COUNTING SOLUTIONS TO RANDOM CNF FORMULAS

ANDREAS GALANIS, LESLIE ANN GOLDBERG, HENG GUO, AND KUAN YANG

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 $(1 + o_k(1))^{2 \log k}$, the Gibbs uniqueness threshold for the model. Instead, our algorithm harnesses a recent technique by Moitra to work for random formulas. 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.

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 $\Phi$ shows an interesting threshold behaviour, where the asymptotic probability that $\Phi$ is satisfiable drops dramatically from 1 to 0 when the density $\alpha := \frac{m}{n}$ crosses a certain threshold $\alpha_\star$. There has been tremendous progress on establishing this phase transition and pinpointing the threshold $\alpha_\star$ [24, 18, 3, 4, 12, 15] guided by elaborate but non-rigorous methods in physics [27, 26]. 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 $\Phi$ is close to (but smaller than) the satisfiability threshold $\alpha_\star$. The best polynomial-time algorithm to find a satisfying assignment of $\Phi$ is due to Coja-Oghlan [9], which succeeds if $\alpha < (1 - o_k(1)) \cdot 2^k \ln k/k$. It is known that beyond this density bound $2^k \ln k$ the solution space of the formula undergoes a phase transition and becomes severely more complicated [2], where algorithms are bound to fail to find a satisfying assignment in polynomial time (see for example [23, 10, 8]).

It is also a natural question to determine the number of satisfying assignments to $\Phi$, 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 $\mathbb{E} \frac{1}{n} \log Z(\Phi)$ 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 $\mathbb{E} \frac{1}{n} \log Z(\Phi)$, see [34, 14] for progress along these lines for more symmetric models of random formulas. Moreover, Montanari and Shah [30] has given an efficient algorithm to approximate $\log Z(\Phi)$ if $\alpha \leq 2^k \ln k(1 + o_k(1))$, based on the correlation decay method and the uniqueness threshold of the Gibbs distribution. Note that this gives only an exponential approximation to $Z(\Phi)$. Also, the threshold 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 $\alpha$ is less than $2^k r$, for sufficiently large $k$ and some constant $r > 0$. 

Date: 26 November 2019.

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.
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, etc. [32, 36, 16, 25] in random graphs. However, previous methods, such as Markov chains Monte Carlo or Barvinok’s method, appear to be difficult to apply for random formulas. Instead, our algorithm is the first adaptation of Moitra’s method [29] to the random instance setting, and it is intimately related to the Lovász local lemma. We give a high level overview of the techniques in Section 1.2.

1.1. The model and the main result. For $k \geq 3$, 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, $\Phi$ 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 $m = \lfloor na \rfloor$, where $\alpha > 0$ is the density of the formula. We say that an event $E$ holds w.h.p. if $\Pr(E) = 1 - o(1)$ as $n \to \infty$.

For a $k$-SAT formula $\Phi$, we let $\Omega = \Omega(\Phi)$ denote the set of satisfying assignments of $\Phi$ and $\pi_\Phi$ be the uniform distribution on $\Omega$ (provided that the latter is non-empty).

**Theorem 1.** There is a polynomial-time algorithm $A$ and two constants $r > 0, k_0 \geq 3$ such that for all $k \geq k_0$ and all $\alpha < 2^k$ the following holds w.h.p. over the choice of the random $k$-SAT formula $\Phi = \Phi(k, n, m)$. The algorithm $A$, on input the formula $\Phi$ and a rational $\varepsilon > 0$, outputs in time $\text{poly}(n, 1/\varepsilon)$ a number $Z$ that satisfies $e^{-\varepsilon}\lvert\Omega(\Phi)\rvert \leq Z \leq e^\varepsilon\lvert\Omega(\Phi)\rvert$.

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 $\alpha$ of the formula $\Phi$ satisfies $\alpha < 2^{k/300}/k^2$, so $r$ can be taken as $\frac{1}{307}$ in Theorem 1. The constant 300 here is not optimised, but we do not expect to use the current techniques to improve it substantially. Our main point here is that for a density exponential in $k$, 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 [33]) that w.h.p. every connected component of $\Phi$, viewed as a hypergraph where variables correspond to vertices and clauses correspond to hyperedges, is of size $O(\log n)$. In such 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 diving into the details. Efficient approximate counting of satisfying assignments of $k$-CNF formulas have been a challenging problem for traditional techniques, since the solution space is complicated and it is disconnected for Markov chains. Recently some new approaches were introduced [29, 19]. Most notably, the breakthrough work of Moitra [29] gives the first (and so far the only) efficient algorithm that can approximately count the number of satisfying assignments of $k$-CNF formulas where each variable appears in at most $d$ clauses, if, roughly speaking, $d \lesssim 2^{k/60}$.

As our goal is to count satisfying assignments of sparse random $k$-CNF formulas, it is natural to choose Moitra’s method in the random instance setting. However, the first difficulty is that Moira’s method is intimately connected to the Lovász local lemma, which requires bounded occurrences of variables. In contrast, for a random $k$-CNF formula, although the average degree of variables is low, there are variables with degrees as high as $\Omega \left( \frac{\log n}{\log \log n} \right)$ with high probability. In the presence of these high degree variables, the local lemma does not hold any more.

Our solution to this issue is to separate out high degree variables as well as those that are heavily influenced by high degree variables. In particular, 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 well as all variables they contain. We run this process until no more bad clause can be found. In the rest of the formula, all good variables
have a degree upper bound, and all good clauses still contain at least $9k/10$ variables to meet the conditions of the local lemma. Such a process has not been analysed for random formulas before.

Once we have found out the “good” variables, we follow Moitra’s method on these variables. We now explain what Moitra’s method does and then proceed with our modifications. The first step is to mark variables, so that every clause contains a good fraction of marked and unmarked variables. Then, for a particular marked variable $v$, we set up a linear program whose variables correspond to states of a greedy disagreement coupling between two distributions on marked variables conditioned on different values of $v$. Solving this linear program yields enough information to compute the marginal probability of $v$. However, to obtain a polynomial size of the linear program, we truncate the coupling at a suitable level. Thus, the crucial part is to show that the error caused by the truncation is sufficiently small. To count the number of satisfying assignments, we use self-reducibility to sequentially estimate the marginal probability of variables, conditioned on setting previous variables.

For a random formula, we can only mark/unmark good variables, and we “give up” on bad variables. Because of this, the main change to the procedure above is to the design and the analysis of the disagreement coupling. The reason that Moitra’s analysis works is that the local lemma guarantees local uniformity of marked variables. Now this property only holds for good marked variables, and we have lost control on bad variables completely. 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. The most important part of our analysis is to upper bound the size of connected bad components. With this upper bound, we rework all of Moitra’s analysis incorporating the presence of bad variables, and show that the coupling process can be successfully truncated at a desired size, and the error caused by the truncation is not too large.

Although we have given an efficient algorithm for densities that are exponentially large in $k$, our constants are really small comparing to the satisfiability threshold or the threshold under which efficient search algorithms exist. Perhaps a modest start towards comparable thresholds is 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. [22] showed that efficient randomised algorithms exist if $d \leq c^{2k/2}$ 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 $d \leq c^{2k/k}$ 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. High-degree and bad variables

We will apply the method of Moitra [29], which was introduced to approximately count the satisfying assignments of $k$-CNF formulas in which each literal appears a bounded number of times. The main difference between the formulas studied by Moitra and the random formulas that we study is that, in our formulas, some variables will occur many more times than the average.

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

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 $\Phi$. Formally, denote the set of clauses of $\Phi$ by $C$ and the set of variables by $V$. For each $c \in C$, let $\text{var}(c)$ denote the set of variables in $c$. For each subset $C$ of $C$, let $\text{var}(C) := \bigcup_{c \in C} \text{var}(c)$. The bad variables and bad clauses of $\Phi$ are identified by running the following process:

1. $V_0$ (the initial bad variables) ← the set of high-degree variables;
2. $C_0$ ← the set of clauses with at least $k/10$ variables in $V_0$;
Theorem 7 (The local lemma). For each $c \in C$, let $A_c$ be a subset of $\Omega^*$ such that $\var(c) \subseteq \var(c)$. If there exists a function $x: C \to (0, 1)$ such that, for all $c \in C$,

\begin{equation}
\Pr_{\mu^*}(A_c) \leq x(c) \prod_{b \in \Gamma(c)} \left(1 - x(b)\right),
\end{equation}

then

\begin{equation}
\Pr_{\mu^*}\left(\bigwedge_{c \in C} \neg A_c\right) > 0.
\end{equation}

Furthermore, for any $A \subseteq \Omega^*$, $\Pr_{\mu^*}\left(A \cap \bigwedge_{c \in C} \neg A_c\right) \leq \Pr_{\mu^*}(A) \prod_{b \in \Gamma(A)} \left(1 - x(b)\right)^{-1}.$
The second part of Theorem 7 is due to Haeupler et al. [21]. It provides an upper bound on the probability of an event under the uniform distribution over satisfying assignments. It is also possible to find an assignment such that $\bigwedge_{c \in C} \overline{A_c}$ holds in $O(|V|)$ time by the algorithm of Moser and Tardos [31].

In the course of our algorithm it is useful to “mark” some of the variables in $V_{\text{good}}$. For this, we use the approach of Moitra [29]. Formally, a “marking” is an assignment from $V_{\text{good}}$ to $\{\text{marked, unmarked}\}$. Using Theorem 7, we have the following lemma.

**Lemma 8.** There exists a marking on $V_{\text{good}}$ such that:

1. every good clause has at least $3k/10$ marked variables and at least $k/4$ unmarked good variables;
2. there is a partial assignment of bad (and thus unmarked) variables that satisfies all bad clauses.

Furthermore, such a marking can be found in deterministic polynomial time.

**Proof.** We apply Theorem 7 on $C_{\text{good}}$ and $V_{\text{good}}$. Let $\Omega_{\text{good}}^{*}$ be all possible markings: $V_{\text{good}} \rightarrow \{\text{marked, unmarked}\}$, so $\mu_{\Omega_{\text{good}}^{*}}$ is the distribution in which each good variable is marked independently and uniformly at random. For $c \in C_{\text{good}}$, let $M$ be the number of marked good variables in $\var(c)$. Let $A_c$ be the subset of $\Omega_{\text{good}}^{*}$ that $M < 3k/10$ or $M > 3k/5$. Since $c \in C_{\text{good}}$, $c$ contains at least $9k/10$ good variables (see the observation immediately following the process that defines bad variables and clauses). Thus $9k/20 \leq \mathbb{E}_{\mu_{\Omega_{\text{good}}^{*}}} M \leq 2k$, and the number of unmarked good variables in $\var(c)$ is at least $9k/10 - M$. By a Chernoff bound,

$$\Pr_{\mu_{\Omega_{\text{good}}^{*}}} (A_c) = \Pr_{\mu_{\Omega_{\text{good}}^{*}}} (M > 3k/5) + \Pr_{\mu_{\Omega_{\text{good}}^{*}}} (M < 3k/10) \leq e^{-k/110} + e^{-k/40} \leq 2e^{-k/150}.$$ 

Let $x(c) = \frac{1}{k\Delta}$ if $c \in C_{\text{good}}$. Since we only consider assignments of $V_{\text{good}}$, so $\var(A_c) \subseteq \var(c) \cap V_{\text{good}}$ and thus $\Gamma(A_c) \subseteq \Gamma_{\text{good}}(c)$. Note that $\Gamma_{\text{good}}(c)$ is the set of neighbours of $c$ in $G_{\Phi, \text{good}}$, and the maximum degree in $\Gamma_{\text{good}}(c)$ is at most $k(\Delta - 1)$, so we can verify (1) as follows, for any $c \in C_{\text{good}}$ and sufficiently large $k$,

$$x(c) \prod_{b \in \Gamma(A_c)} \left(1 - x(b)\right) \geq x(c) \prod_{b \in \Gamma_{\text{good}}(c)} \left(1 - x(b)\right) \geq \frac{1}{k\Delta} \left(1 - \frac{1}{k\Delta}\right)^{k\Delta} \geq \frac{1}{k\Delta} \geq \frac{1}{e^{k\Delta}} \\
\geq e^{-k/300 - 1/k} \geq 2e^{-k/150} \geq \Pr_{\mu_{\Omega_{\text{good}}^{*}}} (A_c).$$

Thus, there is a marking such that item (1) holds.

Item (2) always holds for any marking of $V_{\text{good}}$. This is because our density is well below the critical threshold [15]. Thus there is at least one satisfying assignment to $\Phi$. As $\var(c) \cap V_{\text{good}} = \emptyset$ for any $c \in C_{\text{bad}}$, the restriction of the satisfying assignment to $V_{\text{bad}}$ satisfies the condition.

The marking can be found using the deterministic algorithm [7] by verifying

$$\Pr_{\mu_{\Omega_{\text{good}}^{*}}} (A_c) \leq \left( x(c) \prod_{b \in \Gamma_{\text{good}}(c)} \left(1 - x(b)\right) \right)^{1.01} \quad \square$$

We will stick to an arbitrary marking given by Lemma 8 from now on. We use $\text{marked}(c)$ to denote the marked variables in clause $c$ and $V_{\text{marked}} := \cup_{c \in C_{\text{good}}} \text{marked}(c)$ be the set of all marked variables.

Let $\Omega$ be the set of satisfying assignments of $\Phi$. We will be particularly interested in the uniform distribution $\mu_{\Omega}(\cdot)$. For any partial assignment $\Lambda$ of some of $\text{good}$ variables of $\Phi$, let $\Phi^\Lambda$ be the formula of $\Phi$ simplified under $\Lambda$. In other words, we remove all satisfied clauses under $\Lambda$ and all false literals in all clauses. (Some clauses may become empty in $\Phi^\Lambda$, in which case $\Phi^\Lambda$ cannot be satisfied.) We use $C^\Lambda$ to denote the set of clauses of $\Phi^\Lambda$, and similarly $V^\Lambda$. We also define $V_{\text{good}}^\Lambda = V_{\text{good}} \cap V^\Lambda$ and $C_{\text{good}}^\Lambda = C_{\text{good}} \cap C^\Lambda$ to denote the sets of remaining good variables
and clauses of $\Phi$ simplified under $\Lambda$. (Remark: Note that $\Lambda$ does not contain bad variables so if we define $\Phi_{\text{bad}}^\Lambda$ and $C_{\text{bad}}^\Lambda$ similarly then we will have $\Phi_{\text{bad}}^\Lambda = \Phi_{\text{bad}}$ and $C_{\text{bad}}^\Lambda = C_{\text{bad}}$.) Let $\Omega^\Lambda$ denote the set of satisfying assignments of $\Phi^\Lambda$, namely, those satisfying assignments of $\Phi$ consistent with $\Lambda$.

Let $s := 2^{k/4}/(ek\Delta)$. Under an arbitrary conditioning of marked variables, we have good control of events defined by good variables, and the marginal distribution of good variables.

**Lemma 9.** Let $V \subseteq V_{\text{marked}}$. For any partial assignment $\Lambda : V \rightarrow \{T,F\}$ such that $\Lambda \not\equiv \emptyset$ and any subset $A \subseteq \Omega^\Lambda$ such that $\text{var}(A) \subseteq \Omega_{\text{good}} \setminus V$,

$$Pr_{\mu^\Lambda}(A) \leq Pr_{\mu^\Lambda}(A) \left(1 - \frac{1}{sk\Delta}\right)^{-|\Gamma(A)|},$$

where $A$ is the set of all assignments of $\text{var}(A)$. In particular, for $v \in V_{\text{marked}} \setminus V$,

$$\left(1 - \frac{1}{3s}\right)\frac{1}{2} \leq Pr_{\mu^\Lambda}(v \rightarrow T) \leq \left(1 + \frac{1}{3s}\right)\frac{1}{2}. $$

**Proof.** Let $\Lambda_{\text{bad}}$ be an arbitrary assignment on $V_{\text{bad}}$ such that all bad clauses are satisfied. Such a $\Lambda_{\text{bad}}$ exists because of Item (2) of Lemma 8.

We apply Theorem 7 to $\Phi^\Lambda \cup \Lambda_{\text{bad}}$. The remaining variables are $V' = \Omega_{\text{good}} \setminus V$ and clauses are $C' := C_{\text{good}}^\Lambda \cup \Lambda_{\text{bad}}$. Let $\Omega'$ be the set of all $2^{|V'|}$ assignments $V' \rightarrow \{T,F\}$. Let $A_c$ be the subset of $\Omega'$ that $c$ is not satisfied, for $c \in C'$. Then,

$$Pr_{\mu^\Lambda}(\cdot \mid \Lambda_{\text{bad}}) = Pr_{\mu^\Lambda}(\cdot \mid \bigwedge_{c \in C'} \overline{A_c}).$$

We verify the condition (1) of Theorem 7 and show

$$Pr_{\mu^\Lambda}(A \mid \bigwedge_{c \in C'} \overline{A_c}) \leq Pr_{\mu^\Lambda}(A) \left(1 - \frac{1}{sk\Delta}\right)^{-|\Gamma(A)|},$$

and

$$\left(1 - \frac{1}{3s}\right)\frac{1}{2} \leq Pr_{\mu^\Lambda}(v \rightarrow T \mid \bigwedge_{c \in C'} \overline{A_c}) \leq \left(1 + \frac{1}{3s}\right)\frac{1}{2}. $$

The first part of the lemma follows directly and the second part follows from (2), (4), and the law of total probabilities.

