$ implies that either $\mathcal{C}_{g_1}(\mathcal{U}(x)) = 1$ or $\mathcal{C}_{g_2}(\mathcal{U}(x)) = 1$ and $\mathcal{C}_{g_1}(\mathcal{V}(y)) = 0$ and $\mathcal{C}_{g_2}(\mathcal{V}(y)) = 0$, implying that the protocol will output 0 on (x, y) by definition.

Similarly, if g is an AND gate, then again Alice privately simulates $\mathcal{C}_{g_1}(\mathcal{U}(x))$ and $\mathcal{C}_{g_2}(\mathcal{U}(x))$ and Bob privately simulates $\mathcal{C}_{g_2}(\mathcal{V}(y))$ and $\mathcal{C}_{g_2}(\mathcal{V}(y))$. If (i) $\mathcal{C}_{g_1}(\mathcal{U}(x)) = 1$ and $\mathcal{C}_{g_2}(\mathcal{U}(x)) = 1$ and (ii) either $\mathcal{C}_{g_2}(\mathcal{V}(y)) = 0$ or $\mathcal{C}_{g_2}(\mathcal{V}(y)) = 0$, then they ouput 0, and otherwise they output 1. By an analogous argument to the OR case, it's easy to see that the protocol will output 0 whenever $\mathcal{C}_g(\mathcal{U}(x)) = 1$ and $\mathcal{C}_g(\mathcal{V}(y)) = 0$. $\qquad\square$

The following theorem was recently proven [16], showing that RCC_1-proofs imply monotone real circuits for the associated search problem.

Theorem 3.3. *[16] Let $\mathcal{F}$ be an unsatisfiable CNF formula on n variables and let $X = \{x_1, \ldots, x_{n_1}\}$, $Y = \{y_1, \ldots, y_{n_2}\}$ be any partition of the variables. If there is a RCC_1 refutation of $\mathcal{F}$ with respect to the partition (X, Y) of length ℓ, then there is a monotone real circuit separating the accepting and rejecting instances $\mathcal{U}(\{0, 1\}^{n_1})$, $\mathcal{V}(\{0, 1\}^{n_2})$ of $\mathrm{CSP}\text{-}\mathrm{SAT}_{\mathsf{Search}(\mathcal{F})}$ of size polynomial in ℓ.*

In particular, the above theorem implies that for any family of formulas $\mathcal{F}$ and for any partition of the underlying variables into X, Y, a Cutting Planes refutation of $\mathcal{F}$ of size S implies a similar size monotone real circuit for separating the accepting and rejecting instances $\mathcal{U}(\{0, 1\}^{n_1})$, $\mathcal{V}(\{0, 1\}^{n_2})$ of $\mathrm{CSP}\text{-}\mathrm{SAT}_{\mathsf{Search}(\mathcal{F})}$.

4 Lower Bounds for Random CNFs

In this section using Theorem 3.3 we prove lower bounds for RCC_1-refutations (and therefore cutting planes refutations) of uniformly random d-CNFs with sufficient clause density.

Definition 4.1. Let $\mathcal{F}(m, n, d)$ denote the distribution of random d-CNFs on n variables obtained by sampling m clauses (out of the $\binom{n}{d}2^d$ possible clauses) uniformly at random with replacement.

The proof is delayed to Section 4.2; to get a feeling for the proof, we first prove an easier lower bound for a simpler distribution of *balanced* random CNFs.

4.1 Balanced Random CNFs

Definition 4.2. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ be two disjoint sets of variables, and the distribution $\mathcal{F}(m, n, d)^{\otimes 2}$ denotes the following distribution over $2d$-CNFs: First sample

$$\mathcal{F}^1 = C_1^1 \wedge C_2^1 \wedge \cdots \wedge C_m^1$$

from $\mathcal{F}(m, n, d)$ on the X variables, and then

$$\mathcal{F}^2 = C_1^2 \wedge C_2^2 \wedge \cdots \wedge C_m^2$$

11

from $\mathcal{F}(m, n, d)$ on the Y variables independently. Then output

$$\mathcal{F} = (C_1^1 \vee C_1^2) \wedge (C_2^1 \vee C_2^2) \wedge \cdots \wedge (C_m^1 \vee C_m^2).$$

This distribution shares the well-known property with $\mathcal{F}(m, n, d)$ that dense enough formulas are unsatisfiable with high probability.

