A Dichotomy Theorem for the Resolution Complexity of Random Constraint Satisfaction Problems

Siu On Chan*
Department of EECS
UC Berkeley
siuon@cs.berkeley.edu

Michael Molloy
Department of Computer Science
University of Toronto
molloy@cs.toronto.edu

August 8, 2012

Abstract

We consider random instances of constraint satisfaction problems where each variable has domain size $O(1)$, each constraint is on $O(1)$ variables and the constraints are chosen from a specified distribution. The number of constraints is cn where c is a constant. We prove that for every possible distribution, either the resolution complexity is almost surely polylogarithmic for sufficiently large c, or it is almost surely exponential for every $c > 0$. We characterize the distributions of each type. To do so, we introduce a closure operation on constraint sets which yields the set of all constraints that, in some sense, appear implicitly in the random CSP.

1 Introduction

Constraint satisfaction problems (CSP's) form an active area of research in many areas of computer science. They generalize SAT by allowing variables to take values from a domain more general than $\{\text{true}, \text{false}\}$, and having more general restrictions on values jointly taken by variables in each clause. The widespread interest in random k-SAT has spread to its generalisations, such as random instances of 1-in-k-SAT [3], NAE-k-SAT [3, 5], k-XOR-SAT [19, 33] and $(2+p)$-SAT [4]. All of these can be expressed as CSP's. As a result, the interest has spread to random instances of CSP's, rigorously in e.g. [37, 20, 36, 35] and experimentally even earlier (see [25] for a survey).

Unsatisfiability of k-SAT and CSP instances can be demonstrated by resolution proof systems, and many natural algorithms for k-SAT and CSP can be simulated as resolution proof procedures. In fact, virtually every complete[1] SAT-solver or CSP-solver used in practice is resolution based. The running time of any such algorithm is lower-bounded by the resolution complexity of the input. In a seminal paper [15], Chvátal and Szemerédi consider the resolution complexity of random k-SAT formulas, $k \geq 3$; i.e. the asymptotic order of the length of a shortest resolution refutation. As the clause-variable ratio c grows, the resolution complexity decreases monotonically, but is still almost surely[2] (a.s.) exponential for any constant c. This explained the empirical observation that

*This work is part of the first author's master research paper prepared at University of Toronto under the supervision of the second author. A preliminary version appeared in the proceedings of FOCS 2008.

[1] A solver is *complete* if it can recognize every satisfiable and every unsatisfiable input.

[2] We say that an event A occurs *almost surely* if $\lim_{n\to\infty} \mathbb{P}(A) = 1$.

SAT-solvers take a very long time on these formulas when the number of clauses is a bit higher than (what appears to be) the threshold at which the formula is a.s. unsatisfiable [34]. Their result has been generalized and extended in many directions, to super-constant clause-variable ratio [24, 11, 10], and to general classes of CSP's [36, 39]. There are also a.s. exponential lower bounds on the resolution complexity of specific graph problems, such as k-colorability and k-independent sets [9, 8], when the clause-variable ratio is an arbitrarily high constant.

In contrast, many other random models a.s. have at most polynomial resolution complexity when the clause-variable ratio is a sufficiently large constant; we call this property POLY. It is natural to ask which models of random CSP's have Property POLY, and for which models the resolution complexity a.s. remains exponentially high for every constant clause-variable ratio. This question has been resolved for several specific models and families of models in the past (see below). Our main contribution is to resolve this question for *every* model from a very broad family which contains, in some sense, all random models with constant clause and domain size.

Practitioners have long been using random CSP's to gain insights into difficult problems. This paper may shed some insight into what can cause CSP's to have high resolution complexity, even for a very high linear number of constraints. It is well-understood that long paths in the CSP which constrain the joint assignments to pairs of distant variables can cause low resolution complexity; for example, this is what causes the resolution complexity of 2-SAT to be small. We introduce a path-like structure, called a *petal*, and prove that for a high linear *random* number of constraints, polynomial resolution complexity only occurs for models in which many long petals constrain the joint assignments of pairs of distant variables.

We study the family of models of random k-ary CSP's introduced by the second author[37] and independently, in slightly less generality, by Creignou and Daudé[18]. Roughly speaking, the models are as follows (formal definitions will appear in Section 1.2): One begins by randomly selecting k-tuples of n variables on which to place a constraint. Then, for each chosen k-tuple, one chooses a random constraint. The distribution $\mathcal{P}$ from which this random constraint is chosen is what specifies the model. For example, with one distribution, the model is random k-SAT, with another it is random k-XOR-SAT and with yet another it is d-colorability of random graphs. We denote a random instance by $\mathsf{CSP}_{n,M}(\mathcal{P})$, where M is the number of k-tuples selected.

The main result of this paper is a characterization of exactly which models have property POLY. Denote by $\operatorname{supp}\mathcal{P}$ the *support* of $\mathcal{P}$, i.e. $\operatorname{supp}\mathcal{P} = \{C \mid \mathcal{P}(C) > 0\}$. Informally, $\operatorname{supp}\mathcal{P}$ is the set of "types" of constraints that appear in the random CSP. It turns out that whether POLY holds for $\mathsf{CSP}_{n,M}(\mathcal{P})$ depends only on the set $\operatorname{supp}\mathcal{P}$ and not on the actual distribution over $\operatorname{supp}\mathcal{P}$.

For a particular $\mathcal{P}$, let $\mathcal{C}$ denote $\operatorname{supp}\mathcal{P}$. It is a bit deceptive to focus only on $\mathcal{C}$. The reason is that long paths of constraints can induce a constraint, C', on their endpoints. If those endpoints lie in a constraint $C \in \mathcal{C}$ then, in effect, they are constrained by the more restrictive constraint: $C' \cup C$. So we determine all constraints C that are *likely* to be induced by long paths, and for each, we form $\mathcal{C}'$ by adding $C' \cup C$ for each $C \in \mathcal{C}$. Naturally, we need to iterate until we reach what we call the *closure* of $\mathcal{C}$, $\operatorname{cl}(\mathcal{C})$. We say that $\operatorname{cl}(\mathcal{C})$ is *complete* if it contains the unsatisfiable constraint; i.e. the constraint that forbids every k-tuple of values. These definitions appear more formally in Section 2.

This closure may be applicable to other problems regarding random CSP's, since it contains the constraints which appear *implicitly* in the CSP as opposed to the constraints of $\mathcal{C}$ that appear *explicitly*. For the purposes of this paper, focussing on $\operatorname{cl}(\mathcal{C})$ rather than $\mathcal{C}$ yields a simple characterization of those $\mathcal{P}$ that have the property POLY:

Theorem 1.1. *If* $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ *is complete then there exists constants* c *and* a *such that a.s. the resolution complexity of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *is* $O(\log^a n)$.

Theorem 1.2. *If* $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ *is incomplete then for every* $c > 0$, *there exists* $\zeta > 0$ *such that with uniformly positive probability*[3] *(w.u.p.p.) the resolution complexity of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *is at least* $2^{\zeta n}$.

The above two theorems together form a dichotomy theorem. Given $\mathcal{P}$ it is easy to determine $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ and hence to decide POLY in finite time (see Lemma 3.15).

Note that Theorem 1.1 is stronger than what we aimed for: It implies not only a.s. polynomial, but a.s. *polylogarithmic*, resolution complexity. So we have the following interesting corollary:

Corollary 1.3. *For every* $\mathcal{P}$*: if for* c *sufficiently large,* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *a.s. has subexponential resolution complexity then for* c *sufficiently large,* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *a.s. has polylogarithmic resolution complexity.*

For the case where POLY does not hold, i.e. the case covered by Theorem 1.2, we determine whether $\mathsf{CSP}_{n,M}(\mathcal{P})$ in fact a.s. (rather than just w.u.p.p.) has exponential resolution complexity. A *cyclic* CSP is a CSP whose constraint hypergraph forms a cycle (the formal definition appears in Subsection 2). See Definition 2.6 for the meaning of "null-constraining".

Theorem 1.4. *Suppose* $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ *is incomplete.*

(a) If for some null-constraining subdomain $\mathcal{D}'$*, every cyclic CSP formed from* $\mathrm{supp}\,\mathcal{P}$ *is satisfiable using only values from* $\mathcal{D}'$ *then for every* $c > 0$, *there exists* $\zeta > 0$ *such that a.s. the resolution complexity of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *is at least* $2^{\zeta n}$*;*

(b) else there exists constants c *and* a *such that w.u.p.p. the resolution complexity of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *is* $O(\log^a n)$.

So in case (b) there is some $\epsilon > 0$ such that with probability at least ϵ, $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ has at most polylogarithmic resolution complexity and with probability at least ϵ, $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ has exponential resolution complexity. The proof implies that, in this case, small resolution complexity must be caused by problematic cycles of length $O(1)$.

Having small resolution complexity by itself does not imply that a resolution proof can be found efficiently. For example, a resolution proof of logarithmic size may take quasipolynomial time to locate by exhaustive search. However, the resolution proofs that appear in our proofs are structured and can be found quickly.

Theorem 1.5. *There is a polynomial time algorithm which, for sufficiently large* c*:*

(a) will a.s. find a $\mathrm{poly}(\log n)$ *resolution refutation of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *under the condition of Theorem 1.1;*

(b) will w.u.p.p. find a $\mathrm{poly}(\log n)$ *resolution refutation of* $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ *under the condition of Theorem 1.4(b).*

The above $\mathrm{poly}(\log n)$ *terms depend only on* k, d *and the distribution* $\mathcal{P}$.

In the course of proving Theorem 1.1, we study a certain convergence property of random walks on general directed graphs (Theorem 4.2). This result may be of independent interest.

[3] We say that an event occurs *with uniformly positive probability* if $\liminf \mathbb{P}(A) > 0$.

1.1 Related Work

The first result along these lines was by Chvátal and Szemerédi [15] who proved that random 3-SAT a.s. has exponentially high resolution complexity for every constant c; i.e. it does not have property POLY. This result was extended to the case where c grows with n in Beame and Pitassi[11] and Beame et al.[10] and the proof was simplified greatly by Ben-Sasson and Wigderson in [12] where they introduced their Width Lemma (which we use here). Achlioptas et al [2] began with the easy observation that random $(2+p)$-SAT, a mixture of random 2-SAT and random 3-SAT has polynomial resolution complexity if the number of clauses is so high that the 2-clauses alone are unsatisfiable; thus random $(2+p)$-SAT has POLY. They then proved that for any smaller clause-density, the resolution complexity is exponentially high, thus establishing a sharp threshold for exponential resolution complexity in this model. They also showed how this can explain the empirical observation that resolution-based SAT-solvers have a difficult time with random 3-SAT slightly *below* the generally conjectured value of the satisfiability threshold (see also [1] for random k-SAT with $k > 3$).

Mitchell [36, 35] extended the (by then standard) techniques for proving such theorems about random k-SAT to the more general setting of random CSP's. He used these techniques to study the (d, k, t)-model – where the domain size is d and the constraints are uniformly random amongst those with k variables and t restrictions. He proved that for a wide range of triples (d, k, t), the model does not have property POLY. Molloy and Salavatipour [39] determined precisely which triples (d, k, t) have property POLY; moreover, for those that do have POLY they determined a sharp threshold for exponential resolution complexity. See also [8, 16] for other examples of specific models of random CSP's that are shown to not have property POLY.

All of these models are specific instances of the general family of models considered in this paper. Thus Theorems 1.1 and 1.2 imply all the results described above, except for those that actually determine the sharp threshold for exponential resolution complexity and those where c is superlinear.

There is also a body of work studying some random CSP's where the constraint-sizes and/or the domains grow with n (see e.g. [42, 23, 20, 22]). Such models do not fall into our general family (in fact, they are very different - see Remark 1.7), so this paper says nothing about them. For example, our theorems do not imply the resolution lower bounds in [42].

