Counting Solutions to Random CNF Formulas

Andreas Galanis
Department of Computer Science, University of Oxford, UK
andreas.galanis@cs.ox.ac.uk

Leslie Ann Goldberg
Department of Computer Science, University of Oxford, UK
leslie.goldberg@cs.ox.ac.uk

Heng Guo
School of informatics, University of Edinburgh, UK
hguo@inf.ed.ac.uk

Kuan Yang
Department of Computer Science, University of Oxford, UK
kuan.yang@cs.ox.ac.uk

Abstract

We give the first efficient algorithm to approximately count the number of solutions in the random k-SAT model when the density of the formula scales exponentially with k. The best previous counting algorithm was due to Montanari and Shah and was based on the correlation decay method, which works up to densities $(1+o_k(1))\frac{2\log k}{k}$, the Gibbs uniqueness threshold for the model. Instead, our algorithm harnesses a recent technique by Moitra to work for random formulas with much higher densities. The main challenge in our setting is to account for the presence of high-degree variables whose marginal distributions are hard to control and which cause significant correlations within the formula.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Randomness, geometry and discrete structures; Mathematics of computing → Discrete mathematics

Keywords and phrases random CNF formulas, approximate counting

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.53

Category Track A: Algorithms, Complexity and Games

Related Version A full version with all proofs is at `https://arxiv.org/abs/1911.07020`. The theorem numbers here match those in the full version. Proof sketches here refer to lemmas in the full version without restating them in detail.

Funding The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 334828. The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

1 Introduction

Let $\Phi = \Phi(k, n, m)$ be a k-CNF formula on n Boolean variables with m clauses chosen uniformly at random where each clause has size $k \geq 3$. The random formula Φ shows an interesting threshold behaviour, where the asymptotic probability that Φ is satisfiable drops dramatically from 1 to 0 when the density $\alpha := m/n$ crosses a certain threshold $\alpha_\star$. There has been tremendous progress on establishing this phase transition and pinpointing the

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 53; pp. 53:1–53:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

threshold $\alpha_\star$ [25, 19, 3, 4, 12, 15] guided by elaborate but non-rigorous methods in physics [28, 27]. The exact value of the threshold $\alpha_\star$ is established in [15] for sufficiently large k; it is known that $\alpha_\star = 2^k \ln 2 - \frac{1}{2}(1 + \ln 2) + o_k(1)$ as $k \to \infty$.

In contrast, the "average case" computational complexity of random k-CNF formulas remains elusive. It is a notoriously hard problem to design algorithms that succeed in finding a satisfying assignment when the density of the formula Φ is close to (but smaller than) the satisfiability threshold $\alpha_\star$. The best polynomial-time algorithm to find a satisfying assignment of Φ is due to Coja-Oghlan [8], which succeeds if $\alpha < (1 - o_k(1)) \cdot 2^k \ln k/k$. It is known that beyond this density bound $2^k \ln k/k$ the solution space of the formula undergoes a phase transition and becomes severely more complicated [2], so local algorithms are bound to fail to find a satisfying assignment in polynomial time (see for example [24, 9, 11]).

It is also a natural question to determine the number of satisfying assignments to Φ, denoted by $Z(\Phi)$, when the density is below the satisfying threshold. It has been shown that $\frac{1}{n} \log Z(\Phi)$ is concentrated around its expectation [1, 13] for $\alpha < (1 - o_k(1)) \cdot 2^k \ln k/k$. However, for the k-SAT model, there is no known formula for the expectation $\mathbb{E}\,\frac{1}{n} \log Z(\Phi)$ (though see [35, 14] for progress along these lines for more symmetric models of random formulas). Regarding the algorithmic question, Montanari and Shah [31] have given an efficient algorithm to approximate $\log Z(\Phi)$ if $\alpha \le \frac{2 \log k}{k}(1 + o_k(1))$, based on the correlation decay method and the uniqueness threshold of the Gibbs distribution. Note that this only gives an approximation to $Z(\Phi)$ within an exponential factor. Also, the threshold for α is exponentially lower than the satisfiability threshold. No efficient algorithm was known to give a more precise approximation.

In this paper, we address the algorithmic counting problem by giving the first *fully polynomial-time approximation scheme* (FPTAS) for the number of satisfying assignments to random k-CNF formulas, if the density α is less than 2^{rk}, for sufficiently large k and some constant $r > 0$. Our bound is exponential in k and goes well beyond the uniqueness threshold of $\frac{2 \log k}{k}(1 + o_k(1))$ which is required by the correlation decay method.

Our result is related to other algorithmic counting results on random graphs such as counting colourings, independent sets, and other structures [33, 37, 16, 26] in random graphs. However, previous methods, such as Markov Chain Monte Carlo and Barvinok's method, appear to be difficult to apply to random formulas. Instead, our algorithm is the first adaptation of Moitra's method [30] to the random instance setting. We give a high level overview of the techniques in Section 1.2.

1.1 The model and the main result

For $k \ge 3$, let $\Phi = \Phi(k, n, m)$ denote a k-SAT formula chosen uniformly at random from the set of all k-SAT formulas with n variables and m clauses. Specifically, Φ has n variables $v_1, v_2, \ldots, v_n$ and m clauses $c_1, c_2, \ldots, c_m$. Each clause c_i has k literals $\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,k}$ and each literal $\ell_{i,j}$ is chosen uniformly at random from $2n$ literals $\{v_1, v_2, \ldots, v_n, \neg v_1, \neg v_2, \ldots, \neg v_n\}$. Note that each clause has exactly k literals (repetitions allowed), so there are $(2n)^{km}$ possible formulas; we use $\Pr(\cdot)$ to denote the uniform distribution on the set of all such formulas. Throughout, we will assume that $m = \lfloor n\alpha \rfloor$, where $\alpha > 0$ is the density of the formula. We say that an event $\mathcal{E}$ holds *w.h.p.* if $\Pr(\mathcal{E}) = 1 - o(1)$ as $n \to \infty$.

For a k-SAT formula Φ, we let $\Omega = \Omega(\Phi)$ denote the set of satisfying assignments of Φ.

▶ **Theorem 1.** *There is a polynomial-time algorithm $\mathcal{A}$ and there are two constants $r > 0$ and $k_0 \ge 3$ such that, for all $k \ge k_0$ and all $\alpha < 2^{rk}$, the following holds w.h.p. over the choice of the random k-SAT formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. The algorithm $\mathcal{A}$, given as input the formula Φ and a rational $\varepsilon > 0$, outputs in time poly$(n, 1/\varepsilon)$ a number Z that satisfies $e^{-\varepsilon}|\Omega(\Phi)| \le Z \le e^{\varepsilon}|\Omega(\Phi)|$.*

Throughout this paper, we will assume that $k \geq k_0$ where k_0 is a sufficiently large constant. We will also assume that the density α of the formula Φ satisfies $\alpha < 2^{k/300}/k^3$, so r can be taken to be $1/301$ in Theorem 1. The constant 300 here is not optimised, but we do not expect to be able to use the current techniques to improve it substantially. Our main point is that for a density which is exponential in k, an FPTAS exists for random k-CNF formulas. Finally, we assume that $k^2\alpha \geq 1$, otherwise it is well-known (see, e.g., Theorem 3.6 in [34]) that w.h.p. every connected component of Φ, viewed as a hypergraph where variables correspond to vertices and clauses correspond to hyperedges, is of size $O(\log n)$. In this case we can count the number of satisfying assignments by brute force.