Lemma 4.3. *Let $c > 2/\log e$ and let n be any positive integer. If $d \in [n]$ and $m \geq cn2^{2d}$ then $\mathcal{F} \sim \mathcal{F}(m, n, d)^{\otimes 2}$ is unsatisfiable with high probability.*

Proof. Fix any assignment (x, y) to the variables of $\mathcal{F}$. The probability that the ith clause is satisfied by the joint assignment is $1 - 1/2^{2d}$, and so the probability that *all* clauses are satisfied by the joint assignment is $(1 - 1/2^{2d})^m \leq e^{-m/2^{2d}}$, since the clauses are sampled independently. By the union bound, the probability that some joint assignment satisfies the formula is at most $2^{2n}e^{-m/2^{2d}} = 2^{2n-(\log e)m/2^{2d}} \leq 2^{2n-(\log e)cn} \leq 2^{-\Omega(n)}$. Thus, the probability that the formula is unsatisfiable is at least $1 - 2^{-\Omega(n)}$. $\square$

The main theorem of this section is that $\mathcal{F} \sim \mathcal{F}(m, n, d)^{\otimes 2}$ require large CC- and RCC-proofs, which is obtained by using Theorem 3.3 and applying the well-known method of symmetric approximations [8, 15] to obtain lower bounds on monotone circuits computing the function $\text{CSP-SAT}_{\text{Search}(\mathcal{F})}$. We use the following formalization of the method which is exposited in Jukna's excellent book [17]. First we introduce some notation: if $U \subseteq \{0, 1\}^N$, then for $r \in [N]$ and $b \in \{0, 1\}$ let

$$A_b(r, U) = \max_{I \subseteq [n]:|I|=r} |\{u \in U \mid \forall i \in I : u_i = b\}|.$$

Theorem 4.4 (Theorem 9.21 in Jukna). *Let $f : \{0, 1\}^N \to \{0, 1\}$ be a monotone boolean function and let $1 \leq r, s \leq N$ be any positive integers. Let $U \subseteq f^{-1}(1)$ and $V \subseteq f^{-1}(0)$ be arbitrary subsets of accepting and rejecting inputs of f. Then every real monotone circuit that outputs 1 on all inputs in U and 0 on all inputs in V has size at least*

$$\min \left\{ \frac{|U| - (2s)A_1(1, U)}{(2s)^{r+1}A_1(r, U)}, \frac{|V|}{(2r)^{s+1}A_0(s, V)} \right\}.$$

Next we state the main theorem of this section.

Theorem 4.5. *Let $d = 4\log n$ and $m = cn^2 2^d$ where $c > 2/\log e$ is some constant. Let $\mathcal{F} \sim \mathcal{F}(m, n, d)^{\otimes 2}$ with variable partition (X, Y), and let*

$$U = \mathcal{U}(\{0, 1\}^X), V = \mathcal{V}(\{0, 1\}^Y).$$

Then with high probability any real monotone circuit separating U and V has at least $2^{\tilde{\Omega}(n)}$ gates.

Corollary 4.6. *Let n be a sufficiently large positive integer, and let $d = 4\log n, m = n^6$. If $\mathcal{F} \sim \mathcal{F}(m, n, d)^{\otimes 2}$ then with high probability every RCC_1-refutation (and therefore, Cutting Planes refutation) of $\mathcal{F}$ has at least $2^{\tilde{\Omega}(n)}$ lines.*

Proof. Immediate consequence of Theorems 3.3 and 4.5. $\qquad\square$

The proof of Theorem 4.5 is rather straightforward, and comes down to the essential property that random d-CNFs are good expanders. The next lemma records the expansion properties we require of random CNFs; the proof is adapted from the notes of Salil Vadhan [23].