1.2 The Random Model

We use the family of models introduced by the second author in [37]. The same family, in a slightly less general form, was introduced independently by Creignou and Daudé [18]. The variables of our problem all have the same domain of permissible values, $\mathcal{D} = \{1, \ldots, d\}$, and all constraints will have size k, for some fixed integers d, k. Given a k-tuple of variables, $(x_1, \ldots, x_k)$, a *restriction* on $(x_1, \ldots, x_k)$ is a k-tuple of values $(\delta_1, \ldots, \delta_k)$ where each $\delta_i \in \mathcal{D}$. A set of restrictions on a k-tuple $(x_1, \ldots, x_k)$ is called a *constraint*. A *constraint satisfaction problem* (CSP) consists of a domain-size d, a constraint-size k, a collection of variables, and a set of constraints on k-tuples of those variables. We say that an assignment of values to the variables of a constraint C *satisfies* C if that assignment is not one of the restrictions on C. An assignment of values to all variables in a CSP satisfies that CSP if every constraint is simultaneously satisfied.

The *constraint hypergraph* of a CSP is the k-uniform hypergraph whose vertices correspond to the variables, and whose hyperedges correspond to the k-tuples of variables which have *constraints*.

Of course, when $k = 2$, the constraint hypergraph is simply a graph, and so we often call it the *constraint graph*.

It will be convenient to consider a set of canonical variables $X_1, \ldots, X_k$ which are used only to describe the "pattern" of a constraint. These canonical variables are not variables of the actual CSP. For any d, k there are d^k possible restrictions and 2^{d^k} possible constraints over the k canonical variables. We denote this set of constraints as $\mathcal{C}^{d,k}$. For our random model, one begins by specifying a particular probability distribution, $\mathcal{P}$, over $\mathcal{C}^{d,k}$. Different choices of $\mathcal{P}$ give rise to different instances of the model.

The Random Model: Specify M, n and $\mathcal{P}$ (typically $M = cn$ for some constant c; note that $\mathcal{P}$ implicitly specifies d, k). First choose a random constraint hypergraph with M hyperedges, in the usual manner; i.e., where each k-uniform hypergraph with n vertices and M hyperedges is equally likely. (To be clear: each hyperedge contains k distinct vertices, and no two hyperedges contain the same k vertices.) Next, for each hyperedge e, we choose a constraint on the k variables of e as follows: we take a random permutation from the k variables onto $\{X_1, \ldots, X_k\}$ and then we select a random constraint according to $\mathcal{P}$, mapping it onto a constraint on our k variables in the obvious manner. We use $\mathsf{CSP}_{n,M}(\mathcal{P})$ to denote a random CSP drawn from this model with parameters $n, M, \mathcal{P}$.

Remark 1.6. We could have chosen the constraint hypergraph by making an independent choice for each potential hyperedge, deciding to put it in the hypergraph with probability $p = \frac{c \times k!}{n^{k-1}}$. We use $\mathsf{CSP}_{n,p}(\mathcal{P})$ to denote such a random CSP. $\mathsf{CSP}_{n,p}(\mathcal{P})$ is, in many senses, equivalent to the model described above. In particular, standard arguments (see eg. Section 1.4 of [26]) show that the theorems in this paper translate to this alternate model, and allow us to often move back and forth between the models. We will make use of this equivalence in the proofs of Lemmas 3.22 and 3.23, where we analyze $\mathsf{CSP}_{n,p}(\mathcal{P})$ to prove things about $\mathsf{CSP}_{n,M}(\mathcal{P})$.

A constraint set $\mathcal{C}$ is *symmetric* if for any permutation σ of $\{1, \ldots, k\}$, any $C \in \mathcal{C}$, we have $\tilde{\sigma}(C) \in \mathcal{C}$, where $\tilde{\sigma}$ is the map induced by σ with the obvious definition: $\tilde{\sigma}(C) = \{(\delta_{\sigma(1)}, \ldots, \delta_{\sigma(k)}) \mid (\delta_1, \ldots, \delta_k) \in C\}$. Since the random model takes a random permutation from the k variables in a hyperedge to the k canonical variables before selecting the constraint, $\mathcal{P}$ can be assumed to be symmetric, i.e. $\mathcal{P}(C) = \mathcal{P}(\tilde{\sigma}(C))$ for all $\tilde{\sigma}$ and all C.

Remark 1.7. When d and/or k grow with n, the satisfiability threshold will typically occur at a superlinear number of constraints (see e.g. [42, 23, 22]). The structure of the constraint hypergraph in that case is very different than that of one with a linear number of constraints. This is why we restrict our attention to the case $d, k = O(1)$.

1.3 Resolution Complexity

The *resolution complexity* of a boolean CNF-formula ϕ, denoted $\mathbf{RES}(\phi)$, is the length of the shortest resolution proof that ϕ is unsatisfiable. (If ϕ is satisfiable, then $\mathbf{RES}(\phi) = \infty$.) Mitchell [35] discusses two natural ways to extend the notion of resolution complexity to the setting of CSP, **C-RES** and **NG-RES**. All commonly used resolution-type CSP algorithms correspond nicely to the **C-RES** complexity of the input, but there are some that do not correspond to the **NG-RES** complexity. For that reason, we focus in this paper on the **C-RES** complexity, as did Mitchell in [35], but our results also translate to **NG-RES** complexity - see Remarks 3.26, 5.11. Given an instance $\mathcal{I}$ of a CSP, Mitchell constructs an equivalent boolean CNF-formula CNF($\mathcal{I}$)

in a specific natural manner (Definition 1.8), and defines the resolution complexity **C-RES**($\mathcal{I}$) = **RES**(CNF($\mathcal{I}$)).

Definition 1.8. Given an instance $\mathcal{I}$ of a CSP in which every variable has domain $\{1, \ldots, d\}$, we construct a boolean CNF-formula CNF($\mathcal{I}$) as follows. For each variable x of $\mathcal{I}$, there are d variables in CNF($\mathcal{I}$), denoted $x : 1, \ldots, x : d$, and there is a *domain clause* $(x : 1 \vee \cdots \vee x : d)$. For each restriction $(\delta_1, \ldots, \delta_k)$ on variables $(x_1, \ldots, x_k)$, in any constraint of $\mathcal{I}$, CNF($\mathcal{I}$) has a *conflict clause* $(\overline{x_1 : \delta_1} \vee \cdots \vee \overline{x_k : \delta_k})$.

It is easy to see that CNF($\mathcal{I}$) has a satisfying assignment iff $\mathcal{I}$ does - if $\mathcal{I}$ has a satisfying assignment, then we produce one for CNF($\mathcal{I}$) by setting $x : \delta$ to true iff $x = \delta$; if CNF($\mathcal{I}$) has a satisfying assignment, then we produce one for $\mathcal{I}$ by setting $x = \delta$ where δ is any one of the values for which $x : \delta$ is true.

It is natural to consider adding an extra set of constraints for each variable x which specify that $x : \delta$ can be true for at most one value of δ. But it is easily verified that each of the results of this paper (in particular, Lemma 5.8) holds regardless of whether we include these clauses; to be specific, we do not include them.

2 The Closure Operation

In this section, we formally define the closure of a constraint set. Then we characterize those constraint sets which have a complete closure in terms of the existence of a subdomain of values which easily satisfies long paths. Such a subdomain will be shown to cause exponential resolution complexity. We begin with some definitions.

A constraint C on variables $x_1, \ldots, x_k$ *permits* $(x_i : \delta, x_j : \gamma)$ if at least one of the d^{k-2} possible tuples $(\delta_1, \ldots, \delta_k)$ with $\delta_i = \delta$ and $\delta_j = \gamma$ is not a restriction of C. Otherwise C *forbids* $(x_i : \delta, x_j : \gamma)$. The constraint $\{(\delta_1, \ldots, \delta_k) \in \mathcal{D}^k \mid \delta_i = \delta \wedge \delta_j = \gamma\}$ forbidding precisely $(x_i : \delta, x_j : \gamma)$ is called the $(x_i : \delta, x_j : \gamma)$-*forbidder* and is denoted $F(x_i : \delta, x_j : \gamma)$.

A *path* of length r in a hypergraph H is a sequence $\langle x_0, \ldots, x_r \rangle$ of distinct vertices together with a sequence $\langle e_1, \ldots, e_r \rangle$ of edges such that (1) the edges e_i are mutually vertex disjoint except at $\{x_1, \ldots, x_{r-1}\}$; (2) among $\{x_0, \ldots, x_r\}$, the only vertices in e_i are x_{i-1} and x_i, for $1 \leq i \leq r$. $x_0, \ldots, x_r$ are the *connecting variables* and x_0, x_r are the *endpoints* of P.[4] A *cycle* is defined the same way as a path with the exception that $x_0 = x_r$. The *length* of a path P, $|P|$, is the number of hyperedges.

A *pendant path* of length r in a hypergraph H is a path in which no vertices other than the endpoints lie in any edges of H off the path. In other words, there is no restriction on the degrees on the endpoints, each other connecting variable has degree 2 in H, and every other vertex in the path has degree 1 in H. (The *degree* of a vertex is the number of hyperedges in which it lies.)

A (pendant) path P of length r in a CSP is a sequence of r constraints whose underlying edges form a (pendant) path of length r in the constraint hypergraph. A *path over* $\mathcal{C}$ is a path whose constraints all lie in the constraint set $\mathcal{C}$. To emphasize that it is a path formed by constraints, we sometimes refer to it as a *constraint path*. If P_1, P_2 are constraint paths, then P_1P_2 is their *concatenation*; i.e. the constraint path formed by identifying the last endpoint of P_1 with the first endpoint of P_2. For $i \geq 0$, P^i denotes the constraint path $PP \ldots P$ consisting of i concatenated copies of P. A constraint path is *empty* if it has length zero.

[4] We consider the endpoints as connecting variables, to simplify statements in counting arguments below.

If at least one assignment α to the variables of P satisfies all the constraints of P with $\alpha(x_0) = \delta$ and $\alpha(x_r) = \gamma$, we say that P *permits* $(x_0 : \delta, x_r : \gamma)$; otherwise P *forbids* $(x_0 : \delta, x_r : \gamma)$. Sometimes we say P permits/forbids (δ, γ), omitting the endpoints.

A *cyclic CSP* is a CSP whose constraint hypergraph is a cycle.

We now come to the key definitions. Some of them ($\sim_{\mathcal{C}}$, closure and completeness) will be motivated in the discussion preceding Example 3.1 in Section 3.

Definition 2.1. For any $\mathcal{D}' \subset \mathcal{D}$, an assignment α of a CSP $\mathcal{J}$ is a $\mathcal{D}'$*-assignment* if it only uses values in $\mathcal{D}'$. $\mathcal{J}$ is $\mathcal{D}'$*-satisfiable* if it is satisfied by some $\mathcal{D}'$-assignment. A constraint path P with endpoints (x_0, x_r) $\mathcal{D}'$*-permits* a pair of values $(\delta, \gamma) \in \mathcal{D}'^2$ if there is a $\mathcal{D}'$-assignment α that satisfies all constraints of P and has $\alpha(x_0) = \delta$ and $\alpha(x_r) = \gamma$.

Definition 2.2. For a constraint set $\mathcal{C}$ and values $\delta, \gamma \in \mathcal{D}$, we write $\delta \sim_{\mathcal{C}} \gamma$ if there is some t such that every constraint path over $\mathcal{C}$ with length at least t permits (δ, γ).