1.2 Algorithm overview

We give a high-level overview of our algorithm here before giving the details. Approximately counting the satisfying assignments of a k-CNF formula has been a challenging problem using traditional algorithmic techniques, since the solution space (the set of satisfying assignments) is complicated and it is not connected, using the transitions of commonly-studied Markov chains. Recently some new approaches were introduced [30, 20]. Most notably, the breakthrough work of Moitra [30] gives the first (and so far the only) efficient deterministic algorithm that can approximately count the satisfying assignments of k-CNF formulas in which each variable appears in at most d clauses, if, roughly, $d \lesssim 2^{k/60}$. Inspired by this, Feng et al. [18] have also given a MCMC algorithm which applies when $d \lesssim 2^{k/20}$.

As our goal is to count satisfying assignments of sparse random k-CNF formulas, where these degree bounds do not hold, but average degrees are small, it is natural to also choose Moitra's method in the random instance setting. However, the first difficulty is that Moitra's method relies on the fact that the marginal probability of each variable (the probability that the variable is true in a uniformly-chosen satisfying assignment) is nearly 1/2. This is necessary because Moitra's method involves solving a certain linear program (LP) and the size of this LP is polynomially-bounded only if a certain process couples quickly. The proof that the process couples quickly relies on the fact that the marginals are nearly 1/2 (and certainly on the fact that they are bounded away from 0 and 1). In contrast, for a random k-CNF formula, although the *average* degree of variables is low, w.h.p. there are variables with degrees as high as $\Omega\left(\log n/\log\log n\right)$. In the presence of these high-degree variables, the marginal probabilities of the variables can be arbitrarily near 0 or 1, instead of 1/2.

Our solution to this issue is to separate out high-degree variables, as well as those that are heavily influenced by high-degree variables. To do this, we define a process to recursively label "bad" variables. At the start, all high-degree variables are bad. Then, all clauses containing more than $k/10$ bad variables are labelled bad, as are all variables that they contain. We run this process until no more bad clauses are found. We call the remaining variables and clauses of the formula "good". A key property is that all good variables have an upper bound on their degree and all good clauses contain at least $9k/10$ good variables; this allows us to show that the marginal probabilities of good variables are close to 1/2.

The next step is to attempt to apply Moitra's method. The goal of Moitra's method is to compute more precise estimates for the marginal probabilities of the variables; given accurate estimates on the marginal probabilities, it is then relatively easy to approximate the number of satisfying assignments using refined self-reducibility techniques.

Of course, we need to modify the method to deal with the bad variables, which still appear in the formula. We first explain Moitra's method and then proceed with our modifications. The first step is to mark variables, so that every clause contains a good fraction of marked variables and a good fraction of unmarked variables. Then, for a particular marked variable

v, we set up an LP. As noted earlier, the variables of the LP correspond to the states of a certain coupling process which couples two distributions on satisfying assignments using the marked variables – the first distribution over satisfying assignments in which v is true, and the second distribution over satisfying assignments in which v is false. Solving the LP recovers the transition probabilities of the coupling process and yields enough information to approximate the marginal probability of v.

In order to guarantee that the size of the LP is bounded by a polynomial in the size of the original CNF formula, we have to restrict the coupling process. The process can be viewed as a tree and it suffices to truncate this tree at a suitable level.

Thus, a crucial part of the proof (both in Moitra's case and in ours) is to show that the error caused by the truncation is sufficiently small. The reason that the error caused by the truncation is small is that, with high probability, branches of the coupling tree "die out" before reaching a large level. The reason for this is that the marginals of marked variables stay near $1/2$, even when conditioning on partial assignments.

In our case where Φ is a random formula, the marginals are not all near $1/2$, even without any conditioning. But the good variables do have marginals near $1/2$. So we only mark/unmark good variables and we "give up" on bad variables. Given that we don't have any control over the bad variables, we have to modify the coupling process. Thus, whenever we meet a bad variable in the coupling process, we have to assume the worst case and treat this variable and all bad variables connected to it as if they all have failed the coupling, meaning that the disagreement spreads quickly over bad components.

The most important part of our analysis is to upper bound the size of connected bad components and how often we encounter them during the coupling processs. Given these upper bounds, we are able to show that the coupling still dies out sufficiently quickly, so the error caused by the truncation is not too large. Solving the LP then allows us to estimate the marginals of the good variables. Given that the bad components have small size, this turns out to be enough information to estimate the number of satisfying assignments of the original formula (containing both good and bad clauses).

We conclude this summary by discussing the prospects for improving our work. Although we have given an efficient algorithm which works for densities that are exponentially large in k, the densities that we can handle are still small compared to the satisfiability threshold or to the threshold under which efficient search algorithms exist. Perhaps a modest start towards obtaining comparable thresholds for approximate counting algorithms would be to consider models whose state spaces are connected. For example, for monotone k-CNF formulas where each variable appears in at most d clauses, Hermon et al. [23] showed that efficient randomised algorithms exist if $d \leq c2^{k/2}$ for some constant $c > 0$, which is optimal up to the constant c due to complementing hardness results [6]. They also showed that the same algorithm works for *random regular* monotone k-CNF formulas, if the degree $d \leq c2^k/k$ for some $c > 0$. It remains open whether an average case bound of the same order can be achieved for random monotone k-CNF formulas.

2 The coupling tree

2.1 Identifying bad variables

We start by identifying bad variables; the method that we use is inspired by [12].

► **Definition 2.** *Let Φ be a k-SAT formula. We say that a variable v of Φ is* high-degree *if Φ contains at least $\Delta := 2^{k/300}$ occurrences of literals involving the variable v.*

The reason that high-degree variables are harmful is that their marginal probabilities (when we sample uniformly from satisfying assignments) are not bounded away from 0 and 1. Also, any variable that shares clauses with high-degree variables may also have biased marginals. In our algorithm, we will not be able to control these high degree variables or other variables that are affected by them. This variables contribute to the "bad" part of the formula Φ. Formally, denote the set of clauses of Φ by $\mathcal{C}$ and the set of variables by $\mathcal{V}$. For each $c \in \mathcal{C}$, let $\mathsf{var}(c)$ denote the set of variables in c. For each subset C of $\mathcal{C}$, let $\mathsf{var}(C) := \cup_{c \in C} \mathsf{var}(c)$. The *bad variables* and *bad clauses* of Φ are identified as follows:

1. $\mathcal{V}_0$ (the initial bad variables) $\leftarrow$ the set of high-degree variables;
2. $\mathcal{C}_0 \leftarrow$ the set of clauses with at least $k/10$ variables in $\mathcal{V}_0$;
3. $i \leftarrow 0$;
4. Do the following until $\mathcal{V}_i = \mathcal{V}_{i-1}$:
   - $i \leftarrow i + 1$;
   - $\mathcal{V}_i \leftarrow \mathcal{V}_{i-1} \cup \mathsf{var}(\mathcal{C}_{i-1})$;
   - $\mathcal{C}_i \leftarrow \{c \in \mathcal{C} \mid \mathsf{var}(c) \cap \mathcal{V}_i \geq k/10\}$;
5. $\mathcal{C}_{\text{bad}} \leftarrow \mathcal{C}_i$ and $\mathcal{V}_{\text{bad}} \leftarrow \mathcal{V}_i$;
6. $\mathcal{C}_{\text{good}} \leftarrow \mathcal{C} \setminus \mathcal{C}_i$ and $\mathcal{V}_{\text{good}} \leftarrow \mathcal{V} \setminus \mathcal{V}_i$.