Lemma 4.7. *Let $0 < \varepsilon < 1$ be arbitrary, and let n be any sufficiently large positive integer. Let $d = 4 \log n$, $m = n^2 2^d$, and sample $\mathcal{F} \sim \mathcal{F}(m, n, d)$. For any subset $S \subseteq \mathcal{F}$ of clauses let $\mathsf{vars}(S)$ denote the subset of variables appearing in any clause of S. Any set $S \subseteq \mathcal{F}$ of size $s \le n/ed^2$ satisfies*

$$|\mathsf{vars}(S)| \ge (1 - \varepsilon)ds$$

with high probability.

Proof. Fix any set $S \subseteq \mathcal{F}$ of size s, and for each clause $C \in S$ sample the variables in C one at a time without replacement. Let $v_1, v_2, \ldots, v_{ds}$ denote the concatenation of all sequences of sampled variables over all $C \in S$. We say that variable v_i is a repeat if it has already occurred among $v_1, \ldots, v_{i-1}$. In order for $|\mathsf{vars}(S)| < (1 - \varepsilon)ds$ the concatenated sequence must have at least εds repeats, and the probability that variable v_i is a repeat is at most $(i-1)/n \le ds/n$. This implies that

$$\Pr[|\mathsf{vars}(S)| < (1 - \varepsilon)ds] \le \binom{ds}{\varepsilon ds} \left(\frac{ds}{n}\right)^{\varepsilon ds} \le \left(\frac{eds}{\varepsilon ds}\right)^{\varepsilon ds} \left(\frac{ds}{n}\right)^{\varepsilon ds} \le \left(\frac{1}{\varepsilon d}\right)^{\varepsilon ds}$$

using standard bounds on binomial coefficients and the fact that $s \le n/ed^2$. Thus

$$\Pr[\exists S : |S| = s, |\mathsf{vars}(S)| < (1 - \varepsilon)ds] \le m^s \left(\frac{1}{\varepsilon d}\right)^{\varepsilon ds},$$

and since $m = n^2 2^d$ and $d = 4 \log n$ we get that

$$s \log m \ll \varepsilon ds \log \varepsilon d$$

for sufficiently large n, implying the previous probability is $o(1)$. $\qquad\square$

Using the expansion lemma we are ready to prove Theorem 4.5.

Proof of Theorem 4.5. We shall apply Theorem 4.4 to $U = \mathcal{U}(\{0,1\}^n)$ and $V = \mathcal{V}(\{0,1\}^n)$ (cf. Section 2.1) with $r = s = n/ed^2$. Recall that $\mathcal{U}$ and $\mathcal{V}$ are the functions mapping x inputs to 1-inputs of $\mathrm{CSP\text{-}SAT}_{\mathsf{Search}(\mathcal{F})}$ and mapping y inputs to 0-inputs of $\mathrm{CSP\text{-}SAT}_{\mathsf{Search}(\mathcal{F})}$, respectively. To finish the argument we need to compute $|U|, A_1(1, U), A_1(r, U), |V|, A_0(s, V)$.

It is easy to see that every variable participates in some clause in $\mathcal{F}$ with high probability. This implies that $\mathcal{U}$ is one-to-one with high probability, and thus $|U| = 2^n$ with high probability.

Recall that the 0-inputs of $\mathrm{CSP\text{-}SAT}_{\mathsf{Search}(\mathcal{F})}$ correspond to substituting y-assignment into $\mathcal{F}$ and writing out truth tables of the all the clauses. The truth tables corresponding to the

clauses that were satisfied by the y-assignment are identically 1, and the truth tables corresponding to the clauses that were not satisfied by the given y-assignment contain exactly one 0-entry. Given a y assignment we call the set of clauses that were not satisfied the assignment the *profile* of y. The next lemma implies that the profiles of all y-assignments are distinct with high probability.