By Lemma 8, there are at least $k/4$ unmarked variables in each $c \in C'$. Thus,

$$Pr_{\mu^\Lambda}(A_c) \leq 2^{-k/4}. $$

Let $x(c) = \frac{1}{sk\Delta}$. Since $\Omega'$ is a set of assignments of good variables, we have $\text{var}(A_c) \subseteq \text{var}(c) \cap \Omega_{\text{good}}$, and thus $\Gamma(A_c) \subseteq \Gamma_{\text{good}}(c)$. Again, note that $\Gamma_{\text{good}}(c)$ is the set of neighbours of $c$ in $G_{\Phi,\text{good}}$ and the maximum degree of $G_{\Phi,\text{good}}$ is at most $k(\Delta - 1)$, so we obtain

$$x(c) \prod_{b \in \Gamma(A_c)} \left(1 - x(b) \right) \geq x(c) \prod_{b \in \Gamma_{\text{good}}(c)} \left(1 - x(b) \right) \geq \frac{1}{sk\Delta} \left(1 - \frac{1}{sk\Delta}\right)^{k(\Delta - 1)} 
\geq e^{-1/s} \geq \frac{1}{ek\Delta} = 2^{-k/4}. $$

The two inequalities above verify condition (1) of Theorem 7. Thus, (3) follows directly since $\text{var}(A) \subseteq V'$. Moreover, setting $A = v \rightarrow T$, for sufficiently large $k$,

$$Pr_{\mu^\Lambda}(v \rightarrow T \mid \bigwedge_{c \in C'} \overline{A_c}) \leq \frac{1}{2} \left(1 - \frac{1}{sk\Delta}\right)^{-\Delta} \leq \frac{1}{2} \cdot e^{\Delta/(sk\Delta - 1)} \leq \frac{1}{2} \left(1 + \frac{1}{3s}\right). $$

We get the same upper bound for the event $v \rightarrow \overline{F}$ by the same argument. The bound (4) follows by combining these two bounds. $\Box$
Moreover, we have the following lemma for a partial assignment that we will use to apply self-reducibility.

**Lemma 10.** Let \( \Phi = \Phi(k, n, m) \) and \( v_1, v_2, \ldots, v_n \) be \( n \) variables in \( \Phi \). Suppose that in each clause, the literals are in the order of indices of their corresponding variables. Then there is a partial assignment \( \Lambda^* \) for some subset of \( V_{\text{marked}} \) such that every clause \( c \in C_{\text{good}} \) is satisfied by its first \( k/20 \) literals corresponding to marked variables.

Moreover, \( \Lambda^* \) can be found in deterministic polynomial time.

**Proof.** We replace every \( c \in C_{\text{good}} \) by its first \( k/20 \) literals corresponding to marked variables. Call the new clause \( c' \) and the new set \( C_{\text{good}}'. \) This induces a new formula \( \Phi' \) whose clause set is \( C_{\text{good}}' \) with no unmarked variables. We apply Theorem 7 to \( \Phi' \). Let \( A_c \) be the event that \( c \) is not satisfied where \( c \in C_{\text{good}}' \) and set \( x(c) = \frac{1}{k\Delta} \) for all \( c \in C_{\text{good}}' \). Since \( \Phi' \) is a smaller formula, the size of \( \Gamma_{\text{good}}(c) \) is still at most \( k(\Delta - 1) \). It is straightforward to verify that the condition (1) holds and Theorem 7 applies. Thus, the desired \( \Lambda^* \) exists.

To find \( \Lambda^* \), once again we apply the deterministic algorithm for the local lemma [7]. \( \square \)

Recall \( \Phi^\Lambda \), which is \( \Phi \) simplified under \( \Lambda \). Under \( \Lambda^* \), neither bad clauses nor bad variables will be removed. More importantly, we have the following corollary.

**Corollary 11.** For any prefix \( \Lambda \) of \( \Lambda^* \) in Theorem 10, any remaining good clause \( c \) in \( \Phi^\Lambda \) satisfies \( \text{marked}(c) \geq k/4 \).

**Proof.** Under \( \Lambda \), for any \( c \in C_{\text{good}} \), either \( c \) has at most \( k/20 \) marked variables assigned, or \( c \) has at most \( k/20 \) marked variables assigned. In the first case, by Lemma 10, \( c \) is satisfied and thus is removed in \( \Phi^\Lambda \). In the second case, even if \( c \) is not satisfied, by Lemma 8, \( c \) has at least \((3/10 - 1/20)k = k/4\) marked variables unassigned. \( \square \)

### 4. The coupling tree

We will use the marking from Lemma 8. Conditioned on a prefix \( \Lambda \) of the partial assignment \( \Lambda^* \) from Lemma 10, the main subroutine of our algorithm is to calculate the marginal probability of the next variable in \( \Lambda^* \), say \( v^* \). Let \( \Omega^\Lambda_1 = \Omega^{\Lambda \cup \{v^* \to T\}} \) and \( \Omega^\Lambda_2 = \Omega^{\Lambda \cup \{v^* \to F\}} \). Then our goal is to estimate \( |\Omega^\Lambda_1|/|\Omega^\Lambda_2| \).

We will eventually set up a linear program to approximate \( |\Omega^\Lambda_1|/|\Omega^\Lambda_2| \). Before introducing the linear program, we will define the so-called coupling tree, which represents a variable-by-variable greedy coupling process between \( \mu^\Omega_1 \) and \( \mu^\Omega_2 \). Denote by \( T^\Lambda \) the coupling tree. Each node \( \rho \) of \( T^\Lambda \) is a tuple \( \rho = (A_1, A_2, V_I, V_{\text{set}}, C_{\text{rem}}, F, R) \) where

- \( V_I \) and \( V_{\text{set}} \) are subsets of \( V \);
- \( C_{\text{rem}} \) and \( F \) are subsets of \( C \);
- \( A_1 \) and \( A_2 \) are functions from \( V_{\text{set}} \) to \( \{T, F\} \);
- \( R \) is a function from \( C \) to the subsets of \( \{\text{bad, disagree, 1, 2}\} \).

We use the notation \( A_1(\rho), A_2(\rho) \), etc. to denote components of \( \rho \). Note that the information in \( \rho \) is redundant, as some component can be deduced from others.

Intuitively, for a node \( \rho \), \( V_{\text{set}} \) is the set of variables that we have tried to couple, and \( V_I \) is the set of variables that cause discrepancies, or for which we have given up. The set \( C_{\text{rem}} \) denotes remaining clauses, and \( F \) denotes “failed” clauses. The two partial assignments \( A_1 \) and \( A_2 \) assign truth values to the variables in \( V_{\text{set}} \). Finally, the function \( R(c) \) gives the reason why \( c \) is in \( F \). The possibilities are: (1) \( c \) is a bad clause; (2) \( A_1 \) and \( A_2 \) disagree on some variable in this clause; (3) \( c \) is not satisfied by the partial assignment \( A_1 \) or \( A_2 \) (or both). These reasons may not be mutually exclusive. If \( c \notin F \), then \( R(c) = \emptyset \).

We will only consider a partial assignment \( \Lambda \) that is a prefix of \( \Lambda^* \) from Lemma 10. Thus, by Corollary 11, in any \( c \in C_{\text{good}}^\Lambda \), there are at least \( k/4 \) marked variables remaining.

We will (inductively) guarantee that every node \( \rho = (A_1, A_2, V_I, V_{\text{set}}, C_{\text{rem}}, F, R) \) of the coupling tree satisfies the following properties:
\(\{v^*\} \subseteq V_{\text{set}} \cap V_I.\)

(P2) Every clause \(c \in C_{\text{rem}}\) satisfies one of the following: \(\text{var}(c) \subseteq V_I, \text{var}(c) \subseteq V^A \setminus V_I,\) or \(\text{marked}(c) \setminus V_{\text{set}}\) is non-empty.

(P3) For any \(c \in C_{\text{rem}} \cap C_{\text{bad}}, \text{var}(c) \subseteq V^A \setminus V_I.\)

(P4) For any \(c \in C^A \setminus C_{\text{rem}},\) at least one of the following is true:

- \(c\) is satisfied by both of the partial assignments \(A_1\) and \(A_2;\)
- \(\text{var}(c) \subseteq V_I(\rho) \cup V_{\text{set}}(\rho).\)

(P5) \(H_\Phi[V_I]\) is connected.

(P6) For any \(u \in V_I, \exists c \in F(\rho)\) s.t. \(u \in \text{var}(c).\)

(P7) For any \(u \in V_{\text{set}} \setminus V_I, A_1(u) = A_2(u).\)

(P8) \(R(c)\) is non-empty if and only if \(c \in F.\)

(P9) For any \(c \in F(\rho),\)

- (P9.1) \(\text{bad} \in R(c)\) iff \(c\) is a bad clause;
- (P9.2) \(\text{disagree} \in R(c)\) iff there is a variable \(v \in \text{var}(c) \cap V_I(\rho) \cap V_{\text{set}}(\rho)\) such that \(A_1(v) \neq A_2(v);\)
- (P9.3) for \(i \in \{1, 2\}, i \in R(c)\) iff the following three conditions hold: \(\text{var}(c) \subseteq V_I(\rho),\)
- \(\text{marked}(c) \subseteq V_{\text{set}}(\rho),\) and \(A_i(\rho)\) does not satisfy \(c.\)

The root of the coupling tree is the node \(\rho^*\) with \(V_{\text{set}}(\rho^*) = V_I(\rho^*) = \{v^*\}, C_{\text{rem}}(\rho^*) = C^A,\) and \(F(\rho^*)\) assigned to the set of clauses containing \(v^*\). The assignment \(A_1(\rho^*)\) sets \(v^*\) to \(T\) and the assignment \(A_2(\rho^*)\) sets \(v^*\) to \(F.\) The function \(R\) maps every clause in \(F(\rho^*)\) to \{\text{disagree}\} and every other clause to \(\emptyset.\)

It is straightforward to see that \(\rho^*\) satisfies the coupling-tree properties. To see property (P2) note that each clause has at least \(k/4 > 1\) marked variables so \(\text{marked}(c) \setminus V_{\text{set}}(\rho)\) is non-empty. Property (P3) follows from the fact that \(v^*\) is good, so it is not in a bad clause (Observation 3). The rest are straightforward.

In order to maintain a polynomial size of the coupling tree, we will set a truncation depth \(L := C_0(3k^2\Delta)[\log(n/\varepsilon)]\) for some sufficiently large absolute constant \(C_0,\) where \(n = |V|\).

**Definition 12** (leaf, truncating node). A node \(\rho\) of the coupling tree is a leaf if \(|V_I(\rho)| \leq L\) and every \(c \in C_{\text{rem}}(\rho)\) has the property that \(\text{var}(c) \subseteq V_I(\rho) \cup V_{\text{set}}(\rho)\) or \(\text{var}(c) \subseteq V^A \setminus (V_I(\rho) \cup V_{\text{set}}(\rho))\).

If \(|V_I(\rho)| > L, then \(\rho\) 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 \(\mathcal{L}^* := \mathcal{L} \cup \mathcal{T}.\)

Suppose that \(\rho\) is not a leaf or a truncating node. Then we define its children as follows. Since \(\rho\) is not a leaf, there is a clause \(c\) such that \(\text{var}(c) \cap V_I(\rho) \neq \emptyset\) and \(\text{var}(c) \cap (V^A \setminus V_I(\rho)) \neq \emptyset.\)

By (P3), \(c\) must be a good clause. Let \(c\) be the first such clause. By (P2), let \(u\) be the first variable in \(\text{marked}(c) \setminus V_{\text{set}}(\rho).\) Since it is in \(\text{marked}(c), u\) is a good variable. We refer to \(c\) as the “first clause of \(\rho\)” and \(u\) as the “first variable of \(\rho.\”

We now define the four children of \(\rho\) in the coupling tree. For each of the four pairs \((\tau_1, \tau_2)\) where \(\tau_1\) and \(\tau_2\) are assignments from \(\{u\}\) to \(\{T, F\},\) we create a child \(\rho_{\tau_1, \tau_2}\) of \(\rho\) using Algorithm 1. The following lemma shows that the coupling tree properties are maintained.

**Lemma 13.** If \(\rho\) satisfies the coupling tree properties, then so does \(\rho_{\tau_1, \tau_2}.\)

**Proof.** Property (P1) holds trivially for all nodes of the coupling tree. Property (P2) holds because clauses not satisfying the conditions are removed in the while loop of Line 15. Property (P3) holds because clauses not satisfying the conditions are removed in the while loop of Line 23. For Property (P4), if a clause is removed from \(C_{\text{rem}},\) then it is removed in Line 13, 21, or 27. The first case satisfies the first condition of (P4), and the other two satisfy the second condition of (P4). For Property (P5), since \(H_\Phi[V_I(\rho)]\) is connected, we just need to verify that when \(V_I\) expands, the new vertices are connected with the old \(V_I.\) The set \(V_I\) expands in Line 6 and Line 20 and 25. All cases can be verified straightforwardly. Property (P6) holds because all variables are added to \(V_I\) in Line 6 and Line 20 and 25. In all three cases the corresponding clauses are added to \(F.\) For Property (P7), \(V_{\text{set}}\) is only expanded in line 1 and the property is guaranteed by line 6. Property (P8) follows from the way that \(R\) and \(F\) are updated by
Properties of the coupling tree for a random formula. The following two properties will be useful. Their proofs are deferred to Section 7.5.

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

Remark. The number of nodes in the coupling tree is a polynomial in $n$ since the depth of the tree does not exceed $\max_{\rho \in \mathbb{T}^\Lambda} |V_{\text{set}}(\rho)| \leq 3k^3\alpha L + 1 = O(\log \frac{n}{\varepsilon})$ and each node has at most 4 children.

According to Definition 12, we truncate the node $\rho$ of the coupling tree if $|V_I(\rho)| > L$; the following lemma shows that $|V_I(\rho)|$ cannot be too large.

Lemma 15. W.h.p. over the choice of $\Phi$, for every truncating node $\rho \in \mathcal{T}$ of $\mathbb{T}^\Lambda$, it holds that $|V_I(\rho)| \leq 359L/358$.

5. The linear program

Before introducing the linear program which we will use to estimate $|\Omega_1^\Lambda|/|\Omega_2^\Lambda|$, we define one more piece of notation. In particular, for each node $\rho$ of the coupling tree we define a quantity $r(\rho)$. 

the algorithm. Properties (P9.1), (P9.2), and (P9.3) can also be verified by going through the algorithm.  

4.1. Properties of the coupling tree for a random formula. The following two properties will be useful. Their proofs are deferred to Section 7.5.

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

Remark. The number of nodes in the coupling tree is a polynomial in $n$ since the depth of the tree does not exceed $\max_{\rho \in \mathbb{T}^\Lambda} |V_{\text{set}}(\rho)| \leq 3k^3\alpha L + 1 = O(\log \frac{n}{\varepsilon})$ and each node has at most 4 children.

According to Definition 12, we truncate the node $\rho$ of the coupling tree if $|V_I(\rho)| > L$; the following lemma shows that $|V_I(\rho)|$ cannot be too large.

Lemma 15. W.h.p. over the choice of $\Phi$, for every truncating node $\rho \in \mathcal{T}$ of $\mathbb{T}^\Lambda$, it holds that $|V_I(\rho)| \leq 359L/358$.
Let \( \rho \) be a node of the coupling tree. Let \( C_I(\rho) \) be the set of clauses \( c \in C^A \) such that \( \text{var}(c) \subseteq V_I(\rho) \cup V_{\text{set}}(\rho) \). For \( i \in \{1, 2\} \), let \( N_i(\rho) \) be the number of assignments \( \tau \) to \( V_I(\rho) \setminus V_{\text{set}}(\rho) \) such that every clause in \( C_I(\rho) \) is satisfied by \( \tau \cup A_i(\rho) \).

We will use the following lemma.

**Lemma 16.** If \( \rho \) is a node in the coupling tree, then \( N_i(\rho) \neq 0 \) for any \( i \in \{1, 2\} \).

**Proof.** Since our coupling tree is based on the marking from Lemma 8, there is a partial assignment of bad variables that satisfies all \( c \in C_{\text{bad}} \). Let \( \tau_{\text{bad}} \) be such an assignment and let \( \tau_{\text{bad}}(\rho) \) be the restriction of \( \tau_{\text{bad}} \) to the set \( (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V_{\text{bad}} \). Note that \( V_{\text{set}}(\rho) \subseteq V^A_{\text{good}} \), so (by Observation 3) any bad clause \( c \in C_I(\rho) \) has the property that \( \text{var}(c) \subseteq (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V_{\text{bad}} \), which implies that \( \tau_{\text{bad}}(\rho) \) satisfies all bad clauses in \( C_I(\rho) \).