▶ **Observation 3.** *$\forall c \in \mathcal{C}_{good}$, $|\mathsf{var}(c) \cap \mathcal{V}_{bad}| < k/10$. $\forall c \in \mathcal{C}_{bad}$, $|\mathsf{var}(c) \cap \mathcal{V}_{good}| = 0$.*

2.2 Marking good variables and identifying a satisfying assignment

Apart from the fact that we only mark variables in $\mathcal{V}_{\text{good}}$, our marking follows the approach of Moitra [30]. Formally, a "marking" is an assignment from $\mathcal{V}_{\text{good}}$ to $\{\text{marked}, \text{unmarked}\}$. Using Observation 3 and applying the asymmetric version of the Lovász local lemma [17, 36, 22] and the algorithmic version of the local lemma by Moser and Tardos [32] it is easy to prove the following lemma.

▶ **Lemma 8.** *There exists a marking on $\mathcal{V}_{good}$ such that every good clause has at least $3k/10$ marked variables and at least $k/4$ unmarked good variables. It has the property that there is a partial assignment of bad variables that satisfies all bad clauses. Furthermore, such a marking can be found in deterministic polynomial time.*

We also use the Lovász local lemma to identify a partial assignment Λ^* that we will use to apply self-reducibility.

▶ **Lemma 10.** *Let $\Phi = \Phi(k, n, m)$ and let $v_1, v_2, \ldots, v_n$ be the variables of Φ. In each clause, order the literals in the order induced by the indices of their variables. Then there is a partial assignment Λ^* of truth values to some subset of $\mathcal{V}_{\mathsf{marked}}$ with the property that every clause $c \in \mathcal{C}_{good}$ is satisfied by its first $k/20$ literals corresponding to marked variables. Moreover, Λ^* can be found in deterministic polynomial time.*

2.3 The coupling tree

Fix a prefix Λ of the assignment Λ^* from Lemma 10. Let Φ^Λ be the formula produced by simplifying Φ under Λ (remove clauses that are satisfied under Λ and remove all false literals). $\mathcal{C}^\Lambda$ denotes the clauses of Φ^Λ and $\mathcal{V}^\Lambda$ denotes the variables. We also define $\mathcal{V}^\Lambda_{\text{good}} = \mathcal{V}_{\text{good}} \cap \mathcal{V}^\Lambda$ and $\mathcal{C}^\Lambda_{\text{good}} = \mathcal{C}_{\text{good}} \cap \mathcal{C}^\Lambda$. Ω^Λ denotes the set of satisfying assignments of Φ^Λ.

For a variable $v^* \in \mathcal{V}^\Lambda$, let Ω^Λ_1 be the set of assignments in Ω^Λ in which v^* is true, and let Ω^Λ_2 be the set of assignments in Ω^Λ in which v^* is false. The algorithm estimates the marginal probability that v^* is true by solving a certain LP which allows it to estimate the ratio $|\Omega_1^\Lambda|/|\Omega_2^\Lambda|$. The variables of the LP correspond to the states of a coupling process. The process couples the uniform distribution on Ω_1^Λ with the uniform distribution on Ω_2^Λ. We can now describe process via its "coupling tree" $\mathbb{T}^\Lambda$.

For each node ρ there is a partial assignment $\mathcal{A}_1(\rho) \in \Omega_1^\Lambda$ and a partial assignment $\mathcal{A}_2(\rho) \in \Omega_2^\Lambda$. The variables set in these partial assignments are $\Lambda \cup V_{\mathrm{set}}(\rho)$. The set $V_I(\rho)$ contains "problematic" variables. The details will be clear later. Roughly, these include variables in $V_{\mathrm{set}}(\rho)$ on which $\mathcal{A}_1(\rho)$ and $\mathcal{A}_2(\rho)$ disagree, variables contained in clauses that are not satisfied in some $\mathcal{A}_i(\rho)$, even though all marked variables have already been set, and variables "affected" by bad variables during the coupling process. $\mathcal{C}_{\mathrm{rem}}(\rho)$ is the set of remaining clauses to consider at descendants of ρ in the coupling.

The root of the coupling tree is the node ρ^* with $V_{\mathrm{set}}(\rho^*) = V_I(\rho^*) = \{v^*\}$. The assignment $\mathcal{A}_1(\rho^*)$ sets v^* to T and the assignment $\mathcal{A}_2(\rho^*)$ sets v^* to F. $\mathcal{C}_{\mathrm{rem}}(\rho^*) = \mathcal{C}^\Lambda$. Let $n = |\mathcal{V}|$. In order to ensure that the size of the LP is bounded by a polynomial in n we need to ensure that the size of the coupling tree is also bounded by a polynomial in n. To do this, we choose truncation depth $L := C_0(3k^2\Delta)\lceil\log(n/\varepsilon)\rceil$ where C_0 is a sufficiently large constant. We then truncate the tree as follows.

▶ **Definition 12.** *A node ρ of the coupling tree is* a leaf *if $|V_I(\rho)| \le L$ and every $c \in \mathcal{C}_{rem}(\rho)$ has the property that $\mathsf{var}(c) \subseteq V_I(\rho) \cup V_{set}(\rho)$ or $\mathsf{var}(c) \subseteq \mathcal{V}^\Lambda \setminus (V_I(\rho) \cup V_{set}(\rho))$. If $|V_I(\rho)| > L$, then ρ is a* truncating node. *We denote the set of leaves by $\mathcal{L}$, the set of truncating nodes by $\mathcal{T}$, and their union by $\mathcal{L}^* := \mathcal{L} \cup \mathcal{T}$.*

If ρ is not in $\mathcal{L}^*$ then we define its four children as follows. The "first clause" of ρ is the first good clause c with a variable in $V_I(\rho)$ and a variable in $\mathcal{V}^\Lambda \setminus V_I(\rho)$. (The definitions imply that such a clause exists.) The "first variable" u of ρ is the first (good) variable in $\mathsf{marked}(c) \setminus V_{\mathrm{set}}(\rho)$. For each of the four pairs (τ_1, τ_2) where τ_1 and τ_2 are assignments from $\{u\}$ to $\{\mathsf{T}, \mathsf{F}\}$, we create a child ρ_{τ_1,τ_2} of ρ using the following algorithm.

Algorithm 1 Constructing the child ρ_{τ_1,τ_2} of a non-truncating node ρ of the coupling tree, where τ_1, τ_2 are assignments from $\{u\}$ to $\{\mathsf{T}, \mathsf{F}\}$, and u is the first variable of ρ.

```
$V_{\mathrm{set}}(\rho_{\tau_1,\tau_2}) \leftarrow V_{\mathrm{set}}(\rho) \cup \{u\}$;
$\mathcal{A}_1(\rho_{\tau_1,\tau_2}) \leftarrow$ combine $\mathcal{A}_1(\rho)$ with $\tau_1$;
$\mathcal{A}_2(\rho_{\tau_1,\tau_2}) \leftarrow$ combine $\mathcal{A}_2(\rho)$ with $\tau_2$;
$(V_I, \mathcal{C}_{\mathrm{rem}}) \leftarrow (V_I(\rho), \mathcal{C}_{\mathrm{rem}}(\rho))$;
if $\tau_1(u) \neq \tau_2(u)$ then
    $V_I \leftarrow V_I \cup \{u\}$;
end if
for $c' \in \mathcal{C}_{\mathrm{rem}}$ s.t. $c'$ is satisfied by both $\mathcal{A}_1(\rho_{\tau_1,\tau_2})$ and $\mathcal{A}_2(\rho_{\tau_1,\tau_2})$ do
    $\mathcal{C}_{\mathrm{rem}} \leftarrow \mathcal{C}_{\mathrm{rem}} \setminus \{c'\}$;
end for
while $\exists c' \in \mathcal{C}_{\mathrm{rem}}$ with $\mathsf{var}(c') \cap V_I \neq \emptyset$, $\mathsf{var}(c') \cap (\mathcal{V}^\Lambda \setminus V_I) \neq \emptyset$, and $\mathsf{marked}(c') \setminus$
  $V_{\mathrm{set}}(\rho_{\tau_1,\tau_2}) = \emptyset$ do
    $V_I \leftarrow V_I \cup (\mathsf{var}(c') \setminus V_{\mathrm{set}}(\rho_{\tau_1,\tau_2}))$;
    $\mathcal{C}_{\mathrm{rem}} \leftarrow \mathcal{C}_{\mathrm{rem}} \setminus \{c'\}$;
end while
while $\exists c' \in \mathcal{C}_{\mathrm{rem}} \cap \mathcal{C}_{\mathrm{bad}}$ with $\mathsf{var}(c') \cap V_I \neq \emptyset$ do
    $V_I \leftarrow V_I \cup (\mathsf{var}(c') \setminus V_{\mathrm{set}}(\rho_{\tau_1,\tau_2}))$;
    $\mathcal{C}_{\mathrm{rem}} \leftarrow \mathcal{C}_{\mathrm{rem}} \setminus \{c'\}$;
end while
$(V_I(\rho_{\tau_1,\tau_2}), \mathcal{C}_{\mathrm{rem}}(\rho_{\tau_1,\tau_2})) \leftarrow (V_I, \mathcal{C}_{\mathrm{rem}})$;
```