Proposition 2.3. *For every $\mathcal{C}$, $\sim_{\mathcal{C}}$ is transitive. If $\mathcal{C}$ is symmetric then $\sim_{\mathcal{C}}$ is symmetric as well.*

Proof. Suppose $\delta_1 \sim_{\mathcal{C}} \delta_2$ and $\delta_2 \sim_{\mathcal{C}} \delta_3$. Then there are t_1, t_2 such that all paths of length at least t_1 permit (δ_1, δ_2), and all paths of length at least t_2 permit (δ_2, δ_3). Then all paths of length at least $t_1 + t_2$ permit (δ_1, δ_3), so $\sim_{\mathcal{C}}$ is transitive. If $\mathcal{C}$ is symmetric, then $\sim_{\mathcal{C}}$ is clearly symmetric, i.e. $\delta \sim_{\mathcal{C}} \gamma$ implies $\gamma \sim_{\mathcal{C}} \delta$. □

Definition 2.4. A symmetric constraint set $\mathcal{C}$ is *closed* if for any $\delta, \gamma \in \mathcal{D}$ such that $\delta \not\sim_{\mathcal{C}} \gamma$, any canonical variables X_i, X_j and any $C \in \mathcal{C}$, we have that $\mathcal{C}$ also contains the constraint obtained from C by forbidding $(X_i : \delta, X_j : \gamma)$. Formally, $C \cup F(X_i : \delta, X_j : \gamma) \in \mathcal{C}$. The *closure* $\mathrm{cl}(\mathcal{C})$ of a constraint set $\mathcal{C}$ is the smallest closed constraint set containing $\mathcal{C}$.

Given a constraint set $\mathcal{C}$, its closure $\mathrm{cl}(\mathcal{C})$ can be generated as follows: Let $\mathcal{C}_0 = \mathcal{C}$ and $h = 0$. We initially set $\mathcal{C}_{h+1} = \mathcal{C}_h$. For any $\delta \not\sim_{\mathcal{C}_h} \gamma$, any $C \in \mathcal{C}_h$ and any canonical variables X_i, X_j, add $C \cup F(X_i : \delta, X_j : \gamma)$ to $\mathcal{C}_{h+1}$. Then increase h and repeat. The sequence $\{\mathcal{C}_h\}$ cannot grow indefinitely, so $\mathcal{C}_{h'} = \mathcal{C}_{h'+1}$ for some h', and we have $\mathcal{C}_{h'} = \mathrm{cl}(\mathcal{C})$. Clearly, every closed set containing $\mathcal{C}$ must contain $\mathcal{C}_{h'}$ and so $\mathrm{cl}(\mathcal{C})$ is well-defined.

Definition 2.5. A closed, symmetric constraint set $\mathcal{C}$ is *complete* if $\delta \not\sim_{\mathcal{C}} \gamma$ for all $\delta, \gamma \in \mathcal{D}$. Equivalently, it is complete if it contains the constraint that forbids all d^k of the k-tuples. A constraint set which is not complete is *incomplete*.

The key lemma of this section is the following characterization of incomplete constraint sets. It says that a constraint set is incomplete precisely when, in some sense, long paths can impose no constraint on a particular subdomain of values. This subdomain is called *null-constraining*.

Definition 2.6. Given $\mathcal{C}$, a subdomain $\mathcal{D}' \subseteq \mathcal{D}$ is *null-constraining* if there is some t such that for every constraint path P over $\mathcal{C}$ with length at least t and every pair of values $\delta, \gamma \in \mathcal{D}'$, P $\mathcal{D}'$-permits (δ, γ).

Lemma 2.7. *Let $\mathcal{C}$ be closed and symmetric. $\mathcal{C}$ is incomplete iff some nonempty subdomain $\mathcal{D}' \subseteq \mathcal{D}$ is null-constraining.*

Proof. If some nonempty subdomain $\mathcal{D}'$ is null-constraining, then in particular $\delta \sim_{\mathcal{C}} \delta$ for every $\delta \in \mathcal{D}'$, so $\mathcal{C}$ is incomplete.

Suppose $\mathcal{C}$ is incomplete. There are $\delta_1, \delta_2 \in \mathcal{D}$ such that $\delta_1 \sim_{\mathcal{C}} \delta_2$ (possibly $\delta_1 = \delta_2$). Define $\mathcal{D}' = \{\delta \in \mathcal{D} \mid \delta_1 \sim_{\mathcal{C}} \delta\}$, which is nonempty. By Proposition 2.3, $\delta \sim_{\mathcal{C}} \gamma$ for all $\delta, \gamma \in \mathcal{D}'$; that is, there is a t such that every constraint path P of $\mathcal{C}$ with length at least t permits (δ, γ), possibly using some values from $\mathcal{D} \setminus \mathcal{D}'$ for non-endpoint variables. It remains to show that such a path P still permits (δ, γ) if values can only be chosen from $\mathcal{D}'$.

Claim 2.8. *Any constraint C of P can be replaced by a stronger constraint $C' \in \mathcal{C}$, such that if some variable x_i in C' takes a value from $\mathcal{D}'$, all other variables must take values from $\mathcal{D}'$ as well.*

Proof. Let C' be a superset of C that is maximal in $\mathcal{C}$, i.e. C' is not properly contained in any constraint in $\mathcal{C}$. Assume C' permits $(x_i : \delta, x_j : \gamma)$ for some $\delta \in \mathcal{D}'$, $\gamma \notin \mathcal{D}'$. We must have $\delta \not\sim_{\mathcal{C}} \gamma$, for otherwise $\delta \sim_{\mathcal{C}} \gamma$ and $\delta_1 \sim_{\mathcal{C}} \delta$ implies $\delta_1 \sim_{\mathcal{C}} \gamma$ and hence $\gamma \in \mathcal{D}'$, contradicting the assumption that $\gamma \notin \mathcal{D}'$. Since $\mathcal{C}$ is closed, $C' \cup F(x_i : \delta, x_j : \gamma) \in \mathcal{C}$. This contradicts the maximality of C'. □

By the above claim, we can replace every constraint in P by a stronger constraint from $\mathcal{C}$, none of which permits any $(\delta, \gamma) \in \mathcal{D}' \times (\mathcal{D} \setminus \mathcal{D}')$. The end result is a path P' over $\mathcal{C}$ of length at least t. Recall that all paths of length at least t permit (δ, γ) for any $\delta, \gamma \in \mathcal{D}'$. Therefore P' $\mathcal{D}'$-permits (δ, γ), hence so does P. □

Corollary 2.9. *If* $\mathrm{cl}(\mathcal{C})$ *is incomplete then some non-empty subdomain $\mathcal{D}' \subseteq \mathcal{D}$ is null-constraining in $\mathcal{C}$.*

Proof. Since $\mathcal{C} \subseteq \mathrm{cl}(\mathcal{C})$, any constraint path over $\mathcal{C}$ is also a constraint path over $\mathrm{cl}(\mathcal{C})$. Thus if $\mathcal{D}'$ is null-constraining for $\mathrm{cl}(\mathcal{C})$ then it is null-constraining for $\mathcal{C}$. So the corollary follows from Lemma 2.7. □

Proposition 2.10. *If there is a (δ, γ)-forbidding constraint path P over $\mathcal{C}$ with $|P| \geq 2^{|\mathcal{D}|}$ then there are infinitely many (δ, γ)-forbidding constraint paths over $\mathcal{C}$, one of which has length at most $2^{|\mathcal{D}|}$.*

Proof. Let the constraints of P be $\beta_1, \ldots, \beta_r$. Let $x_0 \in \beta_1, x_r \in \beta_i$ be the endpoints, and let $x_1, ..., x_{r-1}$ be the other connecting variables, where $x_i \in \beta_i \cap \beta_{i+1}$. Suppose x_0 takes the value δ, and let $\mathcal{D}_0 = \{\delta\}$. Define $\mathcal{D}_i$ to be the set of values that x_i can take without violating constraints in P, for $1 \leq i \leq r$. Then $\gamma \notin \mathcal{D}_r$ because P is (δ, γ)-forbidding. Since $r + 1 > 2^{|\mathcal{D}|}$, two $\mathcal{D}_i$'s coincide, i.e. $\mathcal{D}_i = \mathcal{D}_j$ for some $0 \leq i < j \leq r$. It is straightforward to observe that removing the constraints $\beta_{i+1}, \ldots, \beta_j$ and identifying x_i and x_j yields a shorter (δ, γ)-forbidding path. Such shortenings can be repeated until the resulting path has length at most $2^{|\mathcal{D}|}$. On the other hand, if we repeat the subpath between x_i and x_j many times, we can obtain arbitrarily long (δ, γ)-forbidding paths. □

Corollary 2.11. *If $\delta \not\sim_{\mathcal{C}} \gamma$ then there are constraint paths P_A, P_B, P_C over $\mathcal{C}$, with P_B non-empty, such that for every $i \geq 0$, $P_A P_B^i P_C$ is (δ, γ)-forbidding.*

Proof. If $\delta \not\sim_{\mathcal{C}} \gamma$, then there is a (δ, γ)-forbidding path of length greater than $2^{|\mathcal{D}|}$, by definition of $\sim_{\mathcal{C}}$. Now we follow the notation from the previous proof. P_A, P_B, P_C are the subpaths with constraints $(\beta_1, \ldots, \beta_i)$, $(\beta_{i+1}, \ldots, \beta_j)$, $(\beta_{j+1}, \ldots, \beta_r)$ respectively. □

Corollary 2.12. *$\delta \not\sim_{\mathcal{C}} \gamma$ if and only if there is a (δ, γ)-forbidding path of length t with $2^{|\mathcal{D}|} < t < 2 \cdot 2^{|\mathcal{D}|}$.*

Proof. The "if" part follows from Proposition 2.10.

For the "only if" part, when $\delta \not\sim_{\mathcal{C}} \gamma$, we get from Corollary 2.11 three subpaths P_A, P_B, P_C, such that $P_A P_B^q P_C$ are all (δ, γ)-forbidding. The final path $P_A P_C$, obtained after all constraint removals, must have length at most $2^{|\mathcal{D}|}$. We may also ensure $0 < |P_B| \leq 2^{|\mathcal{D}|}$ by removing constraints from (and identifying vertices in) P_B whenever $|P_B| > 2^{|\mathcal{D}|}$. Once this is done, $P_A P_B^q P_C$ will be a (δ, γ)-forbidding path of the desired length for an appropriate $q \geq 0$. □

Lemma 2.13. *If $\mathcal{C}$ is closed, symmetric and incomplete then $\mathcal{D}$ can be partitioned into $\mathcal{D}_1, ..., \mathcal{D}_t, W$ such that*

(a) each $\mathcal{D}_i$ is null-constraining, and

(b) for any $\delta, \gamma \in \mathcal{D}$ such that δ, γ don't both lie in the same $\mathcal{D}_i$, we have $\delta \not\sim_{\mathcal{C}} \gamma$.

Proof. Let $\mathcal{D}' \subseteq \mathcal{D}$ be a maximal null-constraining set; i.e. $\mathcal{D}'$ is null-constraining and if $\mathcal{D}' \subsetneq \mathcal{D}''$ then $\mathcal{D}''$ is not null-constraining. Consider any $\delta \in \mathcal{D}', \gamma \notin \mathcal{D}'$. We will argue that $\delta \not\sim_{\mathcal{C}} \gamma$. This implies that any two maximal null-constraining sets are disjoint, and implies the lemma by letting $\mathcal{D}_1, ..., \mathcal{D}_t$ be the maximal null-constraining sets. (Note that there is at least one such set by Lemma 2.7.)