Next we claim that there is an partial assignment \( \tau_{\text{good}}(\rho) : (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V^A_{\text{good}} \rightarrow \{T, F\} \) such that \( \tau_{\text{good}}(\rho) \) satisfies all good clauses in \( C_I(\rho) \). Let \( c \) be a good clause in \( C_I(\rho) \). Again by Lemma 8, \( c \) has at least \( k/4 \) unmarked good variables. Note that \( \text{var}(c) \subseteq V_I(\rho) \cup V_{\text{set}}(\rho) \) and \( V_{\text{set}}(\rho) \) consists only of marked variables, so \( \text{var}(c) \cap (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V^A_{\text{good}} \geq k/4 \). Denote by \( A_c \) the set of assignments in \( \Omega^* \) of which the restriction on \( (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V^A_{\text{good}} \) do not satisfy \( c \). Thus, \( \text{Pr}_{\mu_{\Omega} \cap \{A_c\}} \leq 2^{-k/4} \). Also, by the definition of \( A_c \), we obtain that \( \text{var}(A_c) \subseteq \text{var}(c) \) and \( \text{var}(A_c) \subseteq V^A_{\text{good}} \). So \( \Gamma(\{A_c\} \subseteq \Gamma_{\text{good}}(c) \). Let \( x(c) = \frac{1}{\sqrt{k} \Delta} \). Since \( \Gamma_{\text{good}}(c) \) has the property that \( \delta_{\text{rem}}(\rho, (V_I(\rho) \setminus V_{\text{set}}(\rho)) \cap V^A_{\text{good}} \geq k/4 \), we conclude our claim by applying Theorem 7 and verifying

\[
\begin{align*}
x(c) \prod_{b \in \Gamma(A_c)} \left(1 - x(b)\right) & \geq \frac{1}{k \Delta} \left(1 - \frac{1}{k \Delta}\right)^{k \Delta} \geq \frac{1}{e k \Delta} > 2^{-k/4} = \text{Pr}_{\mu_{\Omega} \cap \{A_c\}}(A_c).
\end{align*}
\]

Now let \( \tau = \tau_{\text{good}}(\rho) \cup \tau_{\text{bad}}(\rho) \). Then every clause in \( C_I(\rho) \) is satisfied by \( \tau \), which yields that \( N_i(\rho) > 0 \).

Define \( r(\rho) := N_1(\rho)/N_2(\rho) \). Lemma 16 implies that \( r(\rho) \) is always well-defined.

**Observation 17.** The quantity \( r(\rho) \) can be computed in \( m \cdot 2^{O(|V_I(\rho) \setminus V_{\text{set}}(\rho)|)} \) time by considering all assignments \( \tau \) to \( V_I(\rho) \setminus V_{\text{set}}(\rho) \). By Lemma 15, the time taken is polynomial in \( n/\varepsilon \).

The importance of \( r(\rho) \) comes from the following lemma.

**Lemma 18.** If \( \rho \) is a leaf, then \( r(\rho) = |\Omega^{A_1(\rho) \cup A_2(\rho)}|/|\Omega^{A_2(\rho) \cup A_1(\rho)}| \).

**Proof.** Suppose \( \rho = (A_1, A_2, V_I, V_{\text{set}}, C_{\text{rem}}, F, R) \). Since \( \rho \) is a leaf, by Definition 12, for any \( c \in C_{\text{rem}} \), either \( \text{var}(c) \subseteq V_I \cup V_{\text{set}} \) (i.e., \( c \in C_I(\rho) \)), or \( \text{var}(c) \subseteq V^A \setminus (V_I \cup V_{\text{set}}) \). Denote by \( C_1(\rho) \) the set of clauses in \( C_{\text{rem}} \) such that \( \text{var}(c) \subseteq V^A \setminus (V_I \cup V_{\text{set}}) \). Denote by \( C_{\text{other}}(\rho) \) the set of clauses in \( C^A \setminus (C_{\text{rem}} \cup C_I(\rho)) \). Thus, the clauses in \( C^A \) split into the disjoint sets \( C_I(\rho), C_{\text{other}}(\rho) \) and \( C_{\text{other}}(\rho) \).

By (P4), every clause in \( C_{\text{other}}(\rho) \) is satisfied by \( A_1(\rho) \) and \( A_2(\rho) \).

Let \( M(\rho) \) be the number of assignments \( \sigma : V^A \setminus (V_I \cup V_{\text{set}}) \rightarrow \{T, F\} \) such that all clauses in \( C_{\text{other}} \) are satisfied by \( \sigma \). Recall that \( N_i(\rho) \) is the number of assignments \( \tau \) to \( V_I(\rho) \setminus V_{\text{set}}(\rho) \) such that every clause in \( C_I(\rho) \) is satisfied by \( \tau \cup A_i(\rho) \). Then for \( i \in \{1, 2\}, |\Omega^{A_i(\rho) \cup A_{3-i}(\rho)}| = M(\rho)N_i(\rho) \), which implies the lemma, since \( r(\rho) = N_1(\rho)/N_2(\rho) \).

We will use a binary search to approximate the quantity \( |\Omega_{\rho}^1|/|\Omega_{\rho}^2| \). Our linear program relies on two constants \( r_{\text{lower}} \) and \( r_{\text{upper}} \). We will move these closer and closer together by binary search. For each node \( \rho \) of the coupling tree, we introduce two variables \( P_{1, \rho} \) and \( P_{2, \rho} \). The idea is that a solution of the LP should have the property that

\[
\frac{|\Omega_{\rho}^1|}{|\Omega_{\rho}^2|} = \frac{P_{1, \rho}}{P_{2, \rho}} \cdot \frac{|\Omega^{A_1(\rho) \cup A_2(\rho)}|}{|\Omega^{A_2(\rho) \cup A_1(\rho)}|}.
\]

We now introduce the constraint sets of the LP.
Constraint Set 0. For every node $\rho$ of the coupling tree and every $i \in \{1, 2\}$ we add the constraint $0 \leq P_{i, \rho} \leq 1$.

Constraint Set 1. If $\rho \in \mathcal{L}$ then we add the following constraints.

$$r_{\text{lower}} P_{2, \rho} \leq P_{1, \rho} r(\rho)$$
$$P_{1, \rho} r(\rho) \leq r_{\text{upper}} P_{2, \rho}$$

Remark. The purpose of these constraints is to guarantee

$$r_{\text{lower}} \leq \frac{P_{1, \rho}}{P_{2, \rho}} r(\rho) \leq r_{\text{upper}}.$$

Constraint Set 2. For the root $\rho^*$ of the coupling tree, we add the constraints

$$P_{1, \rho^*} = P_{2, \rho^*} = 1.$$

For every node $\rho$ of the coupling tree that is not in $\mathcal{L}^*$, let $u$ be the first variable of $\rho$. Add constraints as follows. For each $X \in \{T, F\}$ add the following constraints.

$$P_{1, \rho} = P_{1, \rho_u \rightarrow X, u \rightarrow T} + P_{1, \rho_u \rightarrow X, u \rightarrow F}$$
$$P_{2, \rho} = P_{2, \rho_u \rightarrow X, u \rightarrow T} + P_{2, \rho_u \rightarrow F, u \rightarrow X}$$

These constraints imply the following lemma.

Lemma 19. Suppose that the LP variables satisfy all of the constraints in Constraint Set 2. Then for any $i \in \{1, 2\}$ and any $\sigma \in \Omega_i^A$,

$$\sum_{\rho \in \mathcal{L}^* : \sigma \in \Omega_{A_i(\rho)} \cup A_i} P_{i, \rho} = 1$$

Proof. For $i \in \{1, 2\}$, we will maintain a set $\Psi_i$ of nodes in the coupling tree with the invariant that $\sum_{\rho \in \Psi_i} P_{i, \rho} = 1$ and every node $\rho \in \Psi_i$ has $A_i(\rho)$ agree with $\sigma$. For the base case, we let $\rho^*$ be the root of the coupling tree and we take $\Psi_i = \{\rho^*\}$. If every node in $\Psi_i$ is in $\mathcal{L}^*$ then we are finished. Otherwise, let $\rho$ be some node in $\Psi_i$ that is not in $\mathcal{L}^*$. Let $u$ be the first variable of $\rho$. Let $\rho'$ and $\rho''$ be the two children of $\rho$ such that $A_i(\rho')$ and $A_i(\rho'')$ both map $u$ to $\sigma(u)$. Replace $\rho$ in $\Psi_i$ with $\rho'$ and $\rho''$. The constraints guarantee that $P_{i, \rho} = P_{i, \rho'} + P_{i, \rho''}$, so the invariant is maintained. \hfill \Box

Constraint Set 3. For every node $\rho$ of the coupling tree that is not in $\mathcal{L}^*$, every $X \in \{T, F\}$, and every $i \in \{1, 2\}$, let $u$ be the first variable of $\rho$ and add the constraint $P_{i, \rho_u \rightarrow X, u \rightarrow X} \leq \frac{1}{8} P_{i, \rho}$.

The intuition behind this set of constraints is Lemma 9.

6. Analysis of the Linear Program

In Section 6.1, we show that whenever $r_{\text{lower}} \leq \frac{\mid \Omega_A^1 \mid / \mid \Omega_A^2 \mid}{\mid \Omega^A_A \mid}$, a solution to the LP exists. We call this “completeness” of the LP. In the remaining subsections of this section, we show “soundness” — namely, that whenever a solution exists, $r_{\text{lower}}$ and $r_{\text{upper}}$ are valid bounds for the quantity $\frac{\mid \Omega_A^1 \mid / \mid \Omega_A^2 \mid}$, up to small errors.

6.1. Complementeness. Recall that we use the marking from Lemma 8 and that $\Lambda$ is a prefix of the partial assignment $\Lambda^*$ from Lemma 10.

We use the following lemma.

Lemma 20. If $\rho$ is a node in the coupling tree then, for any $i \in \{1, 2\}$, $\Omega_{A_i(\rho) \cup \Lambda}$ is non-empty.

Proof. Let $\rho$ be a node in the coupling tree and fix $i \in \{1, 2\}$. Since our coupling tree is based on the marking from Lemma 8, there is a partial assignment of bad variables that satisfies all $c \in \mathcal{C}_{\text{bad}}$. Let $\tau_{\text{bad}}$ be such an assignment. Note that $A_i(\rho) \cup \Lambda$ is a partial assignment of marked (good) variables so it does not assign any variables in common with $\tau_{\text{bad}}$. Let $V'$ be the set of unmarked good variables. It remains to show that there is a partial assignment of variables in $V'$ that satisfies all clauses in $\mathcal{C}_{\text{good}}$. To do this we apply Theorem 7 to $V'_{\text{good}}$ and $\mathcal{C}_{\text{good}}$ in the same was as the proof of Lemma 9. \hfill \Box
Let $\rho$ be a node of the coupling tree with first variable $u$. For $X \in \{T, F\}$, we use the notation

$$
\psi_{\rho,X,1} := \frac{\Omega^A_1(\rho, u \rightarrow X, u \rightarrow X)}{\Omega^A_1(\rho) \cup \Lambda} = \frac{\Omega^A_1(\rho, u \rightarrow X, u \rightarrow X)}{\Omega^A_1(\rho, u \rightarrow X)}.
$$

This is well-defined since $A_1(\rho, u \rightarrow X, u \rightarrow X) = A_1(\rho, u \rightarrow X, u \rightarrow X)$. In other words, $\psi_{\rho,X,1}$ is the probability that $u$ is assigned value $X$ under $\mu_{\Omega^A_1(\rho) \cup \Lambda}$. Thus, $\psi_{\rho,X,1} + \psi_{\rho,\neg X,1} = 1$. We similarly define

$$
\psi_{\rho,X,2} := \frac{\Omega^A_2(\rho, u \rightarrow X, u \rightarrow X)}{\Omega^A_2(\rho) \cup \Lambda} = \frac{\Omega^A_2(\rho, u \rightarrow X, u \rightarrow X)}{\Omega^A_2(\rho, u \rightarrow X)},
$$

noting that $\psi_{\rho,X,2} + \psi_{\rho,\neg X,2} = 1$.

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 $L^*$. For each node $\rho$, $Q(\rho)$ will be the probability that $\rho$ 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 $\rho$ that is not in $L^*$ we can define $Q(\cdot)$ on the children of $\rho$ as follows. Let $u$ be the first variable of $\rho$ and consider the four children $\rho_u \rightarrow T, u \rightarrow T, \rho_u \rightarrow F, u \rightarrow F$. Define the values of $Q$ as follows.

$$
Q(\rho_u \rightarrow T, u \rightarrow T) := Q(\rho) \min\{\psi_{\rho,T,1}, \psi_{\rho,T,2}\},
Q(\rho_u \rightarrow T, u \rightarrow F) := Q(\rho) \min\{\psi_{\rho,T,1}, 1 - \psi_{\rho,T,2}\},
Q(\rho_u \rightarrow F, u \rightarrow T) := Q(\rho) \min\{1 - \psi_{\rho,T,1}, 1 - \psi_{\rho,T,2}\},
Q(\rho_u \rightarrow F, u \rightarrow F) := Q(\rho) \min\{1 - \psi_{\rho,T,1}, 1 - \psi_{\rho,T,2}\}.
$$

Equation (5)

**Observation 21.** For any $X \in \{T, F\}$,

$$
Q(\rho_u \rightarrow X, u \rightarrow T) + Q(\rho_u \rightarrow X, u \rightarrow F) = Q(\rho) \psi_{\rho,X,1},
$$

and

$$
Q(\rho_u \rightarrow T, u \rightarrow X) + Q(\rho_u \rightarrow F, u \rightarrow X) = Q(\rho) \psi_{\rho,X,2}.
$$

Also,

$$
Q(\rho_u \rightarrow T, u \rightarrow T) + Q(\rho_u \rightarrow T, u \rightarrow F) + Q(\rho_u \rightarrow F, u \rightarrow T) + Q(\rho_u \rightarrow F, u \rightarrow F) = Q(\rho).
$$

This implies that $\sum_{\rho \in L^*} Q(\rho) = 1$.

We use $Q$ to define a feasible solution to the LP. Indeed, this definition explains what the variables in the LP were meant to represent.

**Definition 22** (The LP variables). For each node $\rho$ of the coupling tree and each $i \in \{1, 2\}$, define $P_{\rho,i} := Q(\rho) \Omega^A_i / |\Omega^A_i| \cup \Lambda|.$

Theorem 20 ensures that the values in Definition 22 are well defined. Motivated by Definition 22, we give the following upper bound for $Q(\rho)$.

**Lemma 23.** For all nodes $\rho$ in the coupling tree and all $i \in \{1, 2\}$, $Q(\rho) \leq |\Omega^A_i(\rho) \cup \Lambda| / |\Omega^A_i|.$

**Proof.** The proof is by induction — having established the claim for a node $\rho$, we then establish it for the children of $\rho$ using Observation 21. For the base case, $Q(\rho^*) = 1 = |\Omega^A_i(\rho^*) \cup \Lambda| / |\Omega^A_i|.$ For the inductive step, consider a node $\rho$ (for which the claim is established) and let $u$ be the first variable of $\rho$. Consider any child $\rho_u \rightarrow X, u \rightarrow Y$ of $\rho$. We show the lemma for $i = 1$ and the other case is similar. By Observation 21, Equation (5), and the induction hypothesis,

$$
Q(\rho_u \rightarrow X, u \rightarrow Y) \leq Q(\rho) \cdot \psi_{\rho,X,1} = Q(\rho) \cdot \frac{\Omega^A_1(\rho_u \rightarrow X, u \rightarrow Y) \cup \Lambda}{\Omega^A_1(\rho) \cup \Lambda} \leq \frac{\Omega^A_i(\rho) \cup \Lambda}{\Omega^A_i} \cdot \frac{\Omega^A_1(\rho_u \rightarrow X, u \rightarrow Y) \cup \Lambda}{\Omega^A_1(\rho) \cup \Lambda} = \frac{\Omega^A_1(\rho_u \rightarrow X, u \rightarrow Y) \cup \Lambda}{\Omega^A_i} \cdot \frac{\Omega^A_i(\rho) \cup \Lambda}{\Omega^A_i}.
$$

The next lemma relates the values of the LP variables, as defined by Definition 22.
Lemma 24. Let $\rho \notin \mathcal{L}^*$ be a node in the coupling tree. Let $u$ be the first variable of $\rho$. For any $X \in \{T,F\}$ and $i \in \{1,2\}$, $P_{i,\rho \to X, u \to \neg X} \leq P_{i,\rho} / s$.

Proof. We first assume $X = T$ and $i = 1$. Definition 22 implies that

$$
\frac{P_{i,\rho \to T, u \to \neg T}}{P_{i,\rho}} = \frac{Q(\rho \to T, u \to \neg T)}{Q(\rho)} \cdot \frac{|\Omega^{A_1(\rho) \cup \Lambda}|}{|\Omega^{A_1(\rho \to T, u \to \neg T) \cup \Lambda}|}.
$$

From the definition (6) (for the case where $\psi_{\rho, T, 2} \leq \psi_{\rho, T, 1}$) along with Observation 21 (for the other case), we have that

$$
\frac{Q(\rho \to T, u \to \neg T)}{Q(\rho)} \leq |\psi_{\rho, T, 1} - \psi_{\rho, T, 2}|.
$$

Recall that $\psi_{\rho, T, i}$ is the probability that $u$ is assigned the value $T$ under $\mu_{\Omega^{A_1(\rho) \cup \Lambda}}$. By Lemma 20, $\Omega^{A_1(\rho) \cup \Lambda}$ is non empty, so we can apply Lemma 9 to this partial assignment. From the second part of the lemma, we have

$$
\frac{1}{2} \left(1 - \frac{1}{3s}\right) \leq \psi_{\rho, T, i} \leq \frac{1}{2} \left(1 + \frac{1}{3s}\right).
$$

Since $k$ is sufficiently large and so is $s$,

$$
|\psi_{\rho, T, 1} - \psi_{\rho, T, 2}| \leq \frac{1}{s} \cdot \psi_{\rho, T, i}.
$$

The claim in the lemma follows for $X = T$ and $i = 1$ since $\psi_{\rho, T, 1} = \frac{|\Omega^{A_1(\rho \to T, u \to \neg T) \cup \Lambda}|}{|\Omega^{A_1(\rho) \cup \Lambda}|}$.

For $X = F$ and $i = 1$, note that

$$
\frac{Q(\rho \to F, u \to T)}{Q(\rho)} \leq |(1 - \psi_{\rho, T, 1}) - (1 - \psi_{\rho, T, 2})| = |\psi_{\rho, T, 1} - \psi_{\rho, T, 2}|,
$$

and use $\psi_{\rho, F, 1} = \frac{|\Omega^{A_1(\rho \to F, u \to T) \cup \Lambda}|}{|\Omega^{A_1(\rho) \cup \Lambda}|}$ in the end.