2.4 Key property of the coupling tree for a random formula

Recall that the variables of the LP which is used to estimate the marginal of the variable v^* of Φ^Λ correspond to the states of the coupling on the coupling tree $\mathbb{T}^\Lambda$. We will define two LP variables $P_{1,\rho}$ and $P_{2,\rho}$ for each node ρ of $\mathbb{T}^\Lambda$. In order to efficiently solve the LP, we need its size to be bounded by a polynomial in n, so we need the number of nodes of $\mathbb{T}^\Lambda$ to be bounded by a polynomial in n. For a random formula, this follows from the following key lemma, which is a main technical contribution of our work.

▶ **Lemma 14.** *W.h.p. over the choice of Φ, for every prefix Λ of Λ^*, every node ρ in $\mathbb{T}^\Lambda$ has the property that $|V_{set}(\rho)| \le 3k^3\alpha L + 1$.*

To see that Lemma 14 implies that the size of the coupling tree is at most a polynomial in n, note that the depth of the tree does not exceed $\max_{\rho\in\mathbb{T}^\Lambda} |V_{\mathrm{set}}(\rho)| \le 3k^3\alpha L + 1 = O(\log \frac{n}{\varepsilon})$. Also, each node has at most 4 children.

In the rest of this section, we sketch the proof of Lemma 14. We start by defining some graphs associated with Φ. The formula Φ naturally corresponds to a bipartite "factor graph" where one side is variables and the other clauses (a variable has an edge to a clause in the factor graph if one its literals is contained in the clause). We also use the following two graphs.

▶ **Definition 3.** *Let G_Φ be the graph with vertex set $\mathcal{C}$ (the clauses of Φ) in which two clauses c and c' are adjacent if and only if $\mathsf{var}(c) \cap \mathsf{var}(c') \neq \emptyset$. We say that a set $C \subseteq \mathcal{C}$ of clauses is connected if the induced subgraph $G_\Phi[C]$ is connected.*

▶ **Definition 4.** *Let H_Φ be the graph with vertex set $\mathcal{V}$ (the variables of Φ) in which two variables v and v' are adjacent if and only if there exists a clause $c \in \mathcal{C}$ such that $v, v' \in \mathsf{var}(c)$. We say that a set $V \subseteq \mathcal{V}$ of variables is connected if the induced subgraph $H_\Phi[V]$ is connected. Let $H_{\Phi,bad}$ be the graph with vertex set $\mathcal{V}_{bad}$ in which two variables v and v' are adjacent if and only if there exists a bad clause $c \in \mathcal{C}_{bad}$ such that $v, v' \in \mathsf{var}(c)$. We say that a set $V \subseteq \mathcal{V}$ of variables is a* bad component *if V is a connected component in $H_{\Phi,bad}$.*

For $V \subseteq \mathcal{V}$, let $\Gamma_{H_\Phi}(V) = \cup_{v\in V}\Gamma_{H_\Phi}(v)$ be the neighbourhood of V in H_Φ. Let $\Gamma^+_{H_\Phi}(V) = V \cup \Gamma_{H_\Phi}(V)$ be the extended neighbourhood. The proof of Lemma 14 relies on the following rather abstract fact about random formulas.[1]

▶ **Lemma 41.** *W.h.p. over the choice of Φ, there do not exist sets Y', Z of clauses and a set V of variables such that $|Y'| \ge \log n$, $|V| \ge |Y'|$, $|Z| \ge 2k^2\alpha\,|V|$, $Y' \cap Z = \emptyset$, $G_\Phi[Y']$ is connected, $V \subseteq \mathsf{var}(Y')$, and every clause in Z contains at least one variable from V.*

The lemma says that if you take any "large" set of clauses Y' that are connected in G_Φ and any large set V of the variables of Y' then there aren't many clauses outside of Y' that contain variables in V. (There isn't a large set Z of such clauses.) Obviously, the lemma doesn't apply to every Φ, but is highly dependent on the random way in which Φ is chosen. The proof of Lemma 41 relies crucially on upper-bounding the probability that a set of clauses Y' is connected in G_Φ. To do this, we sum over possible trees connecting the clauses in Y'. We use the bound from Lemma 39 of the full version, which shows that the probability that any particular tree T is connected in G_Φ is at most $(k^2/n)^{|V(T)|-1}$.

[1] We need a more general version in the full paper, but this suffices here. The variable names here are chosen to make the (single) application in this short version easy.

Proof of Lemma 14. Let Λ be a prefix of Λ^* and let ρ be a node in $\mathbb{T}^\Lambda$. Our goal is to prove $|V_{\text{set}}(\rho)| \le 3k^3\alpha L + 1$. We first consider the case in which ρ is not a truncating node, so $|V_I(\rho)| \le L$ and we show $|V_{\text{set}}(\rho)| \le 3k^3\alpha L$. The proof has two parts.

Part 1. $V_{\text{set}}(\rho) \subseteq \Gamma^+_{H_\Phi}(V_I(\rho))$.

To prove Part 1, we consider any $u \in V_{\text{set}}(\rho) \setminus V_I(\rho)$ and show that there is a clause c containing u and containing a variable in $V_I(\rho)$.

We first rule out the case that $u = v^*$ by noting (from the construction of the coupling tree) that $v^* \in V_I(\rho) \cap V_{\text{set}}(\rho)$.

So consider $u \in V_{\text{set}}(\rho) \setminus V_I(\rho)$ and let ρ' be the ancestor of ρ in the coupling tree such that u is the first variable of ρ'. The definition of the coupling tree guarantees that ρ' is uniquely defined and that it is a proper ancestor of ρ – the definition of "first variable" guarantees that $u \notin V_{\text{set}}(\rho')$, but for all proper descendants ρ''' of ρ', $u \in V_{\text{set}}(\rho''')$.

Let ρ'' be the child of ρ' on the path to ρ. We will show that there is a clause c containing u and containing a variable in $V_I(\rho')$. Part 1 then follows from the fact that $V_I(\rho)$ contains $V_I(\rho')$. The existence of such a clause c is immediate from the definition of "first variable" – indeed c is the "first clause" of ρ'.

Part 2. W.h.p., the random formula Φ is such that $\forall \rho, |\Gamma^+_{H_\Phi}(V_I(\rho))| \le 3k^3\alpha L$.