Lemma 4.8. *Let $\mathcal{F} \sim \mathcal{F}(m, n, d)$, and define the following $2^n \times m$ matrix M, with the rows labelled by assignments $\alpha \in \{0, 1\}^n$ and the columns are labelled by clauses of $\mathcal{F}$. Namely, for any pair (α, i) set*

$$M[\alpha, i] = \begin{cases} 1 & \text{if the ith clause is not satisfied by } \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

For any $c > 2/\log e$, if $m \geq c2^d n^2/d$ then the rows of M are distinct with high probability.

Proof. We think of M as generated column by column with the columns sampled independently. Fix two assignments α and $\widehat{\alpha}$ such that $\alpha \neq \widehat{\alpha}$. Let S be the set of indices on which the two assignments differ, i.e., $S = \{i \mid \alpha_i \neq \widehat{\alpha}_i\}$. Set $s = |S|$. Let C_i denote the ith clause, and we say that C_i *overlaps* S if C_i contains a variable in S. Then

$$\Pr[C_i \text{ unsat by } \widehat{\alpha} \text{ and satisfied by } \alpha] = \frac{1}{2^d}\left(1 - \frac{\binom{n-s}{d}}{\binom{n}{d}}\right)$$

$$\geq \frac{1}{2^d}\frac{\binom{n}{d} - \binom{n-1}{d}}{\binom{n}{d}} = \frac{1}{2^d}\frac{\binom{n-1}{d-1}}{\binom{n}{d}} = \frac{d}{2^d n}.$$

Thus the probability that rows α and $\widehat{\alpha}$ agree on column i is at most $1 - \frac{d}{2^d n}$. Since columns are sampled independently, the probability that α and $\widehat{\alpha}$ agree on all columns is at most $\left(1 - \frac{d}{2^d n}\right)^m \leq e^{-dm/(2^d n)}$. By a union bound over ordered pairs of assignments, the probability that there exists a pair of rows that agree on all columns is at most $2^{2n}e^{-dm/(2^d n)} = 2^{2n-(\log e)dm/(2^d n)} \leq 2^{2n-(\log e)cn} = 2^{-\Omega(n)}$. Thus, the probability that all columns are distinct is at least $1 - 2^{-\Omega(n)}$. $\square$

Since each profile is distinct with high probability, this implies that $\mathcal{V}$ is 1-1 with high probability, and therefore $|V| = 2^n$. It remains to bound the terms $A_1(1, U)$, $A_1(r, U)$, and $A_0(s, V)$.

Bounding $A_1(1, U)$. Fixing a single bit of a 1-input in U to $\text{CSP-SAT}_{\mathsf{Search}(\mathcal{F})}$ to 1 is the same as selecting a vertex C in the bipartite constraint graph of $\mathsf{Search}(\mathcal{F})$ and an assignment α to the variables which participate in C, and then setting $\text{TT}_C(\alpha) = 1$. By the definition of $\mathcal{U}$, any input $x \in \{0, 1\}^n$ fixing this bit to 1 determines d out of n variables of x exactly. Thus the number of $x \in \{0, 1\}^n$ that are consistent with this partial assignment is 2^{n-d}, and since $\mathcal{U}$ is one-to-one, we have $A_1(1, U) = 2^{n-d}$.