To the contrary, suppose that there exists t_1 such that every constraint path over $\mathcal{C}$ of length at least t_1 permits (δ, γ). Since $\mathcal{D}'$ is null-constraining, there exists t_2 such that for each $\gamma' \in \mathcal{D}'$, every constraint path over $\mathcal{C}$ of length at least t_2 permits (γ, γ'). It follows that every constraint path over $\mathcal{C}$ of length at least $t_1 + t_2$ permits (δ, γ') - we can assign δ to the first variable, γ to the $(t_1 + 1)$th variable and γ' to the last. So $\delta \sim_{\mathcal{C}} \gamma'$ and by Proposition 2.3, $\gamma' \sim_{\mathcal{C}} \delta$. Thus $\mathcal{D}' \cup \{\delta\}$ is null-constraining, which contradicts the fact that $\mathcal{D}'$ is maximal null-constraining. □

We close this section by sketching how Lemma 2.7 implies some of our theorems. If a constraint set has complete closure, then for high clause-variable ratio, the CSP will a.s. contain a small unsatisfiable subproblem, causing polylogarithmic resolution complexity (see Section 3). This proves Theorem 1.1. If a constraint set has incomplete closure, then by Lemma 2.7, there is a nonempty null-constraining subdomain. This will cause exponential resolution complexity, provided there are no short unsatisfiable cycles in $\mathsf{CSP}_{n,M}(\mathcal{P})$ (see Section 5). This will imply Theorem 1.2.

3 Petals and Flowers

In this section, we consider distributions $\mathcal{P}$ for which $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ is complete. We will show that $\mathsf{CSP}_{n,M}(\mathcal{P})$ a.s. contains a small, structured, unsatisfiable subproblem, called a *forbidding flower*. This structured subproblem generalizes the flower from [39], which in turn was inspired by the snakes of [14].

3.1 Forbidding Flowers

A forbidding flower is a union of petals. Petals are recursive structures: Each petal functions like a (δ, γ)-forbidding path for appropriate (δ, γ), and subpetals may be attached to adjacent connecting variables along the main path of a petal.

Given a constraint set $\mathcal{C}$, subpetals are used to simulate constraints in $\mathrm{cl}(\mathcal{C}) \setminus \mathcal{C}$. Indeed, if $C \in \mathcal{C}$ is a constraint, x_i, x_j two of its variables, and P a (δ, γ)-forbidding path from x_i to x_j, then P

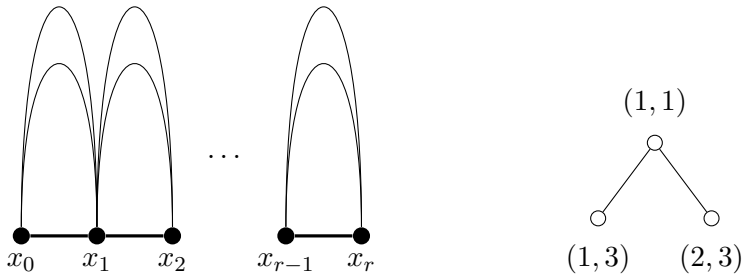

Figure 1: The left figure depicts the $(1,1)$-forbidding petal from Example 3.1. The thick line segment between x_i and x_{i+1} represents a C_1 constraint. Each of the thin paths above the thick line segment represents a path of C_2 constraints of length q. Note that the second thin path is redundant; it is only required so that this petal can be represented by a configuration tree. The right figure depicts that configuration tree (nodes' labels are shown next to the nodes).

essentially strengthens C to forbid $(x_i : \delta, x_j : \gamma)$ as well. More precisely, the constraint plus the path restricts (x_i, x_j) like $C \cup F(x_i : \delta, x_j : \gamma)$. Repeating this, we can simulate any $C \in \mathrm{cl}(\mathcal{C})$. These simulated constraints can then be used to simulate (δ, γ)-forbidding paths for any $\delta \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma$. If $\mathrm{cl}(\mathcal{C})$ is complete, then all (δ, γ)-forbidding paths can be simulated. If these simulated paths share two common endpoint variables x_1 and x_2, all assignments of x_1, x_2 are forbidden from being satisfying, thus yielding a small unsatisfiable CSP. It is straightforward to show that this CSP has a short resolution proof of unsatisfiability.

Example 3.1 (Simulated constraints)**.** Consider $\mathcal{D} = \{1,2,3\}, k = 2$ and $\mathcal{C} = \{C_1, C_2\}$, where $C_1 = \{(1,1),(2,2),(3,3)\}$ and $C_2 = (\{1,2\} \times \{3\}) \cup (\{3\} \times \{1,2\})$. It is easy to see that all constraint paths of length at least 2 permit $(1,1)$. On the other hand, we will show that $1 \not\sim_{\mathrm{cl}(\mathcal{C})} 1$. First note that every path made of the constraint C_2 forbids $(1,3)$, hence $1 \not\sim_{\mathcal{C}} 3$, implying $1 \not\sim_{\mathrm{cl}(\mathcal{C})} 3$. Every such path forbids $(2,3)$ as well, so similarly $2 \not\sim_{\mathrm{cl}(\mathcal{C})} 3$. By definition of $\mathrm{cl}(\mathcal{C})$, $C' = C_1 \cup \{(1,3),(2,3)\} \in \mathrm{cl}(\mathcal{C})$. Any path of odd length made of the constraint C' forbids $(1,1)$, so $1 \not\sim_{\mathrm{cl}(\mathcal{C})} 1$.

Note that odd paths made of C' do not exist in our CSP since $C' \notin \mathcal{C}$. However, our CSP can forbid $(1,1)$, using a different structure, depicted in Figure 3.1. Consider the path $P = \langle x_0, \ldots, x_r \rangle$ consisting of r copies of C_1, for some odd integer r. To every adjacent pair of connecting variables (x_i, x_{i+1}) in P, attach two paths from x_i to x_{i+1}, each consisting only of copies of C_2. The two paths along with C_1 effectively forbid $(x_i : 1, x_{i+1} : 3)$ and $(x_i : 2, x_{i+1} : 3)$.[5] The resulting graph P' forbids $(x_0 : 1, x_r : 1)$.

Petals are defined recursively to form structures like the one from Example 3.1 that forbid pairs of values from being assigned to their endpoints. At this point, some readers will find it easy to see how one can define a petal forbidding $(x_0 : \delta, x_r : \gamma)$ for any $\delta \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma$. To do so formally is a bit cumbersome. We start by introducing configuration trees and forests. These trees and forests

[5]For the purpose of forbidding $(x_i : 1, x_{i+1} : 3)$ and $(x_i : 2, x_{i+1} : 3)$, one actually only needs to attach a single "C_2 path" between x_i and x_{i+1}. We attach two such paths in order to be consistent with Definition 3.5.

serve only to describe the structure of the forbidding flowers; they are not actual subproblems in $\mathsf{CSP}_{n,M}(\mathcal{P})$.

Definition 3.2. A *configuration tree* $\mathcal{T}$ is a nonempty rooted tree, each of whose nodes v gets a label from $\mathcal{D}^2$ (i.e. an ordered pair of values in $\mathcal{D}$), call it $(\delta(v), \gamma(v))$.

Labels in a configuration tree $\mathcal{T}$ specify which pairs of values the petals and subpetals forbid. A petal forbids the label at height 0, i.e. the label of the root. Subpetals are attached to every adjacent pair of connecting variables along the main path of the petal. They forbid labels that are at height 1 in the configuration tree. These subpetals could in turn have subsubpetals along their main paths, forbidding values at height 2, and so on. As an example, P' in Example 3.1 corresponds to a configuration tree with three vertices; the root is labelled $(1,1)$ and its children are labelled $(1,3)$ and $(2,3)$ respectively (see Figure 1). In fact P' also forbids $(2,2)$, and thus also corresponds to another configuration tree with root $(2,2)$. The relation between configuration trees and petals is defined more formally in Definition 3.5.

Definition 3.3. Given a constraint path P with endpoints x_0, x_r and other connecting variables $x_1, ..., x_{r-1}$, and a subset $\Delta \subseteq \mathcal{D}^2$ of pairs of values, the path P *augmented with* Δ, denoted $P \cup \Delta$, is the constraint path with the same underlying hyperedges as P but where for each $0 \leq i \leq r-1$, the constraint C between (x_i, x_{i+1}) is strengthened to forbid all $(\delta, \gamma) \in \Delta$, i.e. C is replaced with $C \cup \bigcup_{(\delta,\gamma)\in\Delta} F(x_i : \delta, x_{i+1} : \gamma)$.

Notation 3.4. If a node v in a configuration tree has children $v_1, \ldots, v_s$, we use $\Delta(v)$ to denote the set of labels of v's children; i.e. $\Delta(v) = \{(\delta(v_i), \gamma(v_i)) \mid 1 \leq i \leq s\}$.

Definition 3.5. Given a configuration tree $\mathcal{T}$ and constraint set $\mathcal{C}$, a $(\mathcal{T}, \mathcal{C})$-*forbidding petal* is defined recursively as follows:

1. If $\mathcal{T}$ has only one vertex v labelled (δ, γ), a $(\mathcal{T}, \mathcal{C})$-forbidding petal is a (δ, γ)-forbidding path P_v over $\mathcal{C}$.

2. If $\mathcal{T}$ has more than one vertex, let v be its root, let $v_1, \ldots, v_s$ be v's children, and let $\mathcal{T}_1, \ldots, \mathcal{T}_s$ be the subtrees rooted at $v_1, \ldots, v_s$. Then a $(\mathcal{T}, \mathcal{C})$-forbidding petal consists of:

   (i) a path $P_v = \langle y_0, \ldots, y_r \rangle$ over $\mathcal{C}$, such that P_v augmented with the children's labels (i.e. $P_v \cup \Delta(v)$) is $(\delta(v), \gamma(v))$-forbidding; and

   (ii) for each $1 \leq j \leq s$: a $(\mathcal{T}_j, \mathcal{C})$-forbidding petal between every adjacent pair of connecting variables (y_i, y_{i+1}), $0 \leq i < r$.

P_v is the *main path* of the petal.

For any child v_i of v: each of the $(\mathcal{T}_j, \mathcal{C})$-forbidding petals in 2(ii) is a *subpetal* of the $(\mathcal{T}, \mathcal{C})$-forbidding petal, and it has *type* v_i. Every subpetal of each of these subpetals is also a subpetal of the $(\mathcal{T}, \mathcal{C})$-forbidding petal. The $(\mathcal{T}, \mathcal{C})$-forbidding petal itself is considered to be a subpetal of type v.

The main path of any subpetal is called a *petal path*. The *type* of the petal path is defined to be the type of the subpetal. Thus, the petal is the union of all of its petal paths.

The *endpoints* of the $(\mathcal{T}, \mathcal{C})$-forbidding petal are the endpoints of P_v; i.e. y_0, y_r.

This $(\mathcal{T}, \mathcal{C})$-forbidding petal is said to be *untangled* if (a) P_v and the $(\mathcal{T}_j, \mathcal{C})$-forbidding petals, $1 \leq j \leq s$ are mutually vertex-disjoint except at the endpoints of the petals, and (b) each of the $(\mathcal{T}_j, \mathcal{C})$-forbidding petals, $1 \leq j \leq s$ is untangled.

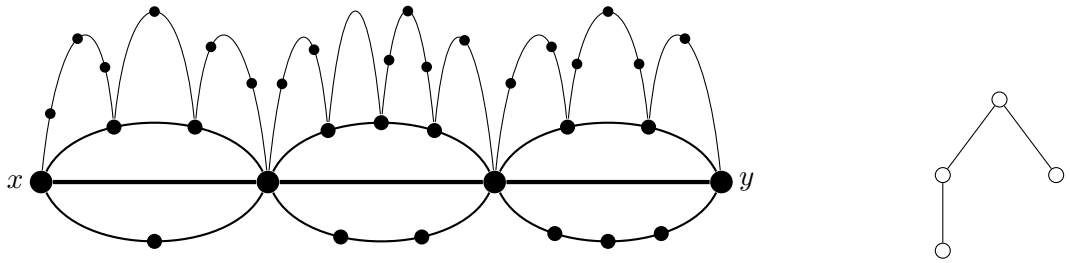

Figure 2: An example of a petal and its configuration tree (the labels are not included). Here, $k = 2$; i.e., the underlying hypergraph is a graph. The endpoints of the petal are x, y and the thick edges indicate the main path. Each pair of vertices that are adjacent in the main path form the endpoints of two petal paths. Each pair of vertices that are adjacent in the uppermost of those two petal paths form the endpoints of another petal path.