For Part 2, it is important that the set $V_I(\rho)$ is connected in H_Φ – this follows from the construction of the coupling tree. We show (this is Lemma 51) that, w.h.p. over the choice of Φ, *every* connected set of variables $V \subseteq \mathcal{V}$ satisfies

$$|\Gamma^+_{H_\Phi}(V)| \le 3k^3\alpha \max\{|V|, k\log n\}, \tag{1}$$

which establishes Part 2 since $|V_I(\rho)| \le L$.

The proof of (1) is as follows. Let V be a connected of variables and let Y be the set of neighbours of V in the factor graph of Φ, i.e., $Y = \{c \in \mathcal{C} \mid \mathsf{var}(c) \cap V \neq \emptyset\}$. Clearly $|\Gamma^+_{H_\Phi}(V)| \le k|Y|$ and hence it suffices to show that $|Y| \le 3k^2\alpha \max\{|V|, k\log n\}$. There are two cases depending on the size of V.

- $|V| \ge k\log n$. Since V is a connected set of variables, there exists a set $Y' \subseteq Y$ such that $|V|/k \le |Y'| \le |V|$ and $V \cup Y'$ is connected in the factor graph of Φ. Hence, Y' is a connected set of clauses and $|Y'| \ge \log n$. Let $Z = Y \setminus Y'$. If $|Z| \ge 2k^2\alpha|V|$ then we obtain a contradiction to Lemma 41, which holds w.h.p. Thus, w.h.p., $|Z| \le 2k^2\alpha|V|$ which implies $|Y| = |Y'| + |Z| \le 3k^2\alpha|V|$, as required.
- Otherwise $|V| < k\log n$. If $|\Gamma^+_{H_\Phi}(V)| < \lceil k\log n\rceil$ then we are finished. Otherwise, consider an arbitrary connected $V' \supset V$ such that $|V'| = \lceil k\log n\rceil$. By the argument of the previous case, the set of neighbours of V' in the factor graph, denoted Y'', satisfies that $|Y''| \le 3k^2\alpha|V'| \le 3k^3\alpha\log n$. Thus, $|Y| \le |Y''| \le 3k^3\alpha\log n$.

This completes the proof of (1), and hence Part 2.

To finish, we consider the case where ρ is a truncating node. Let ρ' be the parent of ρ. Parts 1 and 2 imply that $|V_{\text{set}}(\rho')| \le 3k^3\alpha L$. The result follows since $|V_{\text{set}}(\rho)| = |V_{\text{set}}(\rho')| + 1$. ◀

3 The linear program

Here we briefly list the constraints in the LP so that we can discuss its analysis. For a node ρ of the coupling tree, let $\mathcal{C}_I(\rho)$ be the set of clauses $c \in \mathcal{C}^\Lambda$ such that $\mathsf{var}(c) \subseteq V_I(\rho) \cup V_{\text{set}}(\rho)$. For $i \in \{1,2\}$, let $N_i(\rho)$ be the number of assignments τ to $V_I(\rho) \setminus V_{\text{set}}(\rho)$ such that every clause in $\mathcal{C}_I(\rho)$ is satisfied by $\tau \cup \mathcal{A}_i(\rho)$. It turns out (see Lemma 15) that $N_i(\rho) \neq 0$ for $i \in \{1,2\}$, so we define $r(\rho) = N_1(\rho)/N_2(\rho)$.

The LP relies on two constants r_{lower} and r_{upper}. The algorithm that uses the LP will move these closer and closer together by binary search. For each node ρ of the coupling tree, we introduce two variables $P_{1,\rho}$ and $P_{2,\rho}$. The constraints are as follows. **Constraint Set 0**: For every node ρ of the coupling tree and every $i \in \{1,2\}$ we add the constraint $0 \le P_{i,\rho} \le 1$. **Constraint Set 1**: If $\rho \in \mathcal{L}$ then we add the constraint $r_{\mathsf{lower}}\, P_{2,\rho} \le P_{1,\rho}\, r(\rho)$ and the constraint $P_{1,\rho}\, r(\rho) \le r_{\mathsf{upper}}\, P_{2,\rho}$. **Constraint Set 2**: For the root ρ^* of the coupling tree, we add the constraints $P_{1,\rho^*} = 1$ and $P_{2,\rho^*} = 1$. For every node ρ of the coupling tree that is not in $\mathcal{L}^*$, let u be the first variable of ρ. For each $X \in \{\mathsf{T},\mathsf{F}\}$ add the constraints $P_{1,\rho} = P_{1,\rho_{u\to X,u\to\mathsf{T}}} + P_{1,\rho_{u\to X,u\to\mathsf{F}}}$ and $P_{2,\rho} = P_{2,\rho_{u\to\mathsf{T},u\to X}} + P_{2,\rho_{u\to\mathsf{F},u\to X}}$. **Constraint Set 3**: For every node ρ of the coupling tree that is not in $\mathcal{L}^*$, every $X \in \{\mathsf{T},\mathsf{F}\}$, and every $i \in \{1,2\}$, let u be the first variable of ρ and add the constraint $P_{i,\rho_{u\to X,u\to\neg X}} \le \frac{1}{s} P_{i,\rho}$.

4 Analysis of the linear program for a random formula and how it enables us to conclude Theorem 1

The key lemmas demonstrating the purpose of the linear program are as follows.

▶ **Lemma 24.** *Suppose $r_{\mathsf{lower}} \le |\Omega_1^\Lambda|/|\Omega_2^\Lambda| \le r_{\mathsf{upper}}$. There is a set of variables $\mathbf{P} = \{P_{i,\rho}\}$ that satisfies all constraints of the LP.*

▶ **Lemma 34.** *Fix $r_{\mathsf{lower}} \le r_{\mathsf{upper}}$. W.h.p. over the choice of Φ, the following holds. If the LP has a solution $\mathbf{P}$ using r_{lower} and r_{upper}, then $e^{-\varepsilon/(3n)} r_{\mathsf{lower}} \le |\Omega_1^\Lambda|/|\Omega_2^\Lambda| \le e^{\varepsilon/(3n)} r_{\mathsf{upper}}$.*

The full version proves Theorem 1 using these two lemmas. Here we just give the main idea. First, consider the sub-goal of estimating $|\Omega_1^\Lambda|/|\Omega_2^\Lambda|$ given Φ and a partial assignment Λ of Λ^*. We can do this with accuracy $\exp(\pm\varepsilon/n)$ using the linear program. The proof of Lemma 57 in the full version uses the Lovász local lemma to establish values for r_{lower} and r_{upper} that meet the conditions in Lemma 24. Then, by binary search we bring r_{lower} and r_{upper} closer together until we achieve the desired accuracy (by Lemma 34). The initial values of r_{lower} and r_{upper} guarantee (see the proof of Lemma 57 for details) that the LP is run at most $O(\log(n/\varepsilon))$ times. Since we have already shown that the size of the LP is bounded by a polynomial in n/ε the algorithm runs in polynomial time.

Now consider the proof of Theorem 1. Using standard self-reducibility, we can use the estimates that we have just established to obtain an accurate estimate (within $\exp(\pm\varepsilon)$) of $|\Omega^{\Lambda^*}|/|\Omega|$, which is the probability that a random satisfying assignment is consistent with Λ^*.

To finish we need one last key ingredient – we need a method to estimate $|\Omega^{\Lambda^*}|$. Since all good clauses are satisfied by Λ^*, the set $\mathcal{C}^{\Lambda^*}$ of clauses of Φ^{Λ^*} consists only of bad clauses. Now we need one more key lemma.