14

Bounding $A_1(r, U)$. Similar to the previous bound, but now we fix r of the truth table bits to 1. By definition of $\mathcal{U}$, these bits must be chosen from r distinct truth tables in the 1-input in order to be consistent with any $x \in \{0,1\}^n$. With respect to the underlying CNF $\mathcal{F}$, this corresponds to fixing an assignment to the set of variables appearing in an arbitrary set $\mathcal{S}$ of r clauses in $\mathcal{F}$. By Lemma 4.7, with high probability we have $|\mathsf{vars}(S)| \geq ds/2$. Thus fixing these r bits in the definition of $A_1(r, U)$ corresponds to setting at least $rd/2$ of the input variables participate in the constraints with determined truth tables. The number of x inputs that are consistent with these indices fixed is therefore $\leq 2^{n-rd/2}$, and so $A_1(r, U) \leq 2^{n-rd/2}$.

Bounding $A_0(s, V)$. This case is similar to $A_1(r, U)$. We get $A_0(s, V) \leq 2^{n-sd/2}$.

Observe that $(s-1)A_1(1, U) = (s-1)2^{n-d} = (s-1)2^n/n^2 \leq 2^{n-1}$. Putting this altogether we get the following lower bound on monotone circuit size is at least

$$\frac{2^{n-1}}{(s-1)^s 2^{n-sd/2}} = 2^{sd/2 - s\log(s-1)} \geq 2^{s(d/2 - \log s)} \geq 2^{\tilde{\Omega}(n)},$$

where the last inequality follows from $s = n/ed^2$ and $d/4 \geq \log n$. $\qquad\square$

4.2 Uniformly Random CNFs

In this section we show how to modify the argument from the previous section to apply to the "usual" distribution of random CNFs $\mathcal{F}(m, n, d)$. Our approach is simple: using the probabilistic method we find a partition of the variables of a random formula $\mathcal{F} \sim \mathcal{F}(m, n, d)$ such that many of the clauses in $\mathcal{F}$ are balanced with respect to the partition. Ideally, every clause would be so balanced, but it turns out that this is too strong — instead, we show that we can balance many of the clauses, and the imbalanced clauses that remain are always satisfied by a large collection of assignments. First we introduce our notion of "imbalanced" clauses.

Definition 4.9. Fix $\epsilon > 0$. Given a partition of n variables into x-variables and y-variables, clause C is called X-heavy if it contains more than $(1 - \epsilon)d$ x-variables. Clause C is called Y-heavy if it contains more than $(1 - \epsilon)d$ y-variables. Clause C is called balanced if it is neither X-heavy nor Y-heavy.

We recall some basic facts from probability theory which will be used in our main lemma.

Lemma 4.10 (Lovász Local Lemma). *Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be a finite set of events in the probability space Ω. For $E \in \mathcal{E}$ let $\Gamma(E)$ denote the set of events E_i on which E depends. If there is $q \in [0, 1)$ such that $\forall E \in \mathcal{E}$ we have $\Pr(E) \leq q(1 - q)^{|\Gamma(E)|}$, then the probability of avoiding all sets E_i is at least $\Pr(\overline{E_1} \wedge \overline{E_2} \wedge \cdots \wedge \overline{E_n}) \geq (1 - q)^n$.*

Fact 4.11 (Entropy bound on binomial tail). *Given $\epsilon > 0$ we have*

$$\sum_{j=0}^{\lfloor \epsilon n \rfloor} \binom{n}{j} \leq e^{nH(\epsilon)},$$

where $H(\epsilon) = -\epsilon \log \epsilon - (1 - \epsilon) \log(1 - \epsilon)$ is the binary entropy function.

Fact 4.12 (Multiplicative Chernoff Bound). *Suppose $Z_1, \ldots, Z_n$ are independent random variables taking values in $\{0, 1\}$. Let Z denote their sum and let $\mu = \mathbb{E}(Z)$ denote the sum's expected value. Then for any $\delta \in (0, 1)$ we have*

$$\Pr(X \geq (1 + \delta)\mu) \leq e^{-\delta^2 \mu / 3}.$$

We now prove the main lemma of this section, which shows that for $\mathcal{F} \sim \mathcal{F}(m, n, d)$ a good partition of the variables exists with high probability.