Remark 3.6. The following, less formal, description may be helpful. To build an untangled petal, start with a constraint path between y_0, y_r; this is the main path, and hence is a petal path. Then iteratively: choose any petal path P and add a new constraint path between each consecutive pair of connecting variables/endpoints on P; these are petal paths. The eventual result is that P will be strengthened to forbid more pairs of values at (y_0, y_ℓ). Moreover, the original path will become strengthened to the point where it forbids (δ, γ) at its endpoints (y_0, y_r). See Figure 2 for another example.

If $\mathrm{cl}(\mathcal{C})$ is complete, we shall argue later (Proposition 3.13) that forbidding petals exist for every $(\delta, \gamma) \in \mathcal{D}^2$. A forbidding flower is a collection of such petals, all with the same two endpoints. Formally:

Definition 3.7. A *configuration forest* $\mathcal{F} = \{\mathcal{T}_{(\delta,\gamma)}\}$ is a collection of $|\mathcal{D}|^2$ configuration trees, one with root labelled (δ, γ) for each $(\delta, \gamma) \in \mathcal{D}^2$.

Definition 3.8. Given a configuration forest $\mathcal{F}$ and constraint set $\mathcal{C}$, an $(\mathcal{F}, \mathcal{C})$*-forbidding flower* between (x_1, x_2) is a collection of $(\mathcal{T}_{(\delta,\gamma)}, \mathcal{C})$-forbidding petals between (x_1, x_2), one for each $(\delta, \gamma) \in \mathcal{D}^2$. The flower is said to be *untangled* if (a) each of these petals is untangled, and (b) the petals are mutually vertex-disjoint except at x_1, x_2.

Clearly, an $(\mathcal{F}, \mathcal{C})$-forbidding flower between (x_1, x_2) is unsatisfiable, because all possible assignments to (x_1, x_2) are forbidden by one of the petals.

Consider a distribution $\mathcal{P}$ for which $\mathcal{C} = \mathrm{supp}(\mathcal{P})$ has a complete closure. Following [14, 39], we will show that an $(\mathcal{F}, \mathcal{C})$-forbidding flower a.s. exists in $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ for sufficiently large c, using first and second moment calculations. In the first moment calculation, we need to know the probability that a particular flower exists. To this end, we need to study the probability that a path of length r is (δ, γ)-forbidding (Definition 3.9) and that a petal is is (δ, γ)-forbidding (see Definition 3.12).

Definition 3.9. For a distribution $\mathcal{P}$ and values $\delta, \gamma \in \mathcal{D}$, let $\pi_r^{\mathcal{P}}(\delta, \gamma)$ be the probability over $\mathcal{P}$ that a random constraint path of length r forbids (δ, γ). Define $\beta^{\mathcal{P}}(\delta, \gamma) = \limsup_r \pi_r^{\mathcal{P}}(\delta, \gamma)^{1/r}$.

The definition of $\beta^{\mathcal{P}}(\delta,\gamma)$ takes a lim sup over a sequence, rather than simply a limit, because the sequence may fail to converge. An example is $\operatorname{supp}\mathcal{P} = \{C\}$, where $C = \{(1,1),(2,2)\}$ and $\mathcal{D} = \{1,2\}$. In this case $\pi_r(1,1)$ alternates between 0 and 1 as r increases. More generally, for any arithmetic progression Υ, one can build examples where $\pi_r^{\mathcal{P}}(\delta,\gamma)^{1/r}$ only converges to $\beta^{\mathcal{P}}(\delta,\gamma)$ along $r \in \Upsilon$. Lemma 3.16, below, shows that for every $\mathcal{D}, \delta, \gamma$ there exists an arithmetic progression along which we have this convergence.

Proposition 3.10. *If $\delta \sim_{\operatorname{supp}\mathcal{P}} \gamma$, then $\beta^{\mathcal{P}}(\delta,\gamma) = 0$, otherwise $\beta^{\mathcal{P}}(\delta,\gamma) > 0$.*

The proof is deferred to Section 4.

Definition 3.11. For a distribution $\mathcal{P}$ and a subset $\Delta \subseteq \mathcal{D}^2$, the *distribution of $\mathcal{P}$ augmented with Δ*, denoted $\mathcal{P} \cup \Delta$, is the distribution obtained by "strengthening" every constraint $C \in \mathcal{C} = \operatorname{supp}\mathcal{P}$ with Δ. Formally: for any constraint C' (not necessarily in $\mathcal{C}$), define $(\mathcal{P}\cup\Delta)(C') = \sum \mathcal{P}(C)$, where the sum runs over all $C \in \mathcal{C}$ such that $C \cup \bigcup_{(\delta,\gamma)\in\Delta} F(x_1 : \delta, x_2 : \gamma) = C'$. (Since $\mathcal{P}$ is symmetric by assumption, the sum may as well run over all $C \in \mathcal{C}$ such that $C \cup \bigcup_{(\delta,\gamma)\in\Delta} F(x_i : \delta, x_j : \gamma) = C'$ with arbitrary $i \neq j$.)

Definition 3.12 (Weights)**.** Let $\mathcal{P}$ be a distribution. For a node v in a configuration tree $\mathcal{T}$, define $\pi_r^{\mathcal{P}}(v) = \pi_r^{\mathcal{P}\cup\Delta(v)}(\delta(v),\gamma(v))$ and $\beta^{\mathcal{P}}(v) = \limsup_r \pi_r^{\mathcal{P}}(v)^{1/r}$. For a configuration tree $\mathcal{T}$, define its weight $w^{\mathcal{P}}(\mathcal{T})$ as $\max\{1/(\beta^{\mathcal{P}}(v)k!) \mid v \in V(\mathcal{T})\}$. For a configuration forest $\mathcal{F}$, define its weight $w^{\mathcal{P}}(\mathcal{F})$ as the maximum of the weights of its trees.

Proposition 3.13 (Petals exist)**.** *Let $\mathcal{P}$ be a distribution and set $\mathcal{C} = \operatorname{supp}\mathcal{P}$. For any $\delta_0 \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma_0$, there is a configuration tree $\mathcal{T}$ whose root is labeled (δ_0,γ_0) for which $(\mathcal{T},\mathcal{C})$-forbidding petals exist. Furthermore, $w^{\mathcal{P}}(\mathcal{T}) < \infty$.*

Proof. Let $\mathcal{C}_0 = \mathcal{C}$ and $h = 0$. If $\mathcal{C}_h$ is not closed then we define $\mathcal{C}_{h+1}$ as follows: We initially set $\mathcal{C}_{h+1} = \mathcal{C}_h$. For any $\delta \not\sim_{\mathcal{C}_h} \gamma$, any $C \in \mathcal{C}_h$ and any canonical variables X_i, X_j, add $C \cup F(X_i : \delta, X_j : \gamma)$ to $\mathcal{C}_{h+1}$. Then increase h and repeat.

Intuitively, $\mathcal{C}_h$ represents the set of constraints that can be simulated by a constraint from $\mathcal{C}$ and some level h petals. Since $\delta_0 \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma_0$, then by the definition of $\mathrm{cl}(\mathcal{C})$, $(\delta_0,\gamma_0) \in \mathcal{C}_{h_0}$ for some h_0.

For any $h \geq 0$, define $\Delta_h = \{(\delta,\gamma) \mid \delta \not\sim_{\mathcal{C}_h} \gamma\}$. For any $(\delta,\gamma) \in \Delta_h \setminus \Delta_{h-1}$ (where we set $\Delta_{-1} = \emptyset$), we recursively define: If $h = 0$ then $\mathcal{T}_{(\delta,\gamma)}$ has one vertex, and it is labelled (δ,γ). If $h > 0$ then $\mathcal{T}_{(\delta,\gamma)}$: (i) has a root v labelled (δ,γ) and (ii) for each $(\delta',\gamma') \in \Delta_{h-1}$, v has a child that roots a subtree $\mathcal{T}_{(\delta',\gamma')}$. Note that $\mathcal{F}$ has height at most d^2 and degree at most d^2, and so it has at most d^{2d^2} vertices.

It is straightforward to see that $(\mathcal{T}_{(\delta,\gamma)},\mathcal{C})$-forbidding petals exist. Indeed, if $h = 0$ then a $(\mathcal{T}_{(\delta,\gamma)},\mathcal{C})$-forbidding petal is just a (δ,γ)-forbidding path with constraints from $\mathcal{C}$, which exists since $\delta \not\sim_{\mathcal{C}} \gamma$. If $h > 0$ then consider a (δ,γ)-forbidding path P with constraints from $\mathcal{C}_h$. For each constraint $C \in P$, we (i) replace C with its corresponding constraint $C' \in \mathcal{C}$; (ii) for every $(\delta',\gamma') \in \Delta_{h-1}$, add a $(\mathcal{T}_{(\delta,\gamma)},\mathcal{C})$-forbidding petal joining the two connecting variables of C. This will strengthen C' so that it is equivalent to C. Thus the main path of the resulting petal will be equivalent to P and so will forbid (δ,γ).

If $(\delta,\gamma) \in \Delta_h \setminus \Delta_{h-1}$ then $\operatorname{supp}(\mathcal{P} \cup \Delta(v)) = \mathcal{C}_h$. Since $\delta \not\sim_{\mathcal{C}_h} \gamma$, Proposition 3.10 implies that $\mathcal{T}_{(\delta,\gamma)}$ has finite weight. Therefore $w^{\mathcal{P}}(\mathcal{T}) < \infty$. □

Corollary 3.14. *Let $\mathcal{P}$ be a distribution for which $\mathcal{C} = \operatorname{supp}\mathcal{P}$ has a complete closure. Then there is a configuration forest $\mathcal{F}$ with $w^{\mathcal{P}}(\mathcal{F}) < \infty$ for which $(\mathcal{F}, \mathcal{C})$-forbidding flowers exist.*

We close this section by noting that it is simple to test whether $\mathcal{C}$ satisfies various conditions, and construct certain paths and cycles, since $|\mathcal{C}| = O(1)$.

Lemma 3.15. *Given a distribution $\mathcal{P}$ with $\mathcal{C} = \operatorname{supp}\mathcal{P}$, there is a simple deterministic algorithm to:*

(a) test whether $\operatorname{cl}(\mathcal{C})$ is complete;

(b) test, for each $\delta, \gamma \in \mathcal{D}$, whether $\delta \sim_{\mathcal{C}} \gamma$ and whether $\delta \sim_{\operatorname{cl}(\mathcal{C})} \gamma$;

(c) for each $\delta, \gamma \in \mathcal{D}$ with $\delta \not\sim_{\operatorname{cl}(\mathcal{C})} \gamma$, construct constraint paths P_A, P_B, P_C over $\mathcal{C}$ such that $P_A P_B^i P_C$ is (δ, γ)-forbidding for all $i \geq 0$;

(d) for each $\delta, \gamma \in \mathcal{D}$ with $\delta \not\sim_{\operatorname{cl}(\mathcal{C})} \gamma$, construct a configuration tree $\mathcal{T}_{(\delta,\gamma)}$ corresponding to a $(\mathcal{T}_{(\delta,\gamma)}, \mathcal{C})$-forbidding petal;

(e) for any null-constraining set $\mathcal{D}' \subseteq \mathcal{D}$, test whether there is a cyclic CSP formed from $\mathcal{C}$ that is not $\mathcal{D}'$-satisfiable, and if so, construct one.