▶ **Lemma 48.** *W.h.p. over the choice of Φ, every bad component S has size at most $21600k\log n$.*

Lemma 48 implies that $\mathcal{C}^{\Lambda^*}$ can be divided into disjoint subsets where each subset of clauses contains $O(\log n)$ variables. The algorithm can then compute the number of satisfying assignments of each subset by brute force in time *poly*(n). Then $|\Omega^{\Lambda^*}|$ is the product of these numbers.

This concludes the sketch of the proof of Theorem 1 – the details are in the full version. In the rest of this short version, we briefly discuss the proof of the remaining key lemmas, Lemmas 48, 34, and 24.

We start with the proof of Lemma 48. This lemma, which bounds the size of bad components, is one of the main technical achievements allowing us to extend Moitra's method to random CNFs with high density. Here we only have room for a very rough sketch. Recall that a bad component is a set S of variables that is connected in $H_{\Phi,\text{bad}}$. Let $\mathsf{HD}(S) = \mathcal{V}_0 \cap S$ be the set of high-degree variables in S. We wish to show that w.h.p., over the choice of Φ, every bad component S has size at most $21600k \log n$. This follows from the following two lemmas, which give a contradiction for large bad components S.

▶ **Lemma 42.** *W.h.p. over the choice of Φ, every connected set U of variables with size at least $21600k \log n$ satisfies that $|\mathsf{HD}(U)| \le \frac{|U|}{21600}$.*

▶ **Lemma 47.** *W.h.p. over the choice of Φ, for any bad component S, $|S| \le 60\,|\mathsf{HD}(S)|$.*

The proof of Lemma 42 is deferred to the full version. It uses Lemma 41 and studies trees in the factor graph of Φ. The following proof sketch concludes the proof of Lemma 48.

Proof Sketch of Lemma 47. Consider the following process P which we will use to work with bad compoments. The process, for every set S of variables, defines a set of variables $\mathsf{BC}(S)$.

```
1. Let $\mathsf{BC}(S) = S$.
2. While there is a clause $c$ such that $|\mathsf{var}(c) \cap \mathsf{BC}(S)| \ge k/10$ and $\mathsf{BC}(S) \setminus \mathsf{var}(c) \ne \emptyset$
       $\mathsf{BC}(S) := \mathsf{BC}(S) \cup \mathsf{var}(c)$
```

Note that $\mathcal{V}_{\text{bad}} = \mathsf{BC}(\mathcal{V}_0)$, where $\mathcal{V}_0$ is the set of high-degree variables. We show (Lemma 43) that for every bad component S, we have $S = \mathsf{BC}(\mathsf{HD}(S))$. Thus, the process P can be viewed as a "local" process for identifying bad components.

Let S be a bad component. If S contains only an isolated variable, it must be a high-degree variable and hence $\mathsf{HD}(S) = S$ (so we are finished). Otherwise, since a bad component is a connected component of variables in $H_{\Phi,\text{bad}}$, the definition of $H_{\Phi,\text{bad}}$ ensures that the bad component has at least $k/10$ high-degree variables.

Note that $|\mathsf{HD}(S)| \le |\mathcal{V}_0|$. In Lemma 35 of the full version we use Poisson estimates for the degrees of the variables to show that, w.h.p., $|\mathcal{V}_0| \le n/2^{k^{10}}$.

The next step is to apply a counting argument to show that, w.h.p., for *every* set of variables Y such that $2 \le |Y| \le n/2^k$, the number of clauses that contain at least $k/10$ variables from Y is at most $\frac{30}{k}|Y|$. This is Corollary 38 of the full version. We apply the corollary with $Y = \mathsf{HD}(S)$, so we find that there are at most $\frac{30}{k}\,|\mathsf{HD}(S)|$ clauses that contain at least $k/10$ variables from $\mathsf{HD}(S)$.

Now, we run the process P starting with $\mathsf{HD}(S)$. Take Z to be the set of clauses that contain at least $k/10$ variables from $\mathsf{HD}(S)$ (so, from above, we have $|Z| \le \frac{30}{k}\,|\mathsf{HD}(S)| \le \frac{30}{k}\frac{n}{2^{k^{10}}}$).

The next step is to show that, w.h.p., the number of clauses c such that $\mathsf{var}(c) \subseteq \mathsf{BC}(\mathsf{HD}(S))$ is at most $2|Z|$ (which we have already shown to be at most $60\,|\mathsf{HD}(S)|\,/k$). This analysis is contained in Corollary 45. It is essentially an analysis of the process P which follows easily from a lemma of Coja-Oghlan and Frieze [10, Lemma 2.4]. The high-probability guarantees are universal over Z (hence universal over S).

Since $S = \mathsf{BC}(\mathsf{HD}(S))$ and each variable in S is contained in some bad clause, we have

$$|S| \le \left|\bigcup_{c \in \mathcal{C}_{\text{bad}}:\ \mathsf{var}(c) \cap S \ne \emptyset} \mathsf{var}(c)\right| \le 60\,|\mathsf{HD}(S)|\,, \text{ as required.}$$ ◀

We now turn to the proof of Lemma 34. There are two kinds of errors which cause solutions of the LP to differ from the ratio $\left|\Omega_1^\Lambda\right|/\left|\Omega_2^\Lambda\right|$. The first kind comes from so-called "ℓ-wrong assignments" and the second kind comes from the truncation of the coupling tree. To define these more precisely, we need some graph-theoretic notation.

▶ **Definition 25.** *Given a graph G and any positive integer k, let $G^{\leq k}$ be the graph with vertex set $V(G)$ in which vertices u and v are connected iff there is a path from u to v in G of length at most k.*

The main combinatorial structure that we use is a set $\mathcal{D}(G_\Phi)$, which is based on Alon's "2,3-tree" [5]. Similar structures were subsequently used in [30, 21]. The main difference between our definition and previous ones is that we take into account whether clauses are connected via good variables.

▶ **Definition 26.** *Given G_Φ, let $\mathcal{D}(G_\Phi)$ be the set of subsets $T \subseteq V(G_\Phi)$ such that (1) For any $c_1, c_2 \in T$, $\mathsf{var}(c_1) \cap \mathsf{var}(c_2) \cap \mathcal{V}_{good} = \emptyset$; and (2) The graph $G_\Phi^{\leq 4}[T]$, which is the subgraph of $G_\Phi^{\leq 4}$ induced by T, is connected.*

▶ **Definition 29.** *An assignment $\sigma \in \Omega_i^\Lambda$ is ℓ-wrong if $\exists$ a size-ℓ set $T \in \mathcal{D}(G_\Phi)$ such that $c^* \in T$, $|T \cap \mathcal{C}_{good}^\Lambda| \geq 2\,|T|/3$, and there is a size $\lceil \ell/2 \rceil$ subset S of $T \cap \mathcal{C}_{good}^\Lambda$ such that the restriction of σ to marked variables in clauses in S does not satisfy any clause in S. Otherwise σ is ℓ-correct.*

Proof Sketch of Lemma 34. The constraints in **Constraint Set 2** guarantee (see Lemma 18 of the full version) that, for any $i \in \{1,2\}$ and $\sigma \in \Omega_i^\Lambda$, $\sum_{\rho \in \mathcal{L}^*:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho} = 1$. Thus, $|\Omega_i^\Lambda| = \sum_{\sigma \in \Omega_i^\Lambda} 1 = \sum_{\sigma \in \Omega_i^\Lambda} \sum_{\rho \in \mathcal{L}^*:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho}$. Let $\ell = L/(3k^2\Delta)$. We start by defining Z_i, Z_i' and Z_i'' as follows for $i \in \{1,2\}$.