Lemma 4.13. *Let $\mathcal{F} \sim \mathcal{F}(m, n, d)$ where $d = c \log n$ and $m = \mathsf{poly}(n)$. There exists a partition of the variables of $\mathcal{F}$ into two sets (X, Y) such that the following holds:*

1. *The number of X-heavy clauses and Y-heavy clauses are each upper bounded by $m' = m2^{-(1-(\log e)H(\epsilon))d+1}$.*

2. *There exists a set U' of $2^{n/2 \pm o(n)}$ truth assignments to the X variables satisfy all X-heavy clauses, and similarly a set V' of $2^{n/2 \pm o(n)}$ truth assignments to the Y-variables satisfying all of the Y-heavy clauses.*

Proof. We prove the existence of such a partition by the probabilistic method. For each variable, flip a fair coin and place it in X if the coin is heads and in Y otherwise. Let Z_i be the random variable indicating whether clause i is X-heavy. Then

$$\Pr(Z_i = 1) = \sum_{j=0}^{\epsilon d} \binom{d}{j} 2^{-d} \leq 2^{-d} e^{-dH(\epsilon)} \leq 2^{-(1-(\log e)H(\epsilon))d},$$

where the inequality follows from Fact 4.11. Let $Z = \sum_{i=1}^{m} Z_i$; then $\mathbb{E}(Z) \leq m2^{-(1-(\log e)H(\epsilon))d} = m'$. By the multiplicative Chernoff bound (see Fact 4.12) we have

$$\Pr(Z > (3/2)m2^{-(1-(\log e)H(\epsilon))d}) \leq e^{-\frac{m2^{-(1-(\log e)H(\epsilon))d}}{12}},$$

and we thus have $Z \leq m'$ with high probability. An identical calculation applies for the Y-heavy clauses.

Next, let W_i be the random variable indicating whether a given fixed variable occurs in clause i and clause i is X-heavy and let $W = \sum_i W_i$. Then $\Pr(W_i = 1) \leq 2^{-(1-(\log e)H(\epsilon))d}d/n$. By the multiplicative Chernoff bound (see Fact 4.12) we have

$$\Pr(W > (3/2)m2^{-(1-(\log e)H(\epsilon))d}d/n) \leq e^{-\frac{m2^{-(1-(\log e)H(\epsilon))d}d/n}{12}}.$$

We conclude that $W \leq m'd/n$ whp, and an identical calculation again holds for the Y-heavy clauses.

Noting that the number of x-variables is $n/2 \pm o(n)$ with high probability, by the probabilistic method we choose a partition (X, Y) which satisfies each of the above properties (the bound on Z and W, and achieving near balance in the X and Y variables), and note that such a partition exists with high probability over $\mathcal{F}(m, n, d)$. With this partition fixed, consider

selecting a random assignment to the X-variables. Let E_i be the event that X-heavy clause i is falsified by the random assignment, and observe that $\Pr(E_i) \leq 2^{-(1-\epsilon)d}$ since the clause is X-heavy. Then the number of events E_i is at most m', and for any event E_i the number of events that share an x-variable with E_i is at most $m'd^2/n$. Set $q = n/(100m'd)$. Then for each E_i we have

$$q(1-q)^{|\Gamma(E_i)|} \geq qe^{-2qm'd^2/n} \geq \frac{n}{100dm'}e^{-d/50} \geq 2^{-(1-\epsilon)d},$$

provided $d \geq c\log n$ for a big enough constant c. Applying Lovász Local Lemma (see Lemma 4.10) we get that probability that an assignment satisfies all X-heavy clauses is at least

$$(1-q)^{m'} \geq (1 - n/(100dm'))^{m'} \geq e^{-n/(50d)}.$$

Thus the number of assignments to the X-variables satisfying all heavy clauses is at least $2^{n/2 \pm o(n)}$, and an identical calculation applies to the Y variables. $\square$