Proof. Corollary 2.12 gives a finite-time subroutine to test whether $\delta \sim_{\mathcal{C}} \gamma$ for $\delta, \gamma \in \mathcal{D}$: Simply generate all paths over $\mathcal{C}$ of length between $2^{|\mathcal{D}|}$ and $2 \cdot 2^{|\mathcal{D}|} - 1$ and see if any of them is (δ, γ)-forbidding. Once such a path P is found, one can construct (as in the proof of Corollary 2.12) constraint paths P_A, P_B, P_C from P such that $P_A P_B^i P_C$ is (δ, γ)-forbidding for all $i \geq 0$.

Using the above subroutine that tests $\delta \sim_{\mathcal{C}} \gamma$, one can generate $\operatorname{cl}(\mathcal{C})$ given $\mathcal{C}$, as in the discussion that follows Definition 2.4. Hence one can also test whether $\delta \sim_{\operatorname{cl}(\mathcal{C})} \gamma$ and whether $\operatorname{cl}(\mathcal{C})$ is complete (the latter condition is equivalent to $\operatorname{cl}(\mathcal{C})$ containing the constraint that forbids all d^k k-tuples of partial assignments).

Given $\delta, \gamma \in \mathcal{D}$ with $\delta \not\sim_{\mathcal{C}} \gamma$, we construct a configuration tree $\mathcal{T}_{(\delta,\gamma)}$ as in Proposition 3.13 (using the above subroutine to test $\delta \not\sim_{\mathcal{C}_h} \gamma$ when computing Δ_h).

Given a nonempty null-constraining subdomain $\mathcal{D}' \subseteq \mathcal{D}$, it is easy to construct a cyclic CSP that is not $\mathcal{D}'$-satisfiable (if it exists): Simply enumerate all cycles of length at most $|\mathcal{D}'| \cdot 2^{|\mathcal{D}|} + 3$, and output any one with the desired property. This is because, as we will argue, if some cyclic CSP is $\mathcal{D}'$-UNSAT, then there is a $\mathcal{D}'$-UNSAT cyclic CSP that is shorter than $|\mathcal{D}'| \cdot 2^{|\mathcal{D}|} + 3$. To see this, consider any such cyclic CSP of length $r > |\mathcal{D}'| \cdot 2^{|\mathcal{D}|} + 3$. Call its connecting variables $x_0, x_1, \ldots, x_r$ and constraints $\beta_1, \ldots, \beta_r$, with $x_r = x_0$. For each i and each $\delta \in \mathcal{D}'$, we define a subset $\mathcal{D}_{i,\delta} \subseteq \mathcal{D}'$ of values that x_i can take without violating $\beta_1, \ldots, \beta_i$ and under the restriction that $x_0 = \delta$. Since $r > |\mathcal{D}'| \cdot 2^{|\mathcal{D}|}$, there are $0 \leq i < j \leq r$ such that $\mathcal{D}_{i,\delta} = \mathcal{D}_{j,\delta}$ for all $\delta \in \mathcal{D}'$. Further, we can choose such i, j with $j - i \leq |\mathcal{D}'| \cdot 2^{|\mathcal{D}|}$. Remove $\beta_{i+1}, \ldots, \beta_j$ and identify x_i and x_j to obtain a shorter path. This shorter path is also not $\mathcal{D}'$-satisfiable (because the sets $\mathcal{D}_{r,\delta}$ for $\delta \in \mathcal{D}'$ remain unchanged). The resulting path has length $r - (j - i) \geq 3$, and can be realised as a cycle (there is no cycle shorter than length 3 when $k = 2$). □

3.2 Supercritical Phase

We turn to the proof of a.s. appearance of forbidding flowers in $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$, when c is sufficiently large. Consider any distribution $\mathcal{P}$ for which $\mathcal{C} = \operatorname{supp}\mathcal{P}$ has a complete closure.

We choose a particular configuration forest $\mathcal{F}$ with $w^{\mathcal{P}}(\mathcal{F}) < \infty$, for which $(\mathcal{F}, \mathcal{C})$-forbidding flowers exist, as guaranteed by Corollary 3.14. For each vertex $w \in \mathcal{F}$, we will define a length $r(w) = r_n(w) = \Theta(\log n)$. We will focus on $(\mathcal{F}, \mathcal{C})$-forbidding flowers for which every petal path of type w has length $r(w)$.

We have to be careful when choosing the lengths $r(w)$. Fix $w \in \mathcal{F}$ and consider the sequence $a_r = \pi_r^{\mathcal{P}}(w)^{1/r}$. Recall that $\limsup a_r = \beta^{\mathcal{P}}(w)$. We wish to choose a length $r(w)$ for which $a_{r(w)}$ is close to $\beta^{\mathcal{P}}(w)$. To do so, we find a nice subsequence of the integers along which a_r converges to $\beta^{\mathcal{P}}(w)$, and we choose $r(w)$ from that subsequence. As described following Definition 3.9, we can always find a suitable subsequence that forms an arithmetic progression.

Lemma 3.16 (Convergence along AP)**.** *For any $\delta, \gamma \in \mathcal{D}$, there is an arithmetic progression Υ such that $\pi_r^{\mathcal{P}}(\delta, \gamma)^{1/r}$ converges to $\beta^{\mathcal{P}}(\delta, \gamma)$ along $r \in \Upsilon$.*

Using Lemma 3.16, we fix a constant λ (to be specified later) and define:

$$r_n(w) = \text{ the length of the largest member of } \Upsilon \text{ which does not exceed } \lambda \log n.$$

Thus $r_n(w) = (\lambda + o(1)) \log n$. We often drop the subscript n from the notation, just saying $r(w)$.

Definition 3.17. A $(\mathcal{F}, \mathcal{C})$-forbidding flower H is said to *have petal lengths $r(\cdot)$* if for every vertex $w \in \mathcal{F}$, every petal path of type w has length $r(w)$.

Observation 3.18. *Suppose that H is an untangled $(\mathcal{F}, \mathcal{C})$-forbidding flower with petal lengths $r(\cdot)$. Then*

(a) Every hyperedge in the constraint hypergraph of H contains exactly two connecting variables.

(b) If H' is a $(\mathcal{F}, \mathcal{C})$-forbidding flower with petal lengths $r(\cdot)$ then the constraint hypergraph of H' is isomorphic to the constraint hypergraph of H.

(c) Each variable lies in at most $2|\mathcal{F}|$ constraints of H, where $|\mathcal{F}|$ denotes the number of vertices in $\mathcal{F}$.

Proof. Parts (a,b) follow trivially from the definition of an untangled $(\mathcal{F}, \mathcal{C})$-forbidding flower. For part (c), note that for any vertex $w \in \mathcal{F}$ each variable lies in at most two constraints that are from petal paths of type w. Since every constraint lies in a petal path, this implies that the number of constraints a variable can lie in is at most twice the number of vertices in $\mathcal{F}$. □

Lemma 3.19. *Let H be an untangled $(\mathcal{F}, \mathcal{C})$-forbidding flower with petal lengths $r(\cdot)$. Then H has* polylog(n) *many constraints and connecting variables.*

Proof. Let h denote the number of constraints in H, and u denote the number of connecting variables in H. For each node $w \in \mathcal{F}$, let $q(w)$ denote the number of petal paths in H of type w.

Since each such petal path contains $r(w)$ constraints, and the petal paths are edge-disjoint in H, we have $h = \sum_{w \in \mathcal{F}} q(w) r(w)$. To compute u, we begin with the endpoints of the flower and then add the petal paths one at a time, as in Remark 3.6. Each time we add a petal path of type w, we add $r(w) - 1$ new connecting variables, since the endpoints of the petal path have already been selected, so $u = 2 + \sum_{w \in \mathcal{F}} q(w)(r(w) - 1)$.

We have specified $r = \Theta(\log n)$. Note that $q(w) = 1$ if w is a root and $q(w) = r(w')q(w')$ if w' is the parent of w. Consequently, $q(w) = \prod_a r(a)$, where a runs through all proper ancestors of w, i.e. nodes on the path in $\mathcal{F}$ from the root to w (but excluding w). Since $\mathcal{F}$, and hence its height, are fixed and do not depend on n, this yields $q(w) = \text{polylog}(n)$. Therefore $u, h = \text{polylog}(n)$. □

We shall establish the almost sure existence of forbidding flowers by first and second moment calculations. It will be convenient to work with the $\mathsf{CSP}_{n,p}(\mathcal{P})$ model, with $p = ck!/n^{k-1}$. The results carry over to the $\mathsf{CSP}_{n,M}(\mathcal{P})$ model by Remark 1.6.

Let $\mathcal{A}$ be a particular untangled $(\mathcal{F},\mathcal{C})$-forbidding flower with petal lengths $r(\cdot)$. The variables of $\mathcal{A}$ are *not* variables of $\mathsf{CSP}_{n,p}(\mathcal{P})$; rather we think of $\mathcal{A}$ as a template for some forbidding flowers of $\mathsf{CSP}_{n,p}(\mathcal{P})$.

Definition 3.20. A *potential flower*, A, consists of a hypergraph on the variables of $\mathsf{CSP}_{n,p}(\mathcal{P})$ along with an isomorpism from that hypergraph to $\mathcal{A}$. A potential flower A is *realized* if (i) its hyperedges are all selected for constraints of $\mathsf{CSP}_{n,p}(\mathcal{P})$, and (ii) for every petal path $P \in \mathcal{A}$, the path in A that maps onto P is a petal path of the same type as P.

Thus a realized potential flower A is an untangled $(\mathcal{F},\mathcal{C})$-forbidding flower with petal lengths $r(\cdot)$. The constraints of A may differ from those in $\mathcal{A}$, but the petal paths must be of the same type.

Let X be the random variable counting the number of realized potential flowers in $\mathsf{CSP}_{n,p}(\mathcal{P})$.

Remark 3.21. If the constraint hypergraph of $\mathcal{A}$ has nontrivial automorphisms, then the underlying hypergraph of a potential flower A will have multiple isomorphisms to $\mathcal{A}$ and hence will be the underlying hypergraph of multiple potential flowers. For example, this would be the case if $\mathcal{A}$ were Example 3.1. Thus X can be greater than the number of $(\mathcal{F},\mathcal{C})$-forbidding flowers of $\mathsf{CSP}_{n,p}(\mathcal{P})$. Nevertheless, it suffices to prove that a.s. $X > 0$.

Here $f(n) \sim g(n)$ means $f(n) = (1 + o(1))g(n)$.

Lemma 3.22 (First moment)**.** *If $c > w^{\mathcal{P}}(\mathcal{F})$ then $\mathbb{E}[X] \to \infty$.*

Proof. For each potential flower A on the variables of $\mathsf{CSP}_{n,p}(\mathcal{P})$, we define X_A to be the indicator variable for the event that A is realized.

We start by counting the number of potential flowers. As in the proof of Lemma 3.19, we define u, h and $q(w)$ to be the number of connecting variables, constraints and petal paths of type w in $\mathcal{A}$. Thus, we must choose u connecting variables and $h(k-2)$ other variables. Since $u + h(k-2) = \text{poly}(\log n)$, the number of choices is: $(1 + o(1))n^{u+h(k-2)}$.