$$Z_i = \sum_{\sigma \in \Omega_i^\Lambda} \quad \sum_{\rho \in \mathcal{L}:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho},$$

$$Z_i' = \sum_{\sigma \in \Omega_i^\Lambda,\ \sigma \text{ is } \ell\text{-wrong}} \quad \sum_{\rho \in \mathcal{L}^*:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho},$$

$$Z_i'' = \sum_{\sigma \in \Omega_i^\Lambda,\ \sigma \text{ is } \ell\text{-correct}} \quad \sum_{\rho \in \mathcal{T}:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho}.$$

Thus $Z_i \leq \left|\Omega_i^\Lambda\right| \leq Z_i + Z_i' + Z_i''$. The full version proves

$$r_{\mathsf{lower}} \leq Z_1/Z_2 \leq r_{\mathsf{upper}}. \tag{2}$$

$$Z_i'/|\Omega_i^\Lambda| \leq (1 - e^{-\varepsilon/(3n)})/2 \text{ for } i \in \{1,2\}. \tag{3}$$

$$Z_i''/|\Omega_i^\Lambda| \leq (1 - e^{-\varepsilon/(3n)})/2 \text{ for } i \in \{1,2\}. \tag{4}$$

The lemma follows easily from these. Combining (3) and (4) with the fact that $Z_i \leq \left|\Omega_i^\Lambda\right| \leq Z_i + Z_i' + Z_i''$, we get $e^{-\varepsilon/(3n)} \leq \frac{Z_i}{|\Omega_i^\Lambda|} \leq 1$. Plugging in (2) we obtain the result.

To prove (2) we exchange the order of summation in the definition of Z_i to get

$$Z_i = \sum_{\rho \in \mathcal{L}} \quad \sum_{\sigma \in \Omega_i^\Lambda:\sigma \in \Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho} = \sum_{\rho \in \mathcal{L}} P_{i,\rho} \cdot |\Omega^{\mathcal{A}_i(\rho)\cup\Lambda}|.$$

Since $\rho \in \mathcal{L}$, we prove (see Lemma 17) that $r(\rho) = |\Omega^{\mathcal{A}_1(\rho)\cup\Lambda}|/|\Omega^{\mathcal{A}_2(\rho)\cup\Lambda}|$ (this is actually the point of $r(\rho)$). **Constraint Set 1** then guarantees that

$$r_{\mathsf{lower}} \leq \frac{P_{1,\rho} \cdot \left|\Omega^{\mathcal{A}_1(\rho)\cup\Lambda}\right|}{P_{2,\rho} \cdot \left|\Omega^{\mathcal{A}_2(\rho)\cup\Lambda}\right|} = \frac{P_{1,\rho} \cdot r(\rho)}{P_{2,\rho}} \leq r_{\mathsf{upper}}, \text{ which suffices.}$$

The main ingredient in the proof of (3) is Lemma 30, which shows that the fraction of assignments in Ω_i^Λ that are ℓ-wrong is at most $(k\Delta)^{-9\ell}$. The main ingredient in the proof of (4) is showing that, w.h.p., for every ℓ-correct $\sigma \in \Omega_i^\Lambda$, $\sum_{\rho \in \mathcal{T}:\sigma\in\Omega^{\mathcal{A}_i(\rho)\cup\Lambda}} P_{i,\rho} \le (k\Delta)^{-8\ell}$. This is handled in Lemmas 32 and 33.

To prove these lemmas (say for $i = 1$) we consider a sampling procedure for choosing a node $\rho \in \mathcal{L}^*$ conditioned on some $\sigma \in \Omega_1^\Lambda$. The probability that it reaches any node $\rho \in \mathcal{L}^*$ with $\sigma \in \Omega^{\mathcal{A}_1(\rho)\cup\Lambda}$ is designed to be $P_{1,\rho}$ so the goal is to bound the probability that it reaches the set $\Upsilon_\sigma = \{\rho \in \mathcal{T} \mid \sigma \in \Omega^{\mathcal{A}_1(\rho)\cup\Lambda}\}$. This is where the combinatorial structures that we have defined come in. We use $\mathcal{F}(\rho)$ to denote the set of clauses that "fail" in the coupling process, contributing variables to $V_I(\rho)$. Lemma 28 shows that w.h.p., for every node $\rho \in \Upsilon_\sigma$, there is a set $T \subseteq \mathcal{F}(\rho)$ containing the first clause c^* such that $T \in \mathcal{D}(G_\Phi)$, $|T| = \ell$ and $|T \cap \mathcal{C}_{\text{bad}}| \le |T|/3$. This implies that $|T \cap C_{\text{good}}^\Lambda| \ge 2|T|/3$. We therefore need to upper bound the probability that such a T is contained in $\mathcal{F}(\rho)$ when ρ is chosen from the sampling procedure. T has size ℓ and contains c^* and contains enough good clauses. So it turns out that, since σ is ℓ-correct, a lot of these failed clauses in $\mathcal{F}(\rho)$ must have failed due to disagreements in the coupling. Since $T \in \mathcal{D}(G_\Phi)$ these clauses do not share good variables. The constraints in **Constraint Set 3** then imply that the probability of all of these simultaneous disagreements is unlikely.

That concludes the proof, apart from proving the key Lemma 28. This again relies on properties about bad components - in particular on Lemma 50, which says that, w.h.p., for every connected set of clauses Y such that $|\mathsf{var}(Y)| \ge 21600k\log n$, it holds that $|Y \cap \mathcal{C}_{\text{bad}}| \le |Y|/12$. This is somewhat similar to the issues that we discussed regarding the proof of Lemma 48 – we defer the details to the full version. ◀

Proof Sketch of Lemma 24. Suppose $r_{\mathsf{lower}} \le |\Omega_1^\Lambda|/|\Omega_2^\Lambda| \le r_{\mathsf{upper}}$. Our goal is to show that there is a set of variables $\mathbf{P} = \{P_{i,\rho}\}$ that satisfies all constraints of the LP. Here is a suitable assignment. Let ρ be a node of the coupling tree with first variable u. For $X \in \{\mathsf{T}, \mathsf{F}\}$, we use the notation $\psi_{\rho,X,1} := |\Omega^{\mathcal{A}_1(\rho_{u\to X,u\to X})\cup\Lambda}|/|\Omega^{\mathcal{A}_1(\rho)\cup\Lambda}| = |\Omega^{\mathcal{A}_1(\rho_{u\to X,u\to\neg X})\cup\Lambda}|/|\Omega^{\mathcal{A}_1(\rho)\cup\Lambda}|$. This is well-defined since $\mathcal{A}_1(\rho_{u\to X,u\to X}) = \mathcal{A}_1(\rho_{u\to X,u\to\neg X})$. In other words, $\psi_{\rho,X,1}$ is the probability that u is assigned value X under the uniform distribution on $\Omega^{\mathcal{A}_1(\rho)\cup\Lambda}$. We similarly define $\psi_{\rho,X,2} := |\Omega^{\mathcal{A}_2(\rho_{u\to X,u\to X})\cup\Lambda}|/|\Omega^{\mathcal{A}_2(\rho)\cup\Lambda}| = |\Omega^{\mathcal{A}_2(\rho_{u\to\neg X,u\to X})\cup\Lambda}|/|\Omega^{\mathcal{A}_2(\rho)\cup\Lambda}|$.

We will next give an inductive definition of a function Q from nodes of the coupling tree to real numbers in $[0, 1]$. The way to think about this is as follows – we will implicitly define a probability distribution over paths from the root of the coupling tree to $\mathcal{L}^*$. For each node ρ, $Q(\rho)$ will be the probability that ρ is included in a path drawn from this distribution.