Now we will do the whole argument with respect to $\mathcal{U}(U')$ and $\mathcal{V}(V')$ chosen from the previous lemma. The reason that this works is that since every $\alpha \in U'$ satisfies all X-heavy clauses, if we look at a subset S of the variables of the monotone CSP that are set to false and count the number of maxterms that are consistent with it, the count is nonzero only when *none* of these variables come from a X-heavy clause. Similarly if we look at a subset S of the variables of the monotone CSP that are set to true and count the number of maxterms that are consistent with it, this count is nonzero only when *none* of these variables come from an X-heavy clause. Therefore, when we calculate $A_1(s, \mathcal{U}(U'))$ and $A_0(s, \mathcal{V}(V'))$, the calculation is with respect to the X-balanced clauses, and Y-balanced clauses, respectively. Thus we can use the same expansion calculation that we already did.

There is also a minor modification required in order to argue that $\mathcal{V}$ is one-to-one when restricted to V'. As before, it suffices to show that for any two assignments α, β in V', that the probability that they agree on all of the balanced clauses is very small, and then take a union bound over all of the *balanced* clauses. This calculation is nearly identical to the one that we already did, but now the union bound is over the number of balanced clauses, which is at least half of all clauses, so the calculation is essentially the same.

With the above modifications, the arguments from the previous section imply the next theorem.

Theorem 4.14. *Let n be a sufficiently large positive integer. Let $\mathcal{F} \sim \mathcal{F}(m, n, d)$ for $m = \mathsf{poly}(n)$ and $d = c\log n$ for a large universal constant c. With high probability, there exists a partition (X, Y) of the variables of $\mathcal{F}$ and an $\varepsilon > 0$ such that the search problem $\mathsf{Search}(\mathcal{F})$ defined with respect to this partition satisfies the following: any real monotone circuit computing $\mathrm{CSP\text{-}SAT}_{\mathsf{Search}(\mathcal{F})}$ requires at least $2^{\Omega(n^\varepsilon)}$ gates.*

Corollary 4.15. *Let $\mathcal{F}$ be distributed as above. There exists $\varepsilon > 0$ such that with high probability any RCC_1-refutation requires $2^{\Omega(n^\varepsilon)}$ lines.*

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, Baltimore, MD, USA, May 22-24, 2005*, pages 251–256, 2005.

[2] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

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

[4] Boaz Barak, Guy Kindler, and David Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 197–214, 2013.

[5] Paul Beame, Richard M. Karp, Toniann Pitassi, and Michael E. Saks. On the complexity of unsatisfiability proofs for random k-cnf formulas. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 561–571, 1998.

[6] Eli Ben-Sasson and Russell Impagliazzo. Random cnf's are hard for the polynomial calculus. *Computational Complexity*, 19(4):501–519, 2010.

[7] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[8] Christer Berg and Staffan Ulfberg. Symmetric approximation arguments for monotone lower bounds without sunflowers. *Computational Complexity*, 8(1):1–20, 1999.

[9] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *J. Symb. Log.*, 62(3):708–728, 1997.

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

[11] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 59–68, 2015.

[12] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 534–543, 2002.

[13] Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 35:1–35:13, 2016.

[14] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 847–856, 2014.

[15] Armin Haken and Stephen A. Cook. An exponential lower bound for the size of monotone real circuits. *J. Comput. Syst. Sci.*, 58(2):326–335, 1999.

[16] Pavel Hrubes and Pavel Pudlák. A note on monotone real circuits. *Unpublished*, 2016.

[17] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[18] Jan Krajícek. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997.

[19] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

[20] Alexander Razborov. Lower bounds for the monotone complexity of some boolean functions. *Sov. Math. Dokl.*, 31:354–357, 1985.

[21] Alexander Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya Mathematics*, 59(1):205–227, 1995.

[22] Dmitry Sokolov. Dag-like communication and its applications. *ECCC TR16-202*, 2017.

[23] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012.