For $i = 2$, the proof is similar. \(\square\)

Now we are ready to show the completeness.

Lemma 25. Suppose $r_{\text{lower}} \leq |\Omega_1^{A_1}| / |\Omega_2^{A_2}| \leq r_{\text{upper}}$. The variables $P = \{P_{i,\rho}\}$ defined in Definition 22 satisfy all constraints of the LP.

Proof. Constraint Set 0: The fact that the LP variables satisfy these constraints follows from the definition of the variables in Definition 22 (which guarantees that they are all non-negative) and from Lemma 23.

Constraint Set 1: Definition 22 implies that for any node $\rho$ in the coupling tree,

$$
\frac{P_{1,\rho}}{P_{2,\rho}} \frac{|\Omega^{A_1(\rho) \cup \Lambda}|}{|\Omega^{A_2(\rho) \cup \Lambda}|} = \frac{|\Omega_1^{A_1}|}{|\Omega_2^{A_2}|}.
$$

If $\rho$ is a leaf, then by Lemma 18, $r(\rho) = |\Omega^{A_1(\rho) \cup \Lambda}| / |\Omega^{A_2(\rho) \cup \Lambda}|$. Equation (8) implies that

$$
\frac{P_{1,\rho}}{P_{2,\rho}} r(\rho) = \frac{|\Omega_1^{A_1}|}{|\Omega_2^{A_2}|},
$$

so as long as $r_{\text{lower}} \leq |\Omega_1^{A_1}| / |\Omega_2^{A_2}| \leq r_{\text{upper}}$, the LP variables satisfy the constraints in Constraint Set 1, as required.

Constraint Set 2: For the root $\rho^*$ of the coupling tree, it is easy to see from Definition 22 that $P_{1,\rho^*} = 1$.\(\square\)
Let \( \rho \notin \mathcal{L}^* \) be a node in the coupling tree and let \( u \) be the first variable of \( \rho \). For \( X \in \{ T, F \} \) and \( i = 1 \) we wish to establish \( P_{1,\rho} = P_{1,\rho_u \rightarrow X, u \rightarrow T} + P_{1,\rho_u \rightarrow X, u \rightarrow F} \). Plugging in Definition 22 and dividing by \( |\Omega_A^1| \), the constraint is equivalent (for any \( Y \in \{ T, F \} \)) to

\[
\frac{Q(\rho)}{|\Omega_A^1(\rho)^{\Lambda^*}|} = \frac{Q(\rho_{u \rightarrow X, u \rightarrow T})}{|\Omega_A^1(\rho_{u \rightarrow X, u \rightarrow T})^{\Lambda^*}|} + \frac{Q(\rho_{u \rightarrow X, u \rightarrow F})}{|\Omega_A^1(\rho_{u \rightarrow X, u \rightarrow F})^{\Lambda^*}|} = \frac{Q(\rho_{u \rightarrow X, u \rightarrow T})}{|\Omega_A^1(\rho_{u \rightarrow X, u \rightarrow T})^{\Lambda^*}|} + \frac{Q(\rho_{u \rightarrow X, u \rightarrow F})}{|\Omega_A^1(\rho_{u \rightarrow X, u \rightarrow F})^{\Lambda^*}|},
\]

where we used again the fact that \( A_1(\rho_{u \rightarrow X, u \rightarrow X}) = A_1(\rho_{u \rightarrow X, u \rightarrow X}) \). The equation above follows from Observation 21 using (5). The other three constraints are similar.

**Constraint Set 3:** This case directly follows from Lemma 24. \( \Box \)

### 6.2. \( \ell \)-wrong assignments

There are two kinds of errors which cause solutions of the LP to differ from the ratio \( |\Omega_A^1|/|\Omega_A^1| \). The first kind of error involves a notion that we call “\( \ell \)-wrong assignments”. To define them, we need some graph-theoretic notation.

**Definition 26.** Given a graph \( G \) with vertices \( u \) and \( v \) in \( V(G) \), let \( \text{dist}_G(u, v) \) be the distance between \( u \) and \( v \) in \( G \) — that is, the number of edges in a shortest path from \( u \) to \( v \). Given a subset \( T \subseteq V(G) \) and a vertex \( v \in V(G) \), let \( \text{dist}_G(u, T) = \min_{v \in T} \text{dist}_G(u, v) \). For any positive integer \( k \), let \( G^k \) be the graph with vertex set \( V(G) \) in which vertices \( u \) and \( v \) are connected if and only if there is a length-\( k \) path from \( u \) to \( v \) in \( G \). Let \( G^k \) be the graph with vertex set \( V(G) \) in which vertices \( u \) and \( v \) are connected if and only if 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 [29, 20]. The main difference between our definition and previous ones is that we take into account whether clauses are connected via good variables.

**Definition 27.** Given the graph \( G_\Phi \), let \( \mathcal{D}(G_\Phi) \) be the set of subsets \( T \subseteq V(G_\Phi) \) such that the following hold:

1. For any \( c_1, c_2 \in T \), \( \text{var}(c_1) \cap \text{var}(c_2) \cap V_{\text{good}} = \emptyset \);
2. The graph \( G_\Phi^4[T] \), which is the subgraph of \( G_\Phi^4 \) induced by \( T \), is connected.

The following two lemmas regarding \( \mathcal{D}(G_\Phi) \) will be useful. We defer their proofs to Section 7.6.

**Lemma 28.** Let \( \ell \) be an integer which is at least \( \log n \). W.h.p. over the choice of \( \Phi \), every clause \( c \in c^\Lambda_{\text{good}} \) has the property that the number of size-\( \ell \) subsets \( T \in \mathcal{D}(G_\Phi) \) containing \( c \) is at most \( (18k^2\alpha)^{4\ell} \).

Recall that the coupling tree \( T^\Lambda \) is defined with respect to a prefix \( \Lambda \) of the partial assignment \( \Lambda^* \) from Lemma 10. Let \( c^* \) be the first clause of the root node \( \rho^* \).

**Lemma 29.** W.h.p. over the choice of \( \Phi \), every node \( \rho \) in \( T^\Lambda \) with \( |V_1(\rho)| \geq L \) has the property that there is a set \( T \subseteq F(\rho) \) containing \( c^* \) such that \( T \in \mathcal{D}(G_\Phi) \), \( |T| = C_0[\log(n/\varepsilon)] \) and \( |T \cap C_{\text{bad}}| \leq |T|/3 \).

We now define \( \ell \)-wrong assignments.

**Definition 30.** An assignment \( \sigma \in \Omega_1^\Lambda \) is \( \ell \)-wrong if there is a size-\( \ell \) set \( T \in \mathcal{D}(G_\Phi) \) such that

- \( c^* \in T \),
- \( |T \cap C_{\text{good}}^\Lambda| \geq 2 |T|/3 \), and
- there is a size \( \lceil \ell/2 \rceil \) subset \( S \) of \( T \cap C_{\text{good}}^\Lambda \) such that the restriction of \( \sigma \) to marked variables in clauses in \( S \) does not satisfy any clause in \( S \). (Formally, taking \( U \) to be \( \cup_{c \in \text{marked}(\sigma)} c \), the condition is that \( \sigma[U] \) does not satisfy any clauses in \( S \).

Otherwise \( \sigma \) is \( \ell \)-correct.

The following is similar to [20, Lemma 4.8].
Lemma 31. Fix $i \in \{1, 2\}$. Let $\ell = L/(3k^2\Delta)$. Then the fraction of assignments in $\Omega^i_\Lambda$ that are $\ell$-wrong is at most $(k\Delta)^{-9\ell}$.

Proof. Assume $i = 1$, as the other case is similar. We want to show that

$$\Pr_{\mu_{\Omega^1}}(\sigma \text{ is } \ell\text{-wrong}) \leq (k\Delta)^{-9\ell}.$$ 

If every assignment in $\mu_{\Omega^1}$ is $\ell$-correct, then we are finished. Otherwise, consider a size-$\ell$ set $T = \{c_1, c_2, \ldots, c_\ell\}$ in $\mathcal{D}(G_\Phi)$ such that $c_\ell = e^*$ and $\ell_T \geq 2\ell/3$ where $\ell_T := |T \cap \mathcal{C}^\text{good}|.$

Let $V := \mathcal{V}^\Lambda \setminus \{v^*\}$ be the set of unsatisfied variables in $\rho^*$, and let $\Omega'$ be the set of all assignments $\sigma : V \rightarrow \{T, F\}$. Let $U_T := V \cap (\bigcup_{i=1}^{\ell_T} \text{marked}(c_i))$ be the set of all marked variables in $V$ that are in clauses in $T$. Given an assignment $\sigma \in \Omega'$, let $Z_T(\sigma)$ be the number of clauses in $T \cap \mathcal{C}^\text{good}$ that are not satisfied by $\sigma[U_T]$.

Suppose that $\sigma$ is drawn from $\mu_{\Omega^1}$. By Corollary 11, every $c \in \mathcal{C}^\text{good}$ has at least $k/4 - 1$ marked variables in $V$. Thus, for every $c \in T \cap \mathcal{C}^\text{good}$, the probability that $c$ is not satisfied by $\sigma[U_T]$ is at most $p^* := 2^{1-k/4}$. For $c \in T \cap \mathcal{C}^\text{good}$, let $A_c$ be the event that $c$ is not satisfied by $\sigma[U_T]$. Because $T \in \mathcal{D}(G_\Phi)$, clauses in $T$ do not share good variables (so they do not share marked variables). Thus, the events $A_c$ are mutually independent. Since $\ell_T = |T \cap \mathcal{C}^\text{good}|$, we have that the event $Z_T(\sigma) \geq \lceil \ell/2 \rceil$ is dominated above by the probability that $\hat{Z} \geq \lceil \ell/2 \rceil$ where $\hat{Z}$ is a sum of $\ell_T$ independent Bernoulli r.v.s with parameter $p^*$. Setting $\gamma := \ell/(2p^* \ell_T) - 1 \geq 1/(2p^*) - 1$ and applying a Chernoff bound to $\hat{Z}$, we obtain that

$$\Pr_{\mu_{\Omega^1}}(Z_T(\sigma) \geq \lceil \ell/2 \rceil) = \Pr_{\mu_{\Omega^1}}(Z_T(\sigma) \geq \ell/2) = \Pr_{\mu_{\Omega^1}}(Z_T(\sigma) \geq (1 + \gamma)p^* \ell_T) \leq \Pr(\hat{Z} \geq (1 + \gamma)p^* \ell_T) \leq \left(\frac{e^{\gamma}}{(1 + \gamma)^{1+\gamma}}\right)^{p^* \ell_T} \leq \left(2e^{p^*}\right)^{\ell_T/2} \leq \left(2e^{p^*}\right)^{\ell_T/3},$$

where the second-to-last inequality follows by substituting in the lower bound $1/(2p^*) - 1$ for $\gamma$.

Let $\mathcal{E}_T$ be the event that $Z_T(\sigma) \geq \lceil \ell/2 \rceil$, so $\text{var}(\mathcal{E}_T) = U_T$. We apply Lemma 9 with the partial assignment $\Lambda \cup \{v^* \rightarrow T\}$ and with the event $A$ from Lemma 9 being $\mathcal{E}_T$. A clause $c$ is in $\Gamma(\mathcal{E}_T)$ if and only if $\text{var}(c)$ intersects $U_T$. Since all variables in $U_T$ are marked, each $c_i \in T \cap \mathcal{C}^\text{good}$ has at most $3k/4$ variables in $U_T$. Moreover, for each $v \in U_T$, there are at most $\Delta$ clauses containing $v$. Thus, $|\Gamma(\mathcal{E}_T)| \leq 3k\ell \Delta/4$.

Plugging everything into Lemma 9, we obtain

$$\Pr_{\mu_{\Omega^1}}(\mathcal{E}_T) \leq \Pr_{\mu_{\Omega^1}}(\mathcal{E}_T) \left(1 - \frac{1}{sk\Delta}\right)^{-3k\ell \Delta/4} \leq \left(2e^{p^*}\right)^{\ell_T/3} e^{\ell/4s},$$

where the value $s$ (defined just before Lemma 9) is $s = 2k^{1/4}/(ek\Delta)$.

Note from the definition of $L$ that $\ell = C_0[\log(n/\epsilon)]$ for a sufficiently large constant $C_0$, so $\ell \geq \log n$. Applying Lemma 28 to the good clause $c^*$, w.h.p. over the choice of $\Phi$, the number of size-$\ell$ sets $T \in \mathcal{D}(G_\Phi)$ containing $c^*$ is at most $(18k^2\alpha)^{4\ell}$. Hence, by a union bound,

$$\Pr_{\mu_{\Omega^1}}(\sigma \text{ is } \ell\text{-wrong}) \leq \sum_{T \in \mathcal{D}(G_\Phi) : |T| = \ell, c^* \in T, \ell_T \geq 2\ell/3} \Pr_{\mu_{\Omega^1}}(\mathcal{E}_T) \leq (18k^2\alpha)^{4\ell} \left(2e^{p^*}\right)^{\ell_T/3} e^{\ell/4s} \leq (18^4k^8\alpha^32^{1/3}e^{1/3}2^{1/3}2^{-k/2}e^{1/8})^\ell \leq (k\Delta)^{-9\ell},$$

where in the last inequality we used that $\alpha \leq 2^{k/100}$ and $\Delta = 2^{k/300}$. \hfill \Box

6.3. Errors from truncating nodes. We have already noted that the existence of $\ell$-wrong assignments can be viewed as “errors” which cause solutions of the LP to differ from the ratio $|\Omega^1_\Lambda|/|\Omega^2_\Lambda|$. However, even if an assignment $\sigma \in \Omega^1_\Lambda$ is $\ell$-correct, it may still induce some error
in the solution of the LP due to the truncation of the coupling tree. We account for this kind of error in this section.

Fix an assignment \( \sigma \in \Omega_1^A \). Given a solution \( P = \{P_{1,\rho}\} \) to the LP, we consider the following stochastic process, which is a process (depending on \( \sigma \)) for sampling from \( L^* \). The process is defined for \( i = 1 \) rather than for \( i = 2 \). That is it starts with \( \sigma \in \Omega_1^A \) and it uses the LP values \( P_{1,\rho} \). (We will later require a similar process for \( i = 2 \).

**Definition 32** (Sampling conditioned on \( \sigma \in \Omega_1^A \)). Fix \( \sigma \in \Omega_1^A \). Here is a method for choosing a node \( \rho \in L^* \). Start by setting \( \rho \) to be the root \( \rho^* \) of the coupling tree. From every node \( \rho \) that is not in \( L^* \), proceed to a child, as follows. Let \( u \) be the first variable of \( \rho \). For each \( X \in \{T,F\} \), go to \( \rho_{u \rightarrow \sigma(u),u \rightarrow X} \) with probability \( P_{1,\rho_{u \rightarrow \sigma(u),u \rightarrow X}/P_{1,\rho}} \).

This process is well-defined because \( P \) satisfies **Constraint Set 2**. If \( \rho \) is a leaf such that \( A_1(\rho) \) agrees with \( \sigma \), then the probability of reaching \( \rho \) is \( P_{1,\rho} \). The following definition concerns the truncating nodes \( \rho \in T \) where the process can also stop.

**Lemma 33.** Fix \( r_{\text{lower}} \leq r_{\text{upper}} \). Let \( \ell = L/(3k^2 \Delta) \). W.h.p. over the choice of \( \Phi \), the following holds. Let \( \sigma \) be any \( \ell \)-correct assignment in \( \Omega_1^A \). If the LP has a solution using \( r_{\text{lower}} \) and \( r_{\text{upper}} \) then \( \sum_{\rho \in T; \sigma \in \Omega^{A_1(\rho)} \cup \Lambda} P_{1,\rho} \leq (k \Delta)^{-8\ell} \).

**Proof.** Let \( \Upsilon_\sigma \) be the distribution on nodes in \( L^* \) from the sampling procedure in Definition 32. Let \( \Upsilon_\sigma = \{\rho \in T \mid \sigma \in \Omega^{A_1(\rho)} \cup \Lambda\} \). The probability that the sampling procedure reaches any node \( \rho \in L^* \) with \( \sigma \in \Omega^{A_1(\rho)} \cup \Lambda \) is \( P_{1,\rho} \), so the probability that it reaches \( \Upsilon_\sigma \) is \( \sum_{\rho \in \Upsilon_\sigma} P_{1,\rho} \), which is what we want to bound.

From the definition of \( L \), note that \( \ell = C_0 \left( \log(n/\varepsilon) \right) \) for a sufficiently large constant \( C_0 \). Suppose that the formula \( \Phi \) satisfies the condition in Lemma 29 (which it does, w.h.p.). Any node \( \rho \in \Upsilon_\sigma \) satisfies \( |V_1(\rho)| > L \), so by Lemma 29, there is a set \( T \subseteq F(\rho) \) containing \( c^* \) such that \( T \in D(G_\Phi) \), \( |T| = \ell \) and \( |T \cap \Lambda| \leq |T|/3 \). This implies that

\[
|T \cap \Lambda_{\text{good}}| \geq \frac{2|T|}{3}.
\]

Let \( W \) be the set of all size-\( \ell \) sets \( T \in D(G_\Phi) \) containing \( c^* \) such that (9) holds. Then the probability that the sampling procedure reaches \( \Upsilon_\sigma \) is at most

\[
Pr_{\rho \sim \pi_\sigma} (\exists T \in W \text{ such that } T \subseteq F(\rho)).
\]

By a union bound we have

\[
\sum_{\rho \in \Upsilon_\sigma} P_{1,\rho} \leq \sum_{T \in W} Pr_{\rho \sim \pi_\sigma} (T \subseteq F(\rho)),
\]

so to finish we will show

\[
\sum_{T \in W} Pr_{\rho \sim \pi_\sigma} (T \subseteq F(\rho)) \leq (k \Delta)^{-8\ell}.
\]

Consider a set \( T \in W \) and any \( \rho = (A_1,A_2,V_1,V_{\text{set}},C_{\text{rem}},F,R) \) such that \( \rho \) is in the support of \( \pi_\sigma \) and \( T \subseteq F(\rho) \). By the definition of \( \pi_\sigma \), \( \sigma \in \Omega^{A_1(\rho)} \cup \Lambda \). By (P8), every \( c \in T \) has \( R(c) \) non-empty.