Next we compute the probability that a particular potential flower is realized. The probability that all the required hyperedges are chosen for the constraint hypergraph is $\left(\frac{c\times k!}{n^{k-1}}\right)^h$. Recalling Definition 3.9, the probability that the constraints are chosen so as to create petal paths of the correct type is $\prod_{w\in\mathcal{F}} \pi^{\mathcal{P}}_{r(w)}(w)$. Putting this all together, we obtain:

$$\mathbb{E}[X] \sim n^{u+h(k-2)} \left(\frac{c \times k!}{n^{k-1}}\right)^h \prod_{w\in\mathcal{F}} \pi^{\mathcal{P}}_{r(w)}(w).$$

Recall from the proof of Lemma 3.19 that $h = \sum_{w\in\mathcal{F}} q(w)r(w)$ and $u = 2 + \sum_{w\in\mathcal{F}} q(w)(r(w)-1)$. This yields:

$$\mathbb{E}[X] = n^2 \prod_{w\in\mathcal{F}} \left(\frac{(c \times k!)^{r(w)} \pi^{\mathcal{P}}_{r(w)}(w)}{n}\right)^{q(w)}.$$

Recalling Definition 3.12, since $c > w^{\mathcal{P}}(\mathcal{F})$, there is some $\epsilon > 0$ such that $c > (1+\epsilon)/(k!\beta^{\mathcal{P}}(w))$ for all $w \in \mathcal{F}$. Together with $\pi^{\mathcal{P}}_{r(w)}(w)^{1/r} = \beta^{\mathcal{P}}(w) + o(1)$ (by Lemma 3.16 and the fact that $r(w)$ grows with n), we have

$$\mathbb{E}[X] \sim n^2 \prod_{w\in\mathcal{F}} \left(\frac{(c \times k!(\beta^{\mathcal{P}}(w) + o(1)))^{r(w)}}{n} \right)^{q(w)} \tag{1}$$

$$\geq n^2 \prod_{w\in\mathcal{F}} \left(\frac{(1+\epsilon+o(1))^{(\lambda+o(1))\log n}}{n} \right)^{q(w)} \tag{2}$$

$$= n^2 \prod_{w\in\mathcal{F}} \left(n^{(\lambda+o(1))\log(1+\epsilon+o(1))-1} \right)^{q(w)}. \tag{3}$$

Consider the exponent of n in (3). The term $\log(1+\epsilon-o(1))$ is positive for large n, so since $\lambda > 0$, the exponent of n will be positive. Therefore (3) tends to infinity.

□

Lemma 3.23 (Second moment)**.** *If $w^{\mathcal{P}}(\mathcal{F}) < \infty$, then $\mathbb{E}(X^2) = (1+o(1))E(X)^2$ provided c and λ are sufficiently large.*

Proof. Let A be a potential flower. It is well known (see e.g. Corollary 4.3.5, [6]) that we only need to show

$$\sum_{B:E(B)\cap E(A)\neq\emptyset} \mathbb{P}[X_B = 1 | X_A = 1] = o(\mathbb{E}[X]). \tag{4}$$

Writing $\mathcal{E}_B$ as the event that all k-tuples of $E(B)$ are selected as constraints for $\mathsf{CSP}_{n,p}(\mathcal{P})$, we get

$$\mathbb{P}[X_B = 1 \mid X_A = 1] = \mathbb{P}[\mathcal{E}_B \mid X_A = 1]\mathbb{P}[X_B \mid \mathcal{E}_B, X_A = 1]$$

The first term $\mathbb{P}[\mathcal{E}_B \mid X_A = 1]$ equals $\mathbb{P}[\mathcal{E}_B]/p^{|E(B)\cap E(A)|}$, where $p = ck!/n^{k-1}$. We bound the second term with the following claim, whose proof (as well as the proofs of other claims that follow) is deferred until after the second moment calculation.

Claim 3.24. *Letting $\alpha = \min_{C\in\operatorname{supp}\mathcal{P}} \mathcal{P}(C)$,*

$$\mathbb{P}[X_B = 1 \mid \mathcal{E}_B, X_A = 1] \leq \mathbb{P}[X_B = 1 \mid \mathcal{E}_B]/\alpha^{|E(B)\cap E(A)|}. \tag{5}$$

Therefore

$$\sum_{B:E(B)\cap E(A)\neq\emptyset} \mathbb{P}[X_B = 1 \mid X_A = 1] \leq \sum_{B:E(B)\cap E(A)\neq\emptyset} \mathbb{P}[\mathcal{E}_B \mid X_A = 1]\mathbb{P}[X_B = 1 \mid \mathcal{E}_B, X_A = 1] \tag{6}$$

$$\leq \sum_{B:E(B)\cap E(A)\neq\emptyset} \frac{\mathbb{P}[\mathcal{E}_B]\mathbb{P}[X_B = 1 \mid \mathcal{E}_B]}{(\alpha p)^{|E(B)\cap E(A)|}} \tag{7}$$

$$= \mathbb{P}[X_{B_0} = 1] \sum_{B:E(B)\cap E(A)\neq\emptyset} \frac{1}{(\alpha p)^{|E(B)\cap E(A)|}}, \tag{8}$$

where B_0 is any fixed potential flower. We will show below that

$$\sum_{B:E(B)\cap E(A)\neq\emptyset} \frac{1}{(\alpha p)^{|E(B)\cap E(A)|}} \leq o(n^{u+h(k-2)}). \tag{9}$$

Recall from the proof of Lemma 3.22 that the number of potential flowers is $(1+o(1))n^{u+h(k-2)}$. So plugging (9) into (8) gives the desired bound in (4), thus completing our proof. So let us show (9).

Let $\mathbf{h}$ be any vector $(h_1,\ldots,h_J)$ with every $h_i \geq 1$. (Note that $\mathbf{h}$ implicitly specifies J.) For any potential flower B, we say $E(B)\cap E(A) \models \mathbf{h}$ if the hyperedges of $E(B)\cap E(A)$ form J connected components $H_1,...,H_J$ such that each H_j has h_j hyperedges. We will upper bound the number of potential flowers B with $E(B)\cap E(A)\models \mathbf{h}$.

First we count the number of choices of the subgraphs $H_1,...,H_J \subseteq A$. To choose a connected subhypergraph of A with exactly h_j hyperedges, we consider the line graph $L(A)$; i.e. the graph whose vertices are the hyperedges of A and where two vertices are adjacent iff the corresponding hyperedges intersect. The number of choices for H_j is at most the number of connected subgraphs of $L(A)$ with h_j vertices. Pick a vertex $u \in L(A)$. The number of connected subgraphs of size h_j containing u is at most the number of subtrees of $L(A)$ of size h_j rooted at u. It follows from Exercise 11 on page 396 of [29] that the number of such subtrees is at most $(eD)^{h_j}$ where D is the maximum degree of $L(A)$. By Observation 3.18(c), we have $D \leq 2k|\mathcal{F}|$. By Lemma 3.19, there are polylog(n) choices for u. Therefore, the number of choices for H_j is at most polylog$(n)K^{h_j}$ where

$$K = 2ek|\mathcal{F}|.$$

Given H_j, the number of choices of which vertices of $\mathcal{A}$ correspond to H_j in our choice of B is also at most polylog$(n)K^{h_j}$.

Given the choice of $H_1,...,H_J$, we let u_j be the number of variables in H_j that are connecting in A. Thus the total number of variables in H_j is $u_j+(k-2)h_j$. To choose the rest of B, we simply specify which variables map onto the remaining $u-\sum u_j+(k-2)(h-\sum_j h_j)$ variables of $\mathcal{A}$; there are at most $n^{u-\sum u_j+(k-2)(h-\sum_j h_j)}$ such choices.

If H_j has no cycles, then $u_j = h_j+1$. Since the petal paths have length $\Theta(\log n)$, so must any cycles. This allows us to prove:

Claim 3.25.

$$u_j \geq h_j + 1 - \frac{2h_j}{\lambda \log n}.$$

Take λ large enough so that $n^{2/(\lambda\log n)} \leq 2$. Thus we have

$$n^{u-\sum u_j+(k-2)(h-\sum_j h_j)} \leq n^{u+(k-2)h-\sum_j((k-1)h_j+1)}2^{\sum_j h_j}$$

and so the total number of potential flowers B with $E(B)\cap E(A)\models \mathbf{h}$ is at most

$$n^{u+(k-2)h-\sum_j((k-1)h_j+1)}2^{\sum_j h_j}\left(\prod_j \text{poly}(\log n)K^{h_j}\right)^2 k^J(k-1)!^{\sum_j h_j}$$
$$< n^{u+(k-2)h}\prod_j \text{poly}(\log n)k\frac{(K^2\times(k-1)!)^{h_j}}{n^{(k-1)h_j+1}}.$$

Therefore, for a specific $\mathbf{h}$ we have (after absorbing $k = O(1)$ into the $\mathrm{poly}(\log n)$ term):

$$\begin{aligned}
&\sum_{E(B)\cap E(A)\models \mathbf{h}} \frac{1}{(\alpha p)^{|E(B)\cap E(A)|}} \\
&\le n^{u+(k-2)h} \prod_j \frac{\mathrm{poly}(\log n)}{n} \left(\frac{K^2 \times (k-1)!}{n^{k-1}}\right)^{h_j} \left(\alpha \cdot \frac{c \times k!}{n^{k-1}}\right)^{-h_j} \\
&= n^{u+(k-2)h} \prod_{j=1}^{J} \frac{\mathrm{poly}(\log n)}{n} \left(\frac{K^2}{ck\alpha}\right)^{h_j} \\
&< n^{u+(k-2)h} \prod_{j=1}^{J} \frac{\mathrm{poly}(\log n)}{n} 2^{-h_j},
\end{aligned}$$

for $c > 2K^2/k\alpha$. Summing over all $\mathbf{h}$ of length J; i.e. over all subhypergraphs with J components,

$$\begin{aligned}
\sum_{B:E(B)\cap E(A) \text{ has } J \text{ components}} \frac{1}{(\alpha p)^{|E(B)\cap E(A)|}} &\le n^{u+(k-2)h} \prod_{j=1}^{J} \left(\frac{\mathrm{polylog}(n)}{n} \sum_{h_j \ge 1} 2^{-h_j}\right) \\
&\le n^{u+(k-2)h} \left(\frac{\mathrm{polylog}(n)}{n}\right)^{J}.
\end{aligned}$$

Finally summing over all $J \ge 1$ (since $E(B) \cap E(A) \neq \emptyset$ implies $J \ge 1$), we get (9). This completes the proof except for Claims 3.24 and 3.25, whose proofs we turn to next. □

Proof of Claim 3.24. Let C_A be the collection of all possible assignments of constraints to $E(A)$ that would make $X_A = 1$. Abusing notation, we also denote by C_A the event that the k-tuples of $E(A)$ are given an assignment from C_A. For any set of hyperedges $A' \subset E(A)$, we denote by $C_{A'}$ the event that the hyperedges of A' are assigned the restriction of an assignment of C_A to A'. We have

$$\mathbb{P}[X_B = 1 \mid \mathcal{E}_B] \ge \mathbb{P}[X_B = 1 \mid C_{E(B)\cap E(A)}, \mathcal{E}_B] \Pr[C_{E(B)\cap E(A)}].$$

Now the first term on the right equals $\mathbb{P}[X_B = 1 \mid C_A, \mathcal{E}_B] = \mathbb{P}[X_B = 1 \mid X_A = 1, \mathcal{E}_B]$. For the second term, take an arbitrary $\mathbf{C} \in C_{E(B)\cap E(A)}$, and we have $\Pr[C_{E(B)\cap E(A)}] \ge \Pr[\mathbf{C}] \ge (\min_{C \in \mathrm{supp}\, \mathcal{P}} \mathcal{P}(C))^{|E(B)\cap E(A)|}$. □

Proof of Claim 3.25. It will be convenient to view the constraint hypergraph of an untangled forbidding flower as a planar graph. To do so, replace each hyperedge by an edge between its connecting variables. Considering Remark 3.6, a straightforward recursive argument shows that this yields a planar graph and that we can associate each internal face with a petal path, specifically a petal path whose addition (in the construction described in Remark 3.6) led to the creation of that face.