Any such path starts at the root, so we define $Q(\rho^*) = 1$. Once we have defined $Q(\rho)$ for a node ρ that is not in $\mathcal{L}^*$ we can define $Q(\cdot)$ on the children of ρ as follows. Let u be the first variable of ρ and consider the four children $\rho_{u\to\mathsf{T},u\to\mathsf{T}}, \rho_{u\to\mathsf{T},u\to\mathsf{F}}, \rho_{u\to\mathsf{F},u\to\mathsf{T}}, \rho_{u\to\mathsf{F},u\to\mathsf{F}}$. Define the values of Q as follows: $Q(\rho_{u\to\mathsf{T},u\to\mathsf{T}}) := Q(\rho)\min\{\psi_{\rho,\mathsf{T},1}, \psi_{\rho,\mathsf{T},2}\}$, $Q(\rho_{u\to\mathsf{T},u\to\mathsf{F}}) := Q(\rho)(\psi_{\rho,\mathsf{T},1} - \min\{\psi_{\rho,\mathsf{T},1}, \psi_{\rho,\mathsf{T},2}\})$, $Q(\rho_{u\to\mathsf{F},u\to\mathsf{F}}) := Q(\rho)\min\{1 - \psi_{\rho,\mathsf{T},1}, 1 - \psi_{\rho,\mathsf{T},2}\}$, and $Q(\rho_{u\to\mathsf{F},u\to\mathsf{T}}) := Q(\rho)((1 - \psi_{\rho,\mathsf{T},1}) - \min\{1 - \psi_{\rho,\mathsf{T},1}, 1 - \psi_{\rho,\mathsf{T},2}\})$. Finally, we define $P_{i,\rho} := Q(\rho)|\Omega_i^\Lambda|/|\Omega^{\mathcal{A}_i(\rho)\cup\Lambda}|$. In the full version, we prove that this assignment satisfies the constraints of the LP. Mostly, the LP is designed to make this true, though for example to establish the constraint $P_{i,\rho_{u\to X,u\to\neg X}} \le \frac{1}{s}P_{i,\rho}$ (Lemma 23) we need to prove that $\psi_{\rho,X,i}$ is around 1/2. Like Moitra, we prove this using the Lovász local lemma, so this is why it is essential that we restrict the LP to good variables. ◀

References

1 E. Abbe and A. Montanari. On the concentration of the number of solutions of random satisfiability formulas. *Random Struct. Algorithms*, 45(3):362–382, 2014.

2 D. Achlioptas and A. Coja-Oghlan. Algorithmic barriers from phase transitions. In *FOCS*, pages 793–802. IEEE Computer Society, 2008.

3 D. Achlioptas and C. Moore. The asymptotic order of the random k-SAT threshold. In *FOCS*, pages 779–788. IEEE Computer Society, 2002.

4 D. Achlioptas and Y. Peres. The threshold for random k-SAT is $2^k(\ln 2 - o(k))$. In *STOC*, pages 223–231. ACM, 2003.

5 N. Alon. A parallel algorithmic version of the local lemma. *Random Struct. Algorithms*, 2(4):367–378, 1991. doi:10.1002/rsa.3240020403.

6 I. Bezáková, A. Galanis, L. A. Goldberg, H. Guo, and D. Štefankovič. Approximation via correlation decay when strong spatial mixing fails. *SIAM J. Comput.*, 48(2):279–349, 2019.

7 K. Chandrasekaran, N. Goyal, and B. Haeupler. Deterministic algorithms for the Lovász local lemma. *SIAM J. Comput.*, 42(6):2132–2155, 2013.

8 A. Coja-Oghlan. A better algorithm for random k-SAT. *SIAM J. Comput.*, 39(7):2823–2864, 2010.

9 A. Coja-Oghlan. Belief propagation guided decimation fails on random formulas. *J. ACM*, 63(6):Art. 49, 55, 2017.

10 A. Coja-Oghlan and A. Frieze. Analyzing Walksat on random formulas. *SIAM J. Comput.*, 43(4):1456–1485, 2014.

11 A. Coja-Oghlan, A. Haqshenas, and S. Hetterich. Walksat stalls well below satisfiability. *SIAM J. Discrete Math.*, 31(2):1160–1173, 2017.

12 A. Coja-Oghlan and K. Panagiotou. The asymptotic k-SAT threshold. *Adv. Math.*, 288:985–1068, 2016.

13 A. Coja-Oghlan and D. Reichman. Sharp thresholds and the partition function. *Journal of Physics: Conference Series*, 473:012015, 2013.

14 A. Coja-Oghlan and N. Wormald. The number of satisfying assignments of random regular k-SAT formulas. *Combin. Probab. Comput.*, 27(4):496–530, 2018.

15 J. Ding, A. Sly, and N. Sun. Proof of the satisfiability conjecture for large k. In *STOC*, pages 59–68. ACM, 2015.

16 C. Efthymiou, T. P. Hayes, D. Štefankovič, and E. Vigoda. Sampling random colorings of sparse random graphs. In *SODA*, pages 1759–1771. SIAM, 2018.

17 P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets, volume 10 of Colloquia Mathematica Societatis János Bolyai*, pages 609–628, 1975.

18 W. Feng, H. Guo, Y. Yin, and C. Zhang. Fast sampling and counting k-sat solutions in the local lemma regime, 2019. arXiv:1911.01319.

19 E. Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *J. Amer. Math. Soc.*, 12(4):1017–1054, 1999. With an appendix by Jean Bourgain. doi:10.1090/S0894-0347-99-00305-7.

20 H. Guo, M. Jerrum, and J. Liu. Uniform sampling through the Lovász local lemma. *J. ACM*, 66(3):18:1–18:31, 2019.

21 H. Guo, C. Liao, P. Lu, and C. Zhang. Counting hypergraph colorings in the local lemma regime. *SIAM J. Comput.*, 48(4):1397–1424, 2019.

22 B. Haeupler, B. Saha, and A. Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):Art. 28, 28, 2011.

23 J. Hermon, A. Sly, and Y. Zhang. Rapid mixing of hypergraph independent sets. *Random Struct. Algorithms*, 54(4):730–767, 2019.

24 S. Hetterich. Analysing survey propagation guided decimationon random formulas. In *ICALP*, volume 55 of *LIPIcs*, pages 65:1–65:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

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

26 C. Liao, J. Lin, P. Lu, and Z. Mao. Counting independent sets and colorings on random regular bipartite graphs. In *APPROX-RANDOM*, volume 145 of *LIPIcs*, pages 34:1–34:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

27 M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 94:197205, 2005.

28 M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

29 M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

30 A. Moitra. Approximate counting, the Lovász local lemma, and inference in graphical models. *J. ACM*, 66(2):Art. 10, 25, 2019.

31 A. Montanari and D. Shah. Counting good truth assignments of random k-SAT formulae. In *SODA*, pages 1255–1264. SIAM, 2007.

32 R. A. Moser and G. Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010.

33 E. Mossel and A. Sly. Exact thresholds for Ising-Gibbs samplers on general graphs. *Ann. Probab.*, 41(1):294–328, 2013.

34 J. Schmidt-Pruzan and E. Shamir. Component structure in the evolution of random hypergraphs. *Combinatorica*, 5(1):81–94, 1985.

35 A. Sly, N. Sun, and Y. Zhang. The number of solutions for random regular NAE-SAT. In *FOCS*, pages 724–731. IEEE Computer Society, 2016.

36 J. Spencer. Asymptotic lower bounds for Ramsey functions. *Discrete Math.*, 20(1):69–76, 1977.

37 Y. Yin and C. Zhang. Sampling in Potts model on sparse random graphs. In *APPROX-RANDOM*, volume 60 of *LIPIcs*, pages 47:1–47:22. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.