Let \( T_{\text{good}} = T \cap \Lambda_{\text{good}} \) and note that \( |T_{\text{good}}| \geq 2\ell/3 \). By (P9.1), every \( c \in T_{\text{good}} \) has \( R(c) \subseteq \{\text{disagree},1,2\} \).

Let \( S = \{c \in T_{\text{good}} \mid 1 \in R(c)\} \). By (P9.3), every \( c \in S \) has marked(\( c \)) \( \subseteq V_{\text{set}} \) and \( A_1 \) does not satisfy \( c \). Since \( \sigma \) is \( \ell \)-correct, \( |S| < \lceil \ell/2 \rceil \) so (by integrality) \( |S| < \ell/2 \). Thus, \( |T_{\text{good}} \setminus S| \geq (2/3 - 1/2)\ell = \ell/6 \).

We claim that for any \( c \in T_{\text{good}} \setminus S \), disagree \( \in R(c) \). We have already seen that \( R(c) \) is non-empty and that it is contained in \( \{\text{disagree},2\} \). If \( 2 \in R(c) \), then by (P9.3), \( c \) is satisfied by \( \Lambda \cap A_1 \) but not \( \Lambda \cap A_2 \), which implies that disagree \( \in R(c) \) by (P9.2).
Since \((T_{\text{good}} \setminus S) \subseteq T \in D(G_\Phi)\), Definition 27 ensures that the clauses in \(T_{\text{good}} \setminus S\) do not share good variables. Thus by (P9.2), there is a set \(V \subseteq \nu^\Lambda\) with \(|V| \geq \ell/6\) such that for every \(u \in V\), \(A_1(u) \neq A_2(u)\).

By the definition of the sampling procedure (Definition 32), the probability that such a disagreement occurs when the first variable \(u \) of a node \(\rho'\) is set is at most \(P_{1,\rho'_{u \rightarrow X,u \rightarrow X}}/P_{1,\rho'}\) for some \(X \in \{T,F\}\). By Constraint Set 3, this ratio is at most \(1/s\). Also, these events are independent for the variables \(u \in V\). Thus,

\[
\Pr_{\rho \sim \pi}(T \subseteq \mathcal{F}(\rho)) \leq s^{-\ell/6}.
\]

By the choice of \(C_0\), \(\ell\) is at least \(\log n\). Lemma 28 implies that w.h.p., over the choice of \(\Phi\), \(|W| \leq (18k^2\alpha)^{4\ell}\). By a union bound (using the fact that \(k\) is sufficiently large and \(\alpha < 2k/300\)),

\[
\sum_{T \in W} \Pr_{\rho \sim \pi}(T \subseteq \mathcal{F}(\rho)) \leq (18k^2\alpha)^{4\ell}s^{-\ell/6} = \left(18^4k^8\alpha^4s^{-1/6}\right)^\ell \leq (k\Delta)^{-8\ell}. \quad \Box
\]

The following lemma is the same as Lemma 33, except that we take \(i = 2\) rather than \(i = 1\). The proof is exactly the same as that of Lemma 33 except that the sampling procedure (analogous to the one from Definition 32) is conditioned on \(\sigma \in \Omega^2_1\) and the transition from \(\rho\) to \(\rho_{u \rightarrow X,u \rightarrow \sigma(u)}\) is with probability \(P_{2,\rho_{u \rightarrow X,u \rightarrow \sigma(u)}}/P_{2,\rho}\).

**Lemma 34.** Fix \(r_{\text{lower}} \leq r_{\text{upper}}\). Let \(\ell = L/(3k^2\Delta)\). W.h.p. over the choice of \(\Phi\), the following holds. Let \(\sigma\) be any \(\ell\)-correct assignment in \(\Omega^2_1\). If the LP has a solution using \(r_{\text{lower}}\) and \(r_{\text{upper}}\) then \(\sum_{\rho \in T, \sigma \in \Omega^w_{\Lambda_2}(\rho) \cup A} P_{1,\rho} \leq (k\Delta)^{-8\ell}\).

### 6.4 Soundness

In this section we show the “soundness” of the LP, namely, that whenever a solution to the LP exists, it yields a bound on \(|\Omega^1_1|/|\Omega^2_1|\).

**Lemma 35.** Fix \(r_{\text{lower}} \leq r_{\text{upper}}\). W.h.p. over the choice of \(\Phi\), the following holds. If the LP has a solution \(P\) using \(r_{\text{lower}}\) and \(r_{\text{upper}}\), then \(e^{-3\ell/(3k\Delta)}r_{\text{lower}} \leq |\Omega^1_1|/|\Omega^2_1| \leq e^{3\ell/(3k\Delta)}r_{\text{upper}}\).

**Proof.** By Lemma 19, the constraints in Constraint Set 2 guarantee that, for any \(i \in \{1,2\}\) and \(\sigma \in \Omega^i_1\),

\[
\sum_{\rho \in \mathcal{L}^*: \sigma \in \Omega^i_1(\rho) \cup A} P_{i,\rho} = 1.
\]

Thus,

\[
|\Omega^1_1| = \sum_{\sigma \in \Omega^1_1} 1 = \sum_{\sigma \in \Omega^1_1} \sum_{\rho \in \mathcal{L}^*: \sigma \in \Omega^1_1(\rho) \cup A} P_{1,\rho}.
\]

Let \(\ell = L/(3k^2\Delta)\). We start by defining \(Z_i\), \(Z_i'\) and \(Z_i''\) as follows for \(i \in \{1,2\}\).

\[
Z_i = \sum_{\sigma \in \Omega^1_1} \sum_{\rho \in \mathcal{L}^*: \sigma \in \Omega^1_1(\rho) \cup A} P_{1,\rho},
\]

\[
Z_i' = \sum_{\sigma \in \Omega^1_1} \sum_{\rho \in \mathcal{L}^*: \sigma \in \Omega^1_1(\rho) \cup A} P_{1,\rho},
\]

\[
Z_i'' = \sum_{\sigma \in \Omega^1_1} \sum_{\rho \in \mathcal{T}: \sigma \in \Omega^1_1(\rho) \cup A} P_{1,\rho}.
\]

Thus \(Z_i \leq |\Omega^1_1| \leq Z_i + Z_i' + Z_i''\). The proof consists of three parts — as we will see soon, the statement of the lemma follows directly from Equations (11), (12), and (13).

**Part 1:** Showing

\[
(11) \quad r_{\text{lower}} \leq \frac{Z_1}{Z_2} \leq r_{\text{upper}}.
\]
Part 2: Showing, for \( i \in \{1, 2\} \),

\[
\frac{Z_i'}{\Omega_i^A} \leq \frac{1 - e^{-\varepsilon/(3n)}}{2}.
\]

Part 3: Showing, for \( i \in \{1, 2\} \),

\[
\frac{Z''_i}{\Omega_i^A} \leq \frac{1 - e^{-\varepsilon/(3n)}}{2}.
\]

We now present the three parts of the proof.

**Part 1.** This part is straightforward. Exchanging the order of summation in the definition of \( Z_i \), we have

\[
Z_i = \sum_{\rho \in \mathcal{L}} \sum_{\sigma \in \Omega_i^A : \sigma \in \Omega_A^i(\rho) \cup \Lambda} P_{i,\rho} = \sum_{\rho \in \mathcal{L}} P_{i,\rho} \cdot |\Omega_A^i(\rho) \cup \Lambda|.
\]

Since \( \rho \in \mathcal{L} \), Lemma 18 guarantees that \( r(\rho) = |\Omega_A^i(\rho) \cup \Lambda|/|\Omega_A^i(\rho) \cup \Lambda| \) and Constraint Set 1 guarantees that

\[
r_{lower} \leq P_{i,\rho} \cdot r(\rho) \leq r_{upper}.
\]

Plugging in (14), we get (11), as required.

**Part 2.** Note that for any \( \sigma \),

\[
\sum_{\rho \in \mathcal{T}: \sigma \in \Omega_A^i(\rho) \cup \Lambda} P_{i,\rho} \leq 1.
\]

By Lemma 31, we have that

\[
\frac{Z''_i}{\Omega_i^A} \leq \frac{|\{\sigma \in \Omega_i^A : \sigma \text{ is } \ell\text{-wrong}\}|}{|\Omega_i^A|} \leq (k \Delta)^{-9\ell}.
\]

Since \( \ell = L/(3k^2 \Delta) = C_0 \lceil \log(n/\varepsilon) \rceil \), we verify that

\[
(k \Delta)^{-9\ell} \leq (\varepsilon/n)^{-9C_0 \log(k \Delta)} \leq (1 - e^{-\varepsilon/(3n)}/2),
\]

which implies (12). This finishes Part 2.

**Part 3.** By Lemmas 33 and 34, W.h.p. over the choice of \( \Phi \), for every \( \ell\)-correct \( \sigma \in \Omega_i^A \),

\[
\sum_{\rho \in \mathcal{T}: \sigma \in \Omega_A^i(\rho) \cup \Lambda} P_{i,\rho} \leq (k \Delta)^{-8\ell}.
\]

Hence we have

\[
\frac{Z''_i}{\Omega_i^A} \leq \frac{Z''_i}{|\{\sigma \in \Omega_i^A : \sigma \text{ is } \ell\text{-correct}\}|} \leq (k \Delta)^{-8\ell}.
\]

Again, \( \ell = L/(3k^2 \Delta) = C_0 \lceil \log(n/\varepsilon) \rceil \) implies (13). This finishes Part 3.

Having finished the three parts, we now complete the proof. Combining (12) and (13) with the fact that \( Z_i \leq |\Omega_i^A| \leq Z_i + Z'_i + Z''_i \), we get

\[
e^{-\varepsilon/(3n)} \leq \frac{Z_i}{|\Omega_i^A|} \leq 1.
\]

Plugging in (11) we obtain

\[
e^{-\varepsilon/(3n)} r_{lower} \leq \frac{|\Omega_i^A|}{|\Omega_i^A|} \leq e^{\varepsilon/(3n)} r_{upper}.
\]

7. Properties of the random formula

We still need to prove Lemmas 14, 15, 28, and 29. All of these lemmas depend on properties of the random formula. We use \( \Pr_{\Phi}(\cdot) \) to denote the distribution for choosing \( \Phi \).
7.1. Bounding the number of high-degree variables. Recall from Section 2 that $\mathcal{V}_0$ is the set of high-degree variables.

Lemma 36. W.h.p. over the choice of $\Phi$, the size of $\mathcal{V}_0$ is at most $n/2^{k^{10}}$.

Proof. The degrees of the variables in $\Phi$ have the same distribution as a balls-and-bins experiment with $km$ balls and $n$ bins. Let $D_1, \ldots, D_n$ be a set of independent Poisson variables with parameter $k\alpha$, denoted $\text{Poi}(k\alpha)$. It follows by well-known facts (see, e.g., [28, Chapter 5.4]) that the degrees of the variables in $\Phi$ have the same distribution as $\{D_1, \ldots, D_n\}$ conditioned on the event $\mathcal{E}$ that $D_1 + \ldots + D_n = km$, and that $\Pr(\mathcal{E}) = O(1/\sqrt{n})$. Let $U = \{i \in [n] : D_i \geq \Delta\}$, then

$$
\mathbb{E}[|U|] = n \Pr(\text{Poi}(k\alpha) \geq \Delta) \leq ne^{-\Delta} \leq n/2^{k^{10}+1},
$$

where the first inequality follows from $k\alpha \leq \Delta/k^2$ and using standard bounds for the tails of the Poisson distribution (see, e.g., [28, Theorem 5.4]). A Chernoff bound therefore yields that $\Pr(|U| \geq n/2^{k^{10}}) = \exp(-\Omega(n))$. It follows that

$$
\Pr_\Phi(|\mathcal{V}_0| \geq n/2^{k^{10}}) = \Pr(|U| \geq n/2^{k^{10}} | \mathcal{E}) \leq \exp(-\Omega(n)). \quad \square
$$

Lemma 37. Let $2 \leq b \leq k$ be an integer and $t = \frac{2}{1 + k}$. W.h.p. over the choice of $\Phi$, for every set of variables $Y$ such that $2 \leq |Y| \leq n/2^k$, the number of clauses that contain at least $b$ variables from $Y$ is at most $ct|Y|$.

Proof. Let $y$ be an integer between $b$ and $n/2^k$. There are $\binom{n}{y}$ ways to choose a set $Y$ of $y$ variables and $\binom{m}{\lceil ty \rceil}$ ways to choose a set $Z$ of $\lceil ty \rceil$ clauses. The probability that a clause contains at least $b$ variables from $Y$ and $Z$, we therefore obtain that

$$
\Pr_\Phi(\exists Y, Z) \leq \sum_{b \leq y \leq n/2^k} \binom{n}{y} \binom{m}{\lceil ty \rceil} \binom{ky}{b} \leq \sum_{b \leq y \leq n/2^k} \left( \frac{en}{y} \right)^y \left( \frac{e\alpha n}{t^y} \frac{ky}{n} \right)^b \binom{k}{\lceil ty \rceil}.
$$

Note (by taking $k$ to be its upper bound) that the quantity taken to the $\lceil ty \rceil$ power is at most

$$
\frac{2e\alpha k^2}{(b-1)\left(\frac{k}{2^k}\right)^b}.
$$

This is maximised at $b = 2$, so it is always at most $1$ (given that $\alpha < 2^k$ for some $r < 1$ and that $k$ is sufficiently large). Thus, the quantity is maximised by removing the ceiling, so

$$
\Pr_\Phi(\exists Y, Z) \leq \sum_{b \leq y \leq n/2^k} \left( \frac{en}{y} \frac{e\alpha n}{t^y} \right)^t \left( \frac{ky}{n} \right)^b \left( \frac{e^{1+k^b\alpha^t}n}{t^yn} \right)^y = \sum_{b \leq y \leq n/2^k} \left( \frac{e^{1+k^b\alpha^t}n}{t^yn} \right)^y = o(1).
$$

The last estimate follows from observing the inequalities $\left( \frac{e^{1+k^b\alpha^t}n}{t^yn} \right)^y \leq 1/n$ for $2 \leq y \leq \log n$, and $\left( \frac{e^{1+k^b\alpha^t}n}{t^yn} \right)^y \leq 1/10$ for $\log n < y \leq n/2^k$, which hold for all sufficiently large $n$. \quad \square

Applying Lemma 37 with $b = t = 2$ and with $b = \lceil k/10 \rceil$, $t = 2/(b-1) < 30/k$ gives the following two corollaries, respectively.

Corollary 38. W.h.p. over the choice of $\Phi$, for every set of variables $Y$ such that $2 \leq |Y| \leq n/2^k$, the number of clauses that contain at least $2$ variables from $Y$ is at most $2|Y|$.

Corollary 39. W.h.p. over the choice of $\Phi$, for every set of variables $Y$ such that $2 \leq |Y| \leq n/2^k$, the number of clauses that contain at least $k/10$ variables from $Y$ is at most $\frac{2|Y|}{k}$. 

19
7.2. Bounding the number of connected sets of clauses. In this section, we bound the number of connected sets of clauses which we will use in the upcoming sections. Recall that a set \( Y \) of clauses is connected if \( G_\Phi[Y] \) is connected.

**Lemma 40.** For any labelled tree \( T \) on a subset of clauses in \( \Phi \), the probability that \( T \) is a subgraph of \( G_\Phi \) is at most \((k^2/n)^{|V(T)|-1}\).

**Proof.** For a tree \( T \) on a subset of clauses in \( \Phi \), we use \( T \subseteq G_\Phi \) to denote that \( T \) is a subgraph of \( G_\Phi \). We prove our claim by induction on the size of \( T \). If \(|V(T)| = 1\), then \( T \) contains only an isolated clause, so \( \Pr(T \subseteq G_\Phi) = 1 \). Now suppose that the claim holds for all trees with size \(|V(T)| - 1\). Let \( c \) be a leaf in \( T \) and \( c' \) be its neighbour in \( T \). Then we have
\[
\Pr_\Phi(T \subseteq G_\Phi) = \Pr_\Phi((T \setminus c) \subseteq G_\Phi) \Pr_\Phi(\text{var}(c) \cap \text{var}(c') \neq \emptyset \mid (T \setminus c) \subseteq G_\Phi).
\]
For any fixed \( c' \), we have that
\[
\Pr(\text{var}(c) \cap \text{var}(c') \neq \emptyset) \leq \sum_{v \in \text{var}(c')} \Pr(v \in \text{var}(c)) \leq \frac{k^2}{n}.
\]
Note that the events \((T \setminus c) \subseteq G_\Phi \) and \( \text{var}(c) \cap \text{var}(c') \neq \emptyset \) are independent, so
\[
\Pr(\text{var}(c) \cap \text{var}(c') \neq \emptyset \mid (T \setminus c) \subseteq G_\Phi) \leq \frac{k^2}{n}.
\]
Since \( T \setminus c \) is a tree of size \(|V(T)| - 1\), by the induction hypothesis we have that \( \Pr((T \setminus c) \subseteq G_\Phi) \leq (k^2/n)^{|V(T)|-2} \). We conclude that \( \Pr(T \subseteq G_\Phi) \leq (k^2/n)^{|V(T)|-1} \).