This replacement transforms H_j into a connected subgraph H'_j of that planar graph, where H'_j has u_j vertices and h_j edges. By Euler's formula for planar graphs, the number of internal faces of H'_j is exactly $h_j - u_j + 1$. H_j contains the petal path associated with each of those faces, and each path contains $(\lambda + o(1)) \log n > \frac{1}{2}\lambda \log n$ hyperedges. Thus $h_j \ge (h_j - u_j + 1) \times \frac{1}{2}\lambda \log n$, and the claim follows. □

And now we can finally prove:

Proof of Theorem 1.1 Because $\mathcal{C} = \operatorname{supp} \mathcal{P}$ has a complete closure, Proposition 3.13 implies that there is a forest $\mathcal{F}$ of finite weight such that there are $(\mathcal{F}, \mathcal{C})$-forbidding petals. Pick any such a forest $\mathcal{F}$. Lemmas 3.22, 3.23, Chebychev's Inequality (see eg. [26]) and Remark 1.6 assert that a.s. a $(\mathcal{F}, \mathcal{C})$-flower exists in $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ for large c. Let $\mathcal{I}$ be the sup-CSP formed by the $(\mathcal{F}, \mathcal{C})$-flower. Recall the boolean CNF formula $\mathrm{CNF}(\mathcal{I})$ defined in Definition 1.8. We will prove that $\mathbf{RES}(\mathrm{CNF}(\mathcal{I}))$ =polylog(n), hence establishing Theorem 1.1.

We first note that all forbidden pairs of values of each constraint of $\mathcal{I}$ can be resolved in $O(1)$ steps. Indeed, if a constraint in $\mathcal{I}$ forbids $(x_i : \delta, x_{i+1} : \gamma)$, the clause $(\overline{x_i : \delta} \vee \overline{x_{i+1} : \gamma})$ is implied by $O(1)$ clauses of $\mathrm{CNF}(\mathcal{I})$ and therefore can be derived from $\mathrm{CNF}(\mathcal{I})$ with $O(1)$ resolutions. We resolve petal paths in the flower sequentially, starting with paths corresponding to the leaves of $\mathcal{F}$. For each constraint e in a path, we have already resolved the paths corresponding to its children in $\mathcal{F}$, and hence have already obtained the restrictions implied by the subpetals whose endpoints are the connecting vertices of e. Hence, we can obtain the forbidden pair of values on those endpoints in an additional $O(1)$ steps. After obtaining the forbidden pair of values for every constraint in the path, we can obtain the forbidden pair on the endpoints of the path with a number of resolution steps that is proportional to the length of the path. So overall, we can resolve a petal path of length r in $O(r)$ steps. This yields a resolution refutation of $\mathrm{CNF}(\mathcal{I})$ of length at most a constant times the number of constraints in $\mathcal{I}$, which is polylog(n) by Proposition 3.19. □

Consequently, given a path in $\mathcal{I}$ of length r between (x_1, x_2), all clauses corresponding to its forbidding value pairs (i.e. the clauses $(\overline{x_1 : \delta} \vee \overline{x_2 : \gamma})$) can be derived from $\mathrm{CNF}(\mathcal{I})$ with $O(r)$ resolutions.

Remark 3.26. Theorem 1.1 still holds if we measure resolution complexity by **NG-RES**. Indeed, if a constraint forbids $(x_i : \delta, x_j : \gamma)$, the nogood $(x_i \neq \delta, x_j \neq \gamma)$ can be derived with $O(1)$ nogood resolutions (for the definition of nogood and nogood resolution, see [35]). Using the same argument as above, a flower has polylog **NG-RES**.

4 Random Walks on Directed Graphs

This section is mainly devoted to proving Lemma 3.16, which says that $\pi_r^{\mathcal{P}}(\delta, \gamma)^{1/r}$ converges to $\beta^{\mathcal{P}}(\delta, \gamma)$ along some arithmetic progression. The behavior of $\pi_r^{\mathcal{P}}(\delta, \gamma)^{1/r}$ is best studied as a random walk on a related digraph, thus we are led to the analysis of convergence of such a random walk (Theorem 4.2).

Let $G = (V, E)$ be a fixed digraph with positive edge weights. so that at any vertex, the sum of the outgoing edge weights is 1. A random walk on G from u is one which starts at u, and at any stage of the walk at a vertex v, we go to a neighbor w of v with probability the weight of the arc vw.

Definition 4.1. For any $u \in V(G)$, $V' \subseteq V(G)$, we define $\pi_r(u, V')$ to be the probability that a random walk from u of length r lands on a vertex in V'. We define $R(u, V') = \limsup_r \pi_r(u, V')^{1/r}$. If $V' = \{v\}$ then we use the notations $\pi_r(u, v), R(u, v)$.

Theorem 4.2. *$\pi_r(u, V')^{1/r}$ converges to $R(u, V')$ along some arithmetic progression.*

The theorem may be of independent interest. Quite possibly it has appeared elsewhere, but we could not find it. It will be proved using the following sequence of propositions.

Proposition 4.3. *Let T be a finite set. Assume every $t \in T$ is associated with a sequence $\{a_r(t)\}$, such that $a_r(t) \geq 0$ and $\limsup_r (a_r(t))^{1/r} = a(t)$. Let $b_r = \sum_{t \in T} a_r(t)$ and $b = \limsup_r b_r^{1/r}$. Then $b = \max_{t \in T} a(t)$.*

Proof. Pick $s \in T$ so that $a(s)$ is maximized. Clearly $b_r \geq a_r(s)$. Taking r-th root and then $\limsup_r$ on both sides, we get $b \geq a(s)$. On the other hand, $a_r(t) \leq (a(t)+o(1))^r$ for any t, yielding $b_r \leq |T|(a(s) + o(1))^r$. Taking r-th root and then $\limsup_r$ on both sides, we get $b \leq a(s)$. □

Proposition 4.4. $R(u, V') = \max\{R(u, v) \mid v \in V'\}$.

Proof. Apply Proposition 4.3 with $T = V'$ and $a_r(v) = \pi_r(u, v)$. Observe that b_r becomes $\pi_r(u, V')$. □

It turns out $R(u, v)$ depends only on the strongly connected components of u and v.

Proposition 4.5. *Assume u, u' belong to the same strongly connected component, and so do v, v'. Then $R(u, v) = R(u', v')$.*

Proof. Let $\mathbf{p}$ be a path from u to u', and $\mathbf{q}$ a path from v' to v. Let $a > 0$ be the probability of traversing $\mathbf{p}$, and $b > 0$ that of traversing $\mathbf{q}$. Let ℓ be the sum of lengths of $\mathbf{p}$ and $\mathbf{q}$. One way to go from u to v in $r + \ell$ steps is to go along $\mathbf{p}$, then go from u' to v' in r steps and finally go along $\mathbf{q}$. Hence $\pi_{r+\ell}(u, v) \geq ab\pi_r(u', v')$. Taking $(r + \ell)$-th root and then $\limsup_r$ on both sides, we get $R(u, v) \geq R(u', v')$. Reversing the roles of (u, v) and (u', v'), we get $R(u', v') \geq R(u, v)$. □

For a vertex u in G, we denote by $[u]$ the strongly connected component of u. If we let $R([u], [v]) = R(u, v)$, the previous proposition shows that $R([u], [v])$ is well-defined. For convenience, we let[6] $R([u]) = R([u], [u])$. For the given digraph G, let us denote by G^{C} the component digraph of G. It is obtained from G by contracting every strongly connected component. For a strongly connected component $[u]$ in G, we denote by u^{C} its corresponding vertex in G^{C}. For a walk $\mathbf{w}$ in G, its induced (simple) path $\mathbf{w}^{\mathsf{C}}$ is the path in G^{C} obtained by contracting every strongly connected component of G along $\mathbf{w}$.

Definition 4.6. Let p be a (simple) path starting from u^{C} in G^{C}. $\pi_r(\mathsf{p})$ is defined to be the probability that a random walk in G from u of length r has p as its induced path. We define $R(\mathsf{p}) = \limsup_r \pi_r(\mathsf{p})^{1/r}$.

Proposition 4.7. *For any simple path p starting from u^{C} in G^{C},*

$$R([u], [v]) = \max\{R(\mathsf{p}) \mid \mathsf{p} \text{ is a } u^{\mathsf{C}}, v^{\mathsf{C}}\text{-path in } G^{\mathsf{C}}\}.$$

Proof. Apply Proposition 4.3 with $a_r(\mathsf{p}) = \pi_r(\mathsf{p})$ for any $u^{\mathsf{C}}, v^{\mathsf{C}}$-path p. Observe that $b_r = \pi_r([u], [v])$. □

Lemma 4.8. *Let p be a path from u^{C} to v^{C} in G^{C}. Then $R(\mathsf{p}) = \max\{R([w]) \mid w^{\mathsf{C}} \in V(\mathsf{p})\}$.*

[6] Readers familiar with quasi-stationary distributions of absorbing Markov processes (see e.g. [21, 30]) may have realized that $R([u])$ is the spectral radius of the probability transition matrix on $[u]$.

Proof. Suppose p is a $u^{\mathsf{C}}, v^{\mathsf{C}}$-path in G^{C}. Assume w_0 in G maximizes $R([w])$ among $w^{\mathsf{C}} \in V(\mathsf{p})$. Take a u, w_0-path $\mathbf{p}$ and a w_0, v-path $\mathbf{q}$ in G, and let ℓ be their sum of lengths. A possible walk of length $r+\ell$ with its induced path being p is like this: It begins with $\mathbf{p}$, then goes from w_0 to w_0 in r steps, finally ends with $\mathbf{q}$. Let $a > 0$ be the probability of traversing $\mathbf{p}$ and $b > 0$ be that of $\mathbf{q}$. Then $\pi_{r+\ell}(\mathsf{p}) \geq ab\pi_r(w_0, w_0)$. Taking $(r+\ell)$-th root and then $\limsup_r$ on both sides, we get $R(\mathsf{p}) \geq R(w_0, w_0) = R([w_0])$.

For the reverse inequality, assume $v(1)^{\mathsf{C}}, \ldots, v(t)^{\mathsf{C}}$ are the vertices of p. Let $\mathbf{w}$ be a walk such that $\mathbf{w}^{\mathsf{C}} = \mathsf{p}$. Renaming if necessary, we may assume $\mathbf{w}$ enters $[v(i)]$ at the vertex $v(i)$. For $1 \leq i \leq t$, let $s(i)$ be the number of steps that $\mathbf{w}$ makes within $[v(i)]$. Now for any $1 \leq i \leq t$, the probability of $\mathbf{w}$ staying in $[v(i)]$ for $s(i)$ steps is $\pi_{s(i)}(v(i), [v(i)]) \leq (R([v(i)]) + f(s(i)))^{s(i)}$, where $f(s(i)) = o(1)$. Note that $t - 1 + \sum_i s(i) = r$, and let $\mathcal{S}_r = \{(s(1), \ldots, s(t)) \mid t - 1 + \sum_i s(i) = r\}$ be the set of all such t-tuples. We have

$$\pi_r(\mathsf{p}) \leq \sum_{\mathbf{s} \in \mathcal{S}_r} \prod_{1 \leq i \leq t} (R([v(i)]) + f(s(i)))^{s(i)}. \tag{10}$$

Let $R_0 = R([w_0])$ (hence $R_0 \geq R([v(i)])$ for any i). There are $|\mathcal{S}_r| \leq r^t$ terms in the