**Lemma 41.** W.h.p. over the choice of \( \Phi \), for any clause \( c \), the number of connected sets of clauses in \( G_\Phi \) with size \( \ell \geq \log n \) containing \( c \) is at most \((9k^2\alpha)^\ell\).

**Proof.** Let \( c \) be an arbitrary clause. Let \( U \) be a size-\( \ell \) set of clauses containing \( c \) and let \( T_U \) be the set of all labelled trees on the set \( U \); note that \(|T_U| = \ell^{\ell-2}\). For any tree \( T \in T_U \), by Lemma 40, the probability that \( T \subseteq G_\Phi \) is at most \((k^2/n)^{\ell-1}\). Thus
\[
\Pr_\Phi(G_\Phi[U] \text{ is connected}) \leq \sum_{T \in T_U} \Pr_\Phi(T \subseteq G_\Phi) = \ell^{\ell-2}(k^2/n)^{\ell-1}.
\]
Let \( Z_c \) be the number of connected sets of clauses with size \( \ell \) containing \( c \). Then,
\[
E_\Phi[Z_c] = \sum_{U \subseteq c; c \in U; |U| = \ell} \Pr_\Phi(G_\Phi[U] \text{ is connected})
\]
\[
\leq \left( \begin{array}{c} m - 1 \end{array} \right)^{\ell-2} \left( \frac{k^2}{n} \right)^{\ell-1} \leq \left( \frac{e(m - 1)}{\ell - 1} \right)^{\ell-2} \left( \frac{k^2}{n} \right)^{\ell-1}
\]
\[
\leq \left( \frac{emk^2}{n} \frac{\ell-2}{\ell - 1} \right)^{\ell-1} \leq (ek^2\alpha)^{\ell-1}.
\]
As \( \ell \geq \log n \), by Markov’s inequality, we obtain that \( \Pr_\Phi(Z_c \geq (9k^2\alpha)^\ell) \leq \left( \frac{e}{\ell} \right)^{\ell-1} = o(1/n) \).

By a union bound over all clauses \( c \) (note that there are \( O(n) \) for them), we obtain the conclusion of the lemma.

7.3. Bounding the number of high-degree variables in connected sets. In this section, we bound the number of high-degree variables in connected sets of variables.

Recall that \( \mathcal{V}_0 \) is the set of high-degree variables. For every set \( S \) of variables, let \( \text{HD}(S) = \mathcal{V}_0 \cap S \) be the set of high-degree variables in \( S \). With this notation, \( \mathcal{V}_0 = \text{HD}(\mathcal{V}) \). For a set \( Y \subseteq \mathcal{C} \) of clauses, let \( \text{var}(Y) := \bigcup_{c \in Y} \text{var}(c) \).

**Lemma 42.** Let \( \delta_0 > 0 \) and \( \theta_0 \geq \min(k^2\alpha, 2) \) be constants such that \( \delta_0 \theta_0 \log(\theta_0/k^2\alpha) > \log \alpha + 3 \log k \). Then, w.h.p. over the choice of \( \Phi \), there do not exist sets \( Y, Z \) of clauses and a set \( U \) of variables such that:

1. \( |Y| \geq \log n, |U| \geq \delta_0 |Y|, |Z| \geq \theta_0 |U|, \) and \( Y \cap Z = \emptyset; \)
(2) $G_{\Phi}[Y]$ is connected, $U \subseteq \text{var}(Y)$, and every clause in $Z$ contains at least one variable from $U$.

Proof. Let $E$ the event that there exist sets $Y, Z, U$ satisfying conditions (1) and (2).

Call a tuple $(y, \delta, \theta)$ feasible if $y, \delta y, \theta y$ are all integers, where $y \geq \log n$, $\delta \geq \delta_0$ and $\theta \geq \theta_0$.
Fix a feasible tuple $(y, \delta, \theta)$ and three sets of indices $I_Y \in \binom{|m|}{y}$, $I_U \in \binom{|Y| \times |Z|}{\delta y}$, $I_Z \in \binom{|m| \setminus I_Y}{\delta \theta y}$.
Define

\[ S_Y = \{c_i \mid i \in I_Y\} \]
\[ S_U = \{\text{var}(\ell_{i,j}) \mid (i, j) \in I_U\} \]
\[ S_Z = \{c_i \mid i \in I_Z\}. \]

Denote by $E$ the event that $|S_Y| = \delta y$, by $E_Y$ the event that $G_{\Phi}[S_Y]$ is connected and by $E_Z$ the event that every clause in $S_Z$ contains at least one variable from $S_U$. For any labelled tree $T$ on vertex set $S_Y$, by Lemma 40, the probability that $T$ is a subgraph of $G_{\Phi}$ is at most $(k^2/n)^{y-1}$.
We have $y^{y-2}$ such trees, so by a union bound

\[ \Pr_{\Phi}(E_Y) \leq y^{y-2}\left(\frac{k^2}{n}\right)^{y-1}. \]

Moreover, by the independence of clauses, we have that

\[ \Pr_{\Phi}(E_Z \mid E_Y) = \Pr_{\Phi}(E_Z \mid E_U) \leq \left(\frac{k \delta y}{n}\right)^{\theta y}. \]

Note that

\[ \Pr_{\Phi}(E_U \cap E_Y \cap E_Z) \leq \Pr_{\Phi}(E_Y) \Pr_{\Phi}(E_Z \mid E_Y), \]

so by a union bound over the choice of the tuple $(y, \delta, \theta)$ and the sets $I_Y, I_U, I_Z$ we have that

\[ \Pr_{\Phi}(E) \leq \sum_{\text{feasible } (y, \delta, \theta)} \sum_{I_Y \in \binom{|m|}{y}, I_U \in \binom{|Y| \times |Z|}{\delta y}, I_Z \in \binom{|m| \setminus I_Y}{\delta \theta y}} \Pr_{\Phi}(E_Y) \Pr_{\Phi}(E_Z \mid E_Y). \]

It follows that

\[ \Pr_{\Phi}(E) \leq \sum_{\text{feasible } (y, \delta, \theta)} \left(\frac{m}{y}\right) \left(\frac{k y}{\delta y}\right) \left(\frac{m}{\theta y}\right) y^{y-2}\left(\frac{k^2}{n}\right)^{y-1}\left(\frac{k \delta y}{n}\right)^{\theta y} \]

\[ \leq \sum_{\text{feasible } (y, \delta, \theta)} \left(\frac{(e k)^{2+\delta+\theta \delta y \alpha^{\delta+1}}}{\delta^\delta \theta^\theta \alpha^\delta} \right)^y. \]

Note that $\delta \log(\theta/k^2 \alpha) \geq \delta_0 \log(\theta_0/k^2 \alpha) \geq \log \alpha + 3 \log k$ and hence $\theta^\delta \geq (k^2 \alpha)^{\delta \theta} k^3 \alpha$. It follows that

\[ \frac{(e k)^{2+\delta+\theta \delta y \alpha^{\delta+1}}}{\delta^\delta \theta^\theta \alpha^\delta} \leq \frac{e^{2+\delta+\theta \delta y} k^\delta}{\delta^\delta \theta^\theta \alpha^\delta} \leq \frac{e^{2+\delta+\theta \delta y} k^\delta}{\delta^\delta (k/e)^{\theta \delta} k^\delta} \leq \frac{e^{2+\delta+\theta \delta y} k^\delta}{k (k/e)^{\theta \delta} k^\delta} \leq \frac{e^{2+\delta+\theta \delta y} k^\delta}{k^2} < \frac{1}{e^\delta}, \]

where the last few inequalities hold for sufficiently large $k$ combined with the fact that $\delta \geq 1/2$ for all $\delta > 0$ and our assumption $\theta \geq \theta_0 \geq 2$. Plugging this estimate into (15) and noting that there are $O(n^3)$ feasible tuples $(y, \delta, \theta)$ and $y \geq \log n$, we obtain that $\Pr_{\Phi}(E) = o(1)$, as needed.

Lemma 43. W.h.p. over the choice of $\Phi$, every connected set $U$ of variables with size at least $21600 k \log n$ satisfies that $|\text{HD}(U)| \leq \frac{U}{21600}$. 

Proof. Let $\delta_0 = \frac{1}{21600}$ and $\theta_0 = \Delta - 2(k + 1)$. Note that $\delta_0 \theta_0 \log \frac{\theta_0}{k^2 \alpha} \geq 3 \log k + \log \alpha$ for all sufficiently large $k$, so w.h.p. we have that $\Phi$ satisfies Lemma 42. Moreover, w.h.p. we have that $\Phi$ satisfies Corollary 38 and Lemma 36. We will show the conclusion of the lemma whenever $\Phi$ satisfies these properties.

For the sake of contradiction, suppose that $U$ is a connected set of variables with $|U| \geq (k/\delta_0) \log n$ such that $h > \delta_0 |U|$ where $h = |\text{HD}(U)|$ is the number of high-degree variables in $U$. Recall that the factor graph of $\Phi$ is a bipartite graph where one side corresponds to variables
and the other to clauses (whose edges join variables to clauses in the natural way). We next show that there is a tree $T$ in the factor graph of $\Phi$ of size at most $2|U|$ such that

1. every vertex in $T$ is either a variable in $U$ or a clause in $\Phi$, all variables in $\text{HD}(U)$ are vertices in $T$, and $T$ contains at most $|U|$ clauses;
2. every edge in $T$ joins a variable and a clause, and for any variable $v$ and clause $c$, $(v, c)$ is an edge in $T$ only if $c$ contains $v$;
3. $T_L \subseteq \text{HD}(U)$, where $T_L$ is the set of leaves of $T$;
4. if a clause $c \in T$ contains any variable from $\text{HD}(U)$, then at least one of its neighbours in $T$ is a variable from $\text{HD}(U)$.

Since $H_\Phi[U]$ is connected, there is a tree $T'$ of size at most $2|U|$ that satisfies Items (1) and (2) (for example, we may take the Steiner tree with terminals $\text{HD}(U)$ in the subgraph of the factor graph induced by $U$ and its adjacent clauses). We now prune $T'$ so that it satisfies Items (4) and (3) as well. For any clause $c$ in $T'$ such that $c$ contains at least one variable from $\text{HD}(U)$ but none of its neighbours in $T'$ is from $\text{HD}(U)$, let $v$ be a variable from $\text{var}(c) \cap \text{HD}(U)$, and $u$ be the neighbour of $c$ on the path from $c$ to $v$. Then we remove the edge $(c, u)$ from $T'$ and add the edge $(c, v)$. Run this process until there is no such clause $c$. Now $T'$ is a tree that satisfies Items (1), (2), (4). If $T'$ has a leaf node which is not in $\text{HD}(U)$, remove it from $T'$. Run this process until all leaf nodes are in $\text{HD}(U)$ and let $T$ be the remaining tree. Note that removing leaf nodes that are not in $\text{HD}(U)$ does not affect Items (1), (2) and (4). We thus obtain a tree $T$ satisfying all of these four items.

Let $C_T$ be the set of clauses in $T$. From Item (1), we have $h/k \leq |C_T| \leq |U|$. Let $t$ be the number of clauses in $C_T$ that contain at least one variable from $\text{HD}(U)$. By Item (4), we have

$$t \leq \sum_{v \in \text{HD}(U) \setminus T_L} \deg_T(v).$$

Let $D = \sum_{v \in \text{HD}(U) \setminus T_L} \deg_T(v)$. Because $T$ is a tree and $\text{HD}(U) \subseteq T$, we obtain that

$$|T_L| = 2 + \sum_{v \in T \setminus T_L} (\deg_T(v) - 2) \geq 2 + \sum_{v \in \text{HD}(U) \setminus T_L} (\deg_T(v) - 2) = 2 + D - 2(h - |T_L|),$$

which yields that $D + |T_L| \leq 2h$. Thus we have $t \leq D + |T_L| \leq 2h$.

By our assumption on $U$, we have that the number of high-degree variables in $U$ satisfies $h > \delta_0 |U| \geq k \log n$ and therefore $|C_T| \geq \log n$. Each variable in $\text{HD}(U)$ is contained in at least $\Delta$ clauses. Moreover, by Lemma 36 we have $|\text{HD}(U)| \leq |\mathcal{V}_0| \leq n/2^k$, so by Corollary 38 the number of clauses that contain at least 2 variables from $\text{HD}(U)$ is at most $2h$. It follows that the number of clauses that contain at least one variable from $\text{HD}(U)$ is at least $\Delta h - 2hk$. Since $t \leq 2h$, at most $2h$ of these clauses appear in $T$. Hence, there must exist a set $Z$ of clauses of size at least $(\Delta - 2(k + 1))h = \theta_0 h$ such that $Z \cap C_T = \emptyset$ and each clause in $Z$ contains at least one variable from $\text{HD}(U)$.

Note that $|C_T| \geq \log n$, $|\text{HD}(U)| \geq \delta_0 |C_T|$ and $|Z| \geq \theta_0 |\text{HD}(U)|$. Moreover, $C_T$ is a connected set of clauses, $\text{HD}(U) \subseteq \text{var}(C_T)$ and every clause $Z$ contains at least one variable from $\text{HD}(U)$. This contradicts that $\Phi$ satisfies Lemma 42. Therefore, no such set $U$ can exist, proving the lemma.

### 7.4. Bounding the number of bad variables in connected sets

In this section, we bound the number of bad variables in connected sets. Consider the following process $P$ which we will use to work with bad components. The process, for every set $S$ of variables, defines a set of variables $\text{BC}(S)$.

1. Let $\text{BC}(S) = S$.
2. While there is a clause $c$ such that $|\text{var}(c) \cap \text{BC}(S)| \geq k/10$ and $\text{BC}(S) \setminus \text{var}(c) \neq \emptyset$

   $\text{BC}(S) := \text{BC}(S) \cup \text{var}(c)$

Note that $\mathcal{V}_\text{bad} = \text{BC}(\mathcal{V}_0)$, where $\mathcal{V}_0$ is the set of high-degree variables. Recall from Section 2 that a bad component is a connected component of variables in $H_{\Phi, \text{bad}}$. The following lemma shows that the process $P$ can be viewed as a “local” process for identifying bad components.
Lemma 47. W.h.p. over the choice of the corollary. \( \lambda \) and \( \ell \) with

For an integer \( s \) satisfying \( 1 \leq s \leq 2n/2^{k/10} \), let \( \mathcal{E}_z \) be the event that there exists a set \( Z \) with \( |Z| = z \) that does not satisfy the desired property. By Lemma 45 (applied with \( \varepsilon = z/n \) and \( \lambda = k/10 \)), we have \( \Pr_\Phi(\mathcal{E}_z) = o(1/n) \). Taking a union bound over the values of \( z \) yields the corollary.

Corollary 46 allows us to control the number of bad variables.

Lemma 47. W.h.p. over the choice of \( \Phi \), \( |V_{bad}| \leq 4kn/2^{k/10} \).
Thus, by Lemma 48, we have $|\mathcal{V}_0| \leq n/2^{k^{10}}$ and hence by Corollary 38 (applied to $Y = \mathcal{V}_0$), we obtain that $|\mathcal{C}_0| \leq 2n/2^{k^{10}}$. By Lemma 45 (applied to $Z = \mathcal{C}_0$), we conclude that $|\mathcal{C}_{bad}| \leq 4n/2^{k^{10}}$ and hence $|\mathcal{V}_{bad}| \leq 4kn/2^{k^{10}}$. □

We can in fact use Corollary 46 to prove the following lemma.

**Lemma 48.** *W.h.p. over the choice of $\Phi$, for any bad component $S$, $|S| \leq 60|\text{HD}(S)|$.*

**Proof.** W.h.p. we have that $\Phi$ satisfies the properties in Lemma 36 and Corollaries 39 and 46. We will show the conclusion of the lemma whenever $\Phi$ satisfies these three properties.

Let $S$ be a bad component. If $S$ contains only an isolated variable, it must be high-degree and hence $\text{HD}(S) = S$ (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 $|\text{HD}(S)| \leq |\mathcal{V}_0| \leq n/2^{k^{10}}$ by Lemma 36. Applying Corollary 39 with $Y = \text{HD}(S)$, we find that there are at most $\frac{30}{k}|\text{HD}(S)|$ clauses that contain at least $k/10$ variables from $\text{HD}(S)$.

Now, we run the process $P$ starting with $\text{HD}(S)$. Take $Z$ to be the set of clauses that contain at least $k/10$ variables from $\text{HD}(S)$ (so, from above, we have $|Z| \leq \frac{30}{k}|\text{HD}(S)| \leq \frac{30}{2^{k^{10}}}$). Applying Corollary 46, we find that the number of clauses $c$ such that $\text{var}(c) \subseteq \text{BC}(\text{HD}(S))$ is at most $2|Z| \leq 60|\text{HD}(S)|/k$. Since $S = \text{BC}(\text{HD}(S))$ by Lemma 44 and each variable in $S$ is contained in some bad clause, we have

$$|S| \leq \left| \bigcup_{c \in \mathcal{C}_{bad} : \text{var}(c) \cap S \neq \emptyset} \text{var}(c) \right| \leq 60|\text{HD}(S)| \text{.} \quad \Box$$

Next, we show that there is no large bad component.

**Lemma 49.** *W.h.p. over the choice of $\Phi$, every bad component $S$ has size at most $21600k \log n$.*

**Proof.** Suppose there is a bad component $S$ with size $|S| > 21600k \log n$. Since $S$ is a connected component in $H_{\Phi, \text{bad}}$, $S$ is also a connected set in $H_\Phi$. By Lemma 43, $|\text{HD}(S)| \leq \frac{|S|}{21600}$. However by Lemma 48, we have $|S| \leq 60|\text{HD}(S)|$, which gives a contradiction. □

The following lemma shows that every “large” connected set contains few bad variables.

**Lemma 50.** *W.h.p. over the choice of $\Phi$, for every connected set $S$ of variables with size at least $21600k \log n$, $|S \cap \mathcal{V}_{\text{bad}}| \leq |S|/360$.*

**Proof.** W.h.p. we have that $\Phi$ satisfies the properties in Lemmas 43 and 48. We will show the conclusion of the lemma for all such $\Phi$.

For the sake of contradiction, let $S$ be a connected set of variables with size at least $21600k \log n$ and $|S \cap \mathcal{V}_{\text{bad}}| > |S|/360 = 60\delta_0|S|$, where $\delta_0 = 1/21600$. Suppose that there are $t$ bad components $S_1, S_2, \ldots, S_t$ intersecting $S$. Let $S' = S \cup S_1 \cup \cdots \cup S_t$ and let $b = |S' \setminus S|/|S|$. Note that $S'$ is a connected set of variables. Also, all variables in $S' \setminus S$ are bad, so $|S' \cap \mathcal{V}_{\text{bad}}| > (60\delta_0 + b)|S|$. Thus, by Lemma 48, we have

$$|\text{HD}(S')| = \sum_{i=1}^{t} |\text{HD}(S_i)| \geq \sum_{i=1}^{t} \frac{|S_i|}{60} = \frac{|S' \cap \mathcal{V}_{\text{bad}}|}{60} > \left( \frac{\delta_0 + b}{60} \right) |S| > \delta_0(1 + b)|S| = \delta_0|S'| \text{,}$$

which contradicts Lemma 43. □

**Lemma 51.** *W.h.p. over the choice of $\Phi$, for every connected set of clauses $Y$ such that $|\text{var}(Y)| \geq 21600k \log n$, it holds that $|Y \cap \mathcal{C}_{\text{bad}}| \leq |Y|/12$.*

**Proof.** W.h.p. we have that $\Phi$ satisfies the properties in Corollary 39 and Lemmas 47 and 50. We will show the conclusion of the lemma for all such $\Phi$.

Let $Y$ be a connected set of clauses such that $|\text{var}(Y)| \geq 21600k \log n$ and let $S = \text{var}(Y)$. Then $|S| \leq k|Y|$. Since $Y$ is connected, so is $S$. Let $S_{\text{bad}} = S \cap \mathcal{V}_{\text{bad}}$ and note that, by
Lemma 47, $|S_{bad}| \leq |V_{bad}| \leq 4kn/2^{k\alpha}$. By Lemma 50, we also have that $|S_{bad}| \leq |S|/360$. Note that every $c \in Y \cap C_{bad}$ contains at least $k/10$ variables in $S_{bad}$. Applying Corollary 39 (with the “$Y$” in the corollary equal to $S_{bad}$),

$$|Y \cap C_{bad}| \leq \frac{30|S_{bad}|}{k} \leq \frac{|S|}{12k} \leq \frac{|Y|}{12}.$$ 

\[\square\]

7.5. Proofs for the coupling tree. In this section, we prove Lemmas 14 and 15. For $V \subseteq \mathcal{V}$, let $\Gamma_{H_\Phi}(V) = \cup_{v \in V} \Gamma_{H_\Phi}(v)$ be the neighbourhood of $V$ in $H_\Phi$. Let $\Gamma^+_{H_\Phi}(V) = V \cup \Gamma_{H_\Phi}(V)$ be the extended neighbourhood.

**Lemma 52.** W.h.p. over the choice of $\Phi$, every connected set of variables $V \subseteq \mathcal{V}$ satisfies $|\Gamma^+_{H_\Phi}(V)| \leq 3k^2 \alpha \max\{|V|, k \log n\}$.

**Proof.** Let $\delta_0 = 1$, and $\theta_0 = 2k^2 \alpha$. Since $\delta_0 \theta_0 \log(\theta_0/k^2 \alpha) > \log \alpha + 3 \log k$, w.h.p. we have that $\Phi$ satisfies the property in Lemma 42. We will show the conclusion of the lemma for all such $\Phi$.

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

- If $|V| \geq k \log n$, since $V$ is a connected set of variables, there exists a set $Y' \subseteq Y$ such that $|V|/k \leq |Y'| \leq |V|$ and $V \cup Y'$ is connected in the factor graph of $\Phi$. Hence, $Y'$ is a connected set of clauses and $|Y'| \geq \log n$. Let $Z := Y \setminus Y'$. If $|Z| \geq \theta_0 |V|$, then we obtain a contradiction to Lemma 42 (using the sets $U = V, Y', Z$). Thus, $|Z| \leq \theta_0 |V|$ and $|Y| \leq |Y'| + |Z| \leq 3k^2 \alpha |V|$.

- Otherwise $|V| < k \log n$. If $|\Gamma^+_{H_\Phi}(V)| < \lfloor k \log n \rfloor$ then we are finished. Otherwise, consider an arbitrary connected $V' \supset V$ such that $|V'| = \lfloor k \log n \rfloor$. By the argument of the previous case, the set of neighbours of $V'$ in the factor graph, denoted $Y''$, satisfies that $|Y''| \leq 3k^2 \alpha |V'| \leq 3k^3 \alpha \log n$. Thus, $|Y| \leq |Y''| \leq 3k^3 \alpha \log n$.

This completes the proof. \[\square\]

Now we can show Lemma 14, which we restate here for convenience. Recall that $\Lambda^\ast$ is from Lemma 10.

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

**Proof.** W.h.p. we have that $\Phi$ satisfies the property in Lemma 52. We will show the conclusion of the lemma for all such $\Phi$.

Let $\Lambda$ be a prefix of $\Lambda^\ast$ and $\rho$ be a node in $\mathbb{T}^\Lambda$. We first claim that $V_{set}(\rho) \subseteq \Gamma^+_{H_\Phi}(V_\Lambda(\rho))$. To prove the claim, we’ll consider any $u \in V_{set}(\rho) \setminus V_\Lambda(\rho)$ and we will show that there is a clause $c$ containing $u$ and containing a variable in $V_\Lambda(\rho)$.

We first rule out the case that $u = u^\ast$ by noting (via Property (P1)) that $u^\ast \in V_\Lambda(\rho) \cap V_{set}(\rho)$.

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

Let $\rho''$ be the child of $\rho'$ on the path to $\rho$. We will show that there is a clause $c$ containing $u$ and containing a variable in $V_\Lambda(\rho')$. The claim will then follow from the fact that $V_\Lambda(\rho)$ contains $V_\Lambda(\rho')$. The existence of such a clause $c$ is immediate from the definition of “first variable” — indeed $c$ is the “first clause” of $\rho'$. Thus, we have proved the claim.

By (P5), $V_\Lambda(\rho)$ is a connected set of variables. Thus, by Lemma 52 and the claim, $|V_{set}(\rho)| \leq |\Gamma^+_{H_\Phi}(V_\Lambda(\rho))| \leq 3k^3 \alpha \max\{|V_\Lambda(\rho)|, k \log n\}$. If $\rho \notin \mathcal{T}$, then $|V_\Lambda(\rho)| \leq L$ and the lemma holds. Otherwise, apply the above to the parent of $\rho$, which finishes the proof. \[\square\]

Next we show Lemma 15, which we restate here.
Lemma 15. W.h.p. over the choice of $\Phi$, for every truncating node $\rho \in \mathcal{T}$ of $\mathbb{T}^\Lambda$, it holds that $|V_\ell(\rho)| \leq 359L/358$.

Proof. W.h.p. we have that $\Phi$ satisfies the property in Lemma 50. We will show the conclusion of the lemma for all such $\Phi$.

Let $\rho \in \mathcal{T}$ be a truncating node of $\mathbb{T}^\Lambda$ and let $\rho'$ be the parent node of $\rho$. Since $\rho'$ has a child, we have $|V_\ell(\rho')| \leq L$. Let $u$ be the first variable of $\rho'$. Then by the process of creating the child of $\rho'$ (namely Algorithm 1),

$$V_\ell(\rho) \setminus V_\ell(\rho') \subseteq (V_{\text{bad}} \cap V_\ell(\rho)) \cup \bigcup_{c \in u \cap \text{var}(c)} \text{var}(c).$$

By Property (P5) of $\rho$, we have that $V_\ell(\rho)$ is a connected set of variables and hence Lemma 50 gives that

$$|V_{\text{bad}} \cap V_\ell(\rho)| \leq \frac{|V_\ell(\rho)|}{360}.$$

As $u$ is a good variable, we have that $\bigcup_{c \in u \cap \text{var}(c)} \text{var}(c) \leq k\Delta$. It follows that

$$|V_\ell(\rho) \setminus V_\ell(\rho')| \leq k\Delta + \frac{|V_\ell(\rho)|}{360},$$

which gives that

$$\left(1 - \frac{1}{360}\right)|V_\ell(\rho)| - k\Delta \leq |V_\ell(\rho')| \leq L,$$

and hence $|V_\ell(\rho)| \leq (L + k\Delta)360/359$ so $|V_\ell(\rho)| \leq 359L/358$. \hfill $\Box$

7.6. Proofs for $\mathcal{D}(G\Phi)$. In this section we show Lemma 28 and Lemma 29.

Lemma 53. Let $G$ be a connected graph. For any connected induced subgraph $G' = (V', E')$ of $G^{\leq 4}$, there exists a connected induced subgraph of $G$ with size at most $4|V'|$ containing all vertices in $V'$.

Proof. We do an induction on $\ell = |V'|$. If $\ell = 1$ the claim holds since $G'$ is also an induced subgraph of $G$. If $\ell > 1$, assume that the claim holds for all induced subgraphs of $G^{\leq 4}$ with at most $\ell - 1$ vertices. Let $v$ be a vertex of $G'$ such that $G'[V' \setminus \{v\}]$ is connected in $G^{\leq 4}$. Thus, by the induction hypothesis, there exists a connected induced subgraph $G'' = (V'', E'')$ of $G$ such that $(V' \setminus \{v\}) \subseteq V''$ and $|V''| \leq 4(\ell - 1)$. Since $G'$ is connected in $G^{\leq 4}$, there exists a vertex $u \in (V' \setminus \{v\})$ such that $\text{dist}_G(u, v) \leq 4$. Let $U = V'' \cup \{\text{vertices on the path from } u \text{ to } v \text{ in } G\}$. Then the induced subgraph in $G$ whose vertex set is $U$ is connected and $|U| \leq 4\ell$. Thus the claim holds for $G'$. \hfill $\Box$

Corollary 54. Let $G$ be a connected graph and $v \in V(G)$ be a vertex. Let $n_{G, \ell}(v)$ denote the number of connected induced subgraphs of $G$ with size $\ell$ containing $v$. Then

$$n_{G^{\leq 4}, \ell}(v) \leq 2^{\ell'} n_{G, \ell'}(v)$$

where $\ell' := \min\{4\ell, |V(G)|\}$.

Proof. By Lemma 53, for any connected induced subgraph $G'$ of $G^{\leq 4}$ with size $\ell$ containing $v$, there exists a connected induced subgraph $G''$ of $G$ such that $V(G') \subseteq V(G'')$ and $|V(G'')| \leq 4\ell$. In fact we can further assume that $|V(G'')| = \ell'$ since otherwise we can keep adding vertices from neighbours of $G''$ into $G''$ until $|V(G'')| = \ell'$. For any such $G''$ the number of size $\ell$ subsets containing $v$ (corresponding to potential graphs $G'$ which would be mapped to $G''$ by the above construction) is at most $(\ell')^{\ell} \leq 2^{\ell'}$, giving the desired upper bound. \hfill $\Box$

Lemma 28. Let $\ell$ be an integer which is at least $\log n$. W.h.p. over the choice of $\Phi$, every clause $c \in \mathcal{C}_\text{good}^\Lambda$ has the property that the number of size-$\ell$ subsets $T \in \mathcal{D}(G\Phi)$ containing $c$ is at most $(18k^2\alpha)^{4\ell}$.\hfill $\Box$

Proof. Just combine Corollary 54 with Lemma 41.\hfill $\Box$
In the remainder of this section, we will focus on showing Lemma 29. We will need the following ingredients.

**Lemma 55.** For any set $Y \subseteq C_{\text{good}}$ of good clauses, the size of a maximum independent set in $G_{\Phi, \text{good}}[Y]$ is at least $|Y|/(k\Delta)$.

**Proof.** Let $c$ be a clause in $Y$. Note that $c$ contains at most $k$ variables in $V_{\text{good}}$ and each variable in $V_{\text{good}}$ is contained in at most $\Delta$ clauses. So the degree of $c$ in $G_{\Phi, \text{good}}$ is at most $k(\Delta - 1)$. The result follows since every $n$-vertex graph of maximum degree $d$ contains an independent set of size at least $n/(d+1)$. 

We will also use the following properties of $F(\rho)$.

**Lemma 56.** If $\rho$ is a node of the coupling tree, then the following properties hold.

1. $G^\leq_2[F(\rho)]$ is connected.
2. $|F(\rho)| \geq |V_I(\rho)|/k$.

**Proof.** We show Item (1) by induction on the size of $V_{\text{set}}(\rho)$. The base case where $|V_{\text{set}}(\rho)| = 1$ is trivial since, in this case, $\rho = \rho^*$ and $F(\rho^*)$ is the set of clauses containing $v^*$. For the inductive step, we consider a node $\rho' = \rho_{t1, t2}$ being created as a new child of $\rho$ by Algorithm 1 and we consider how clauses are added to $F(\rho')$. We show that each part of the algorithm that adds clauses to $F(\rho')$ maintains the property that $G^\leq_2[F(\rho')]$ is connected. Before line 5, this holds by the inductive hypothesis.

- First, consider the addition of clauses in Line 8. All clauses $c'$ that are added by this line contain the first variable $u$ of $\rho$ which is in the first clause $c$ of $\rho$ so to finish it suffices to show that $F(\rho)$ has a clause which shares a variable with $c$. Since $\text{var}(c) \cap V_I(\rho)$ is non-empty, it suffices to show that every variable in $V_I(\rho)$ is contained in a clause in $F(\rho)$. This is true by Property (P6).
- Next, consider the addition of clauses in Line 18. It is important to note that, after the loop containing Line 8, Property (P6) has been re-established. That is, for any $u'' \in V_I$ there is a clause $c'' \in F$ such that $u'' \in \text{var}(c'')$. All clauses $c'$ added to $F$ in Line 18 have variables in $V_I$ so the introduction of $c'$ leaves $G^\leq_2[F]$ connected. Moreover, the subsequent addition of variables from $\text{var}(c')$ to $V_I$ maintains Property (P6).
- Finally, consider the addition of clauses in Line 24. As in the previous case, Property (P6) guarantees that the introduction of clauses to $F$ leaves $G^\leq_2[F]$ connected. Moreover, the subsequent addition of variables to $V_I$ maintains Property (P6).

Item (2) is a direct consequence of (P6).

We also need the following expansion property (which is a strengthening of Lemma 37 in the case that $b = k$).

**Lemma 57.** ([11, Lemma 2.3]). For all sufficiently large $k$, w.h.p. over the choice of $\Phi$, for any $Y \subseteq C$ such that $|Y| \leq n/k^2$, $|\text{var}(Y)| \geq 0.9k|Y|$.

We are now ready to prove Lemma 29, which we restate here.

**Lemma 29.** W.h.p. over the choice of $\Phi$, every node $\rho$ in $T^\Lambda$ with $|V_I(\rho)| \geq L$ has the property that there is a set $T \subseteq F(\rho)$ containing $c^*$ such that $T \in D(G_\Phi)$, $|T| = C_0\lfloor \log(n/\varepsilon) \rfloor$ and $|T \cap C_{\text{bad}}| \leq |T|/3$.

**Proof.** W.h.p. we have that $\Phi$ satisfies the properties in Lemmas 51 and 57. We will show the conclusion of the lemma for all such $\Phi$. Let $\rho \in T$ be a node of $T^\Lambda$ with $|V_I(\rho)| \geq L$. For a good clause $c$, let $\Gamma^+_\text{good}(c)$ be the set consisting of $c$ and all of its neighbours in $G_{\Phi, \text{good}}$. Recall also from Section 4 that $c^*$ is a good clause (being the first clause of the root node $\rho^*$).

Let $U = F(\rho) \setminus (C_{\text{bad}} \cup \Gamma^+_\text{good}(c^*))$ and $I$ be a maximum independent set of $G_{\Phi, \text{good}}[U]$. We let $T = I \cup \{c^*\} \cup (F(\rho) \cap C_{\text{bad}})$. By Lemma 55 we have $|I| \geq |U|/(k\Delta)$. By construction $T$ contains $c^*$.
Next, we show that $T \in \mathcal{D}(G_\Phi)$. Since $\Gamma^+_{\text{good}}(c^*)$ and $U$ are disjoint, and $I$ is an independent set of $G_{\Phi, \text{good}}[U]$, for any $c_1, c_2 \in T$ we have that $\text{var}(c_1) \cap \text{var}(c_2) \cap \nu_{\text{good}} = \emptyset$ (note that clauses in $C_{\text{bad}}$ only have bad variables). It therefore suffices to show that $T$ is connected in $G_{\Phi}^{\leq 4}$.

Suppose for contradiction that $T$ is not connected in $G_{\Phi}^{\leq 4}$. Then there exists a partition $(S_1, S_2)$ of $T$ such that $S_1 \cup S_2 = T$, $S_1 \cap S_2 = \emptyset$ and $\min_{c_1 \in S_1, c_2 \in S_2} \text{dist}_{G_{\Phi}}(c_1, c_2) \geq 5$. Let $S'_i = (\cup_{c \in S_i} \Gamma^+_{\text{good}}(c)) \cap \mathcal{F}(\rho)$ for $i = 1, 2$. Then we have $\min_{c_1 \in S'_1, c_2 \in S'_2} \text{dist}_{G_{\Phi}}(c_1, c_2) \geq 3$.

Since $T$ is a maximum independent set of $G_{\Phi, \text{good}}[U]$, every clause $c' \in U$ has $\Gamma^+_{\text{good}}(c') \cap I \neq \emptyset$. So $U \subseteq \cup_{c \in I} \Gamma^+_{\text{good}}(c)$. Thus,

$$S'_1 \cup S'_2 \supseteq U \cup (C_{\text{bad}} \cap \mathcal{F}(\rho)) \cup (\Gamma^+_{\text{good}}(c^*) \cap \mathcal{F}(\rho)) = \mathcal{F}(\rho).$$

However, $G_{\Phi}^{\leq 2}[\mathcal{F}(\rho)]$ is connected by Item (1) of Lemma 56, which contradicts

$$\min_{c_1 \in S'_1, c_2 \in S'_2} \text{dist}_{G_{\Phi}}(c_1, c_2) \geq 3.$$ 

Thus, we have finished showing that $T \in \mathcal{D}(G_{\Phi})$.

Now, observe the following lower bound on the size of $T$:

$$|T| = |I| + |\mathcal{F}(\rho) \cap C_{\text{bad}}| + 1 \geq \frac{|\mathcal{F}(\rho)| - |\mathcal{F}(\rho) \cap C_{\text{bad}}| - k\Delta}{k\Delta} + |\mathcal{F}(\rho) \cap C_{\text{bad}}| + 1 \geq \frac{|\mathcal{F}(\rho)|}{k\Delta} \geq V_T(\rho) \geq C_0[\log(n/\varepsilon)],$$

where in the second to last inequality we used Item (2) of Lemma 56. If $|T| > C_0[\log(n/\varepsilon)]$, we make the size of $T$ exactly equal to $C_0[\log(n/\varepsilon)]$ by removing some clauses from it. Note that any subset of $T$ satisfies Item (1) of the definition of $\mathcal{D}(G_{\Phi})$ (cf. Definition 27). To maintain the connectedness of $T$ in $G_{\Phi}^{\leq 4}$, consider an arbitrary spanning tree of the subgraph $G_{\Phi}^{\leq 4}[T]$; remove leaf vertices of the tree from $T$ until until $|T| = C_0[\log(n/\varepsilon)]$. By construction, the remaining $T$ is still connected in $G_{\Phi}^{\leq 4}$ and hence is in $\mathcal{D}(G_{\Phi})$.

Finally, we show that $|T \cap C_{\text{bad}}| \leq \frac{|T|}{12}$. Since $T$ is connected in $G_{\Phi}^{\leq 4}$, Lemma 53 implies that there exists a connected induced set $T'$ of vertices of $G_{\Phi}$ such that $T \subseteq T'$ and $|T'| \leq 4|T|$. Lemma 57 implies that $|\text{var}(T')| \geq 0.9k |T'| \geq 0.9k|T| = 0.9kC_0[\log(n/\varepsilon)] > 21600k \log n$. Thus Lemma 51 implies that $|T' \cap C_{\text{bad}}| \leq \frac{|T'|}{12}$. We conclude that

$$|T \cap C_{\text{bad}}| \leq |T' \cap C_{\text{bad}}| \leq \frac{|T'|}{12} \leq \frac{|T|}{12} \leq \frac{|T|}{3}. $$

This completes the proof. 

8. Proof of Theorem 1

In order to estimate $\Omega(\Phi)$, we use self-reducibility to calculate the marginal probability of the partial assignment $\Lambda^*$ from Lemma 10. This marginal probability is $|\Omega^\Lambda|/|\Omega|$. By Lemma 49, w.h.p. over the choice of $\Phi$, $|\Omega^\Lambda|$ can be computed in polynomial time. This is because $\Lambda^*$ satisfies all of the good clauses (by Lemma 10). The remaining clauses are bad, and Lemma 49 guarantees that all bad components have $O(\log n)$ size, so their satisfying assignments can be counted by brute force.

**Lemma 58.** There is a deterministic algorithm that takes as input a $k$-CNF formula $\Phi$ on $n$ Boolean variables with $m$ clauses, a partial assignment $\Lambda$ of $\Phi$ and an accuracy parameter $\varepsilon > 0$. It returns a rational value $p$ and runs in time $\text{poly}(n, 1/\varepsilon)$. If $\Phi = \Phi(k, n, m)$ then, w.h.p., the guarantees of all of our lemmas apply. In this case, as long as $\Lambda$ is a prefix of the partial assignment $\Lambda^*$ from Lemma 10, the output satisfies $e^{-\varepsilon/n}p \leq |\Omega^\Lambda|/|\Omega^\Lambda_2| \leq e^{\varepsilon/n}p$. 

**Proof.** Let $v^*$ be the first unassigned variable in $\Lambda^* \setminus \Lambda$. By Lemma 14, the depth of $T^\Lambda$ is at most $3k^2\alpha L + 1 = O(\log(n/\varepsilon))$. Thus, the number of nodes of $T^\Lambda$ is bounded by a polynomial in $n/\varepsilon$. 

28
After constructing $T^A$, the algorithm constructs the linear program from Section 5. Lemma 35 guarantees that, given bounds $r_{\text{lower}} \leq r_{\text{upper}}$, if the LP has a solution $P$ using $r_{\text{lower}}$ and $r_{\text{upper}}$, then $e^{-\varepsilon/(3n)}r_{\text{lower}} \leq |\Omega_1^A|/|\Omega_2^A| \leq e^{\varepsilon/(3n)}r_{\text{upper}}$. Thus, the task of the algorithm is to find initial bounds $r_{\text{lower}} \leq r_{\text{upper}}$ for which there is a feasible solution and then (by binary search) to bring $r_{\text{lower}}$ and $r_{\text{upper}}$ closer together, to obtain a more accurate estimate. This is done in Algorithm 2.

The algorithm is based on the premise that there is a feasible solution with $r_{\text{lower}} = (3s - 1)/(3s + 1)$ and $r_{\text{upper}} = (3s + 1)/(3s - 1)$, so we next establish this fact. This follows from Lemma 9, which guarantees that

$$\frac{1 - \frac{1}{3s}}{1 + \frac{1}{3s}} < \frac{|\Omega_1^A|}{|\Omega_2^A|} < \frac{1 + \frac{1}{3s}}{1 - \frac{1}{3s}}.$$ 

**Algorithm 2 Estimate $|\Omega_1^A| / |\Omega_2^A|$**

1: $p_{\text{lower}} \leftarrow \frac{3s - 1}{3s + 1}$;
2: $p_{\text{upper}} \leftarrow \frac{3s + 1}{3s - 1}$;
3: while $p_{\text{upper}} > e^{\varepsilon/(3n)}p_{\text{lower}}$ do
4: \hspace{1em} $r_{\text{lower}} \leftarrow p_{\text{lower}}$;
5: \hspace{1em} $r_{\text{upper}} \leftarrow (p_{\text{lower}} + p_{\text{upper}})/2$;
6: \hspace{1em} if the LP described in Section 5 has a feasible solution then
7: \hspace{2em} $p_{\text{upper}} \leftarrow r_{\text{upper}}$;
8: \hspace{2em} else
9: \hspace{3em} $p_{\text{lower}} \leftarrow r_{\text{upper}}$;
10: \hspace{1em} end if
11: end while
12: return $p \leftarrow (p_{\text{lower}} + p_{\text{upper}})/2$;

We next explore the accuracy of the output. Applying Lemma 25 and Lemma 35, after each step of the while loop of Algorithm 2, it holds that $e^{-\varepsilon/(3n)}p_{\text{lower}} \leq |\Omega_1^A| / |\Omega_2^A| \leq e^{\varepsilon/(3n)}p_{\text{upper}}$. Also, $p_{\text{upper}} \leq e^{\varepsilon/(3n)}p_{\text{lower}}$ when the algorithm terminates, so the output $p$ satisfies $e^{-\varepsilon/np} \leq |\Omega_1^A| / |\Omega_2^A| \leq e^{\varepsilon/np}$.

Finally, we explore the running time of the algorithm. Since each iteration of the while loop takes polynomial time, the main issue is determining how many times the while loop executes. Note that in the beginning $p_{\text{upper}} - p_{\text{lower}} \leq 4/(3s - 1)$, and just before the last iteration, $p_{\text{upper}} - p_{\text{lower}} \geq (e^{\varepsilon/(3n)} - 1)p_{\text{lower}} \geq \frac{\varepsilon(3s - 1)}{3n(3s + 1)}$. This difference halves with each iteration. Thus the algorithm solves the LP at most $O(\log(n/\varepsilon))$ times. Since the size of LP is bounded by a polynomial in $n/\varepsilon$ and the LP can be solved in polynomial time, the whole of the algorithm runs in polynomial time, as required. \hfill $\Box$

We now prove Theorem 1.

**Theorem 1.** There is a polynomial-time algorithm $A$ and two constants $r > 0, k_0 \geq 3$ such that for all $k \geq k_0$ and all $\alpha < 2^k$ the following holds w.h.p. over the choice of the random $k$-SAT formula $\Phi = \Phi(k, n, m)$. The algorithm $A$, on input the formula $\Phi$ and a rational $\varepsilon > 0$, outputs in time $\text{poly}(n, 1/\varepsilon)$ a number $Z$ that satisfies $e^{-\varepsilon}|\Omega(\Phi)| \leq Z \leq e^{\varepsilon}|\Omega(\Phi)|$.

**Proof.** Let $\Phi = \Phi(k, n, m)$ be a random formula and $\Omega = \Omega(\Phi)$. The algorithm first computes $V_{\text{good}}, V_{\text{bad}}, C_{\text{good}}$, and $C_{\text{bad}}$ in time $\text{poly}(n)$. Then using Lemma 8 and Lemma 10, it can compute $V_{\text{marked}}$ and $\Lambda^*$ in polynomial-time. W.h.p. over the choice of $\Phi$, the guarantees of all of our lemmas apply. Let us suppose that this happens (otherwise, the algorithm fails and outputs an arbitrary number).

29
Suppose \( \Lambda^* \) gives values to the \( j \) variables \( v_1, v_2, \ldots, v_j \). By Lemma 58, for any \( i \in \{1, \ldots, j\} \) and any prefix \( \Lambda_i: \{v_1, v_2, \ldots, v_{i-1}\} \rightarrow \{T, F\} \) of \( \Lambda^* \), it takes \( \text{poly}(n, 1/\varepsilon) \) time to compute a number \( p_i \) satisfying \( e^{-\varepsilon/n} p_i \leq |\Omega_1^\Lambda| / |\Omega_2^\Lambda| \leq e^{\varepsilon/n} p_i \). If \( \Lambda^*(v_i) = T \) let \( q_i = \frac{1}{1 + p_i} \). Otherwise, let \( q_i = \frac{1}{1 + p_i} \). Let \( \Lambda_{j+1} = \Lambda^* \).

Suppose that \( \Lambda^*(v_i) = T \). Then
\[
\frac{|\Omega_1^\Lambda|}{|\Omega_{\Lambda^{i+1}}|} = \frac{|\Omega_2^\Lambda|}{|\Omega_1^\Lambda|} = 1 + \frac{|\Omega_2^\Lambda|}{|\Omega_1^\Lambda|} \leq 1 + \frac{e^{\varepsilon/n}}{p_i}.
\]

Thus,
\[
\frac{|\Omega_{\Lambda^{i+1}}|}{|\Omega^\Lambda|} \geq \frac{p_i}{p_i + e^{\varepsilon/n}} \geq \frac{p_i}{e^{\varepsilon/n} p_i} = e^{-\varepsilon/n} q_i.
\]

A similar calculation for the case where \( \Lambda^*(v_i) = F \) and a similar calculation for the lower bound give the following.
\[
e^{-\varepsilon/n} q_i \leq \frac{|\Omega_{\Lambda^{i+1}}|}{|\Omega^\Lambda|} \leq e^{\varepsilon/n} q_i,
\]

and thus
\[
e^{-\varepsilon} \prod_{i=1}^j q_i \leq \frac{|\Omega^\Lambda|}{|\Omega|} \leq e^\varepsilon \prod_{i=1}^j q_i.
\]

Since all good clauses are satisfied by \( \Lambda^* \), \( \mathcal{C}^\Lambda^* \) consists only of bad clauses. Also, by Lemma 49, every bad component of variables has size at most \( 21600k \log n \), so \( \mathcal{C}^\Lambda^* \) can be divided into disjoint subsets where each subset of clauses contains \( O(\log n) \) variables. The algorithm can compute the number of satisfying assignments of each subset by brute force in time \( \text{poly}(n) \). Then \( |\Omega^\Lambda^*| \) is the product of these numbers.

Combining all of the above, our algorithm outputs \( Z = |\Omega^\Lambda^*| \prod_{i=1}^j q_i \), which satisfies \( e^{-\varepsilon} |\Omega| \leq Z \leq e^\varepsilon |\Omega| \). \( \square \)

**APPENDIX A. NOTATION REFERENCE**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formula related</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \mathcal{C} )</td>
<td>The clause set of ( \Phi ) where ( m =</td>
<td>\mathcal{C}</td>
</tr>
<tr>
<td>( \mathcal{C}<em>{\text{good}}, \mathcal{C}</em>{\text{bad}} )</td>
<td>Good and bad clauses, a partition of ( \mathcal{C} )</td>
<td>Section 2</td>
</tr>
<tr>
<td>( \mathcal{V} )</td>
<td>The variable set of ( \Phi ) where ( n =</td>
<td>\mathcal{V}</td>
</tr>
<tr>
<td>( \mathcal{V}<em>{\text{good}}, \mathcal{V}</em>{\text{bad}} )</td>
<td>Good and bad variables, a partition of ( \mathcal{V} )</td>
<td>Section 2</td>
</tr>
<tr>
<td>( \Delta )</td>
<td>The high degree threshold, set to ( 2k/300 ).</td>
<td>Definition 2</td>
</tr>
<tr>
<td>( G_\Phi )</td>
<td>The dependency graph of ( \mathcal{C} ), which contains a subgraph ( G_{\Phi, \text{good}} ).</td>
<td>Definition 4</td>
</tr>
<tr>
<td>( H_\Phi )</td>
<td>The dependency graph of ( \mathcal{V} ), which contains a subgraph ( H_{\Phi, \text{bad}} ).</td>
<td>Definition 5</td>
</tr>
<tr>
<td>( \mathcal{D}(G_\Phi) )</td>
<td>A set of subsets ( T \subseteq V(G_\Phi) ) satisfying some properties</td>
<td>Definition 27</td>
</tr>
<tr>
<td><strong>Local lemma</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \Omega^\ast )</td>
<td></td>
<td>Section 3</td>
</tr>
<tr>
<td>( \mu_A )</td>
<td></td>
<td>Section 3</td>
</tr>
<tr>
<td>( \text{marked}(c) )</td>
<td></td>
<td>Section 3</td>
</tr>
<tr>
<td>( V_{\text{marked}} )</td>
<td></td>
<td>Section 3</td>
</tr>
<tr>
<td>( \Omega )</td>
<td></td>
<td>Section 3</td>
</tr>
<tr>
<td>( s )</td>
<td>Local uniformity parameter ( s := 2^{k/4}/(ek\Delta) ).</td>
<td>Section 3</td>
</tr>
<tr>
<td>( \Lambda^* )</td>
<td>A particular “nice” partial assignment.</td>
<td>Lemma 10</td>
</tr>
<tr>
<td>( \Phi^\Lambda )</td>
<td>The formula ( \Phi ) simplified under ( \Lambda )</td>
<td>Section 3</td>
</tr>
</tbody>
</table>
$C^\Lambda$ Remaining clauses under $\Lambda$. Similar notations include $V^\Lambda$, $C_{\text{good}}^\Lambda$. Note that $V_{\text{bad}}^\Lambda = V_{\text{bad}}$ and $C_{\text{bad}}^\Lambda = C_{\text{bad}}$. Section 3

**Coupling tree**

$L$ Truncation depth, set to $C_0(3k^2\Delta)[\log(n/\varepsilon)]$, where $C_0$ is a sufficiently large integer (independent of $\Phi$, $k$ and $n$). Definition 12

$L$ The set of leaves of the coupling tree. Definition 12

$T$ The set of truncating nodes of the coupling tree. Definition 12

$L^*$ $L^* = L \cup T$. Definition 12

**References**


(Andreas Galanis, Leslie Ann Goldberg, Kuan Yang) DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF OXFORD, OX1 3QD, UNITED KINGDOM.

Email address: \{andreas.galanis,leslie.goldberg,kuan.yang\}@cs.ox.ac.uk

(Heng Guo) SCHOOL OF INFORMATICS, UNIVERSITY OF EDINBURGH, INFORMATICS FORUM, EDINBURGH, EH8 9AB, UNITED KINGDOM.

Email address: hguo@inf.ed.ac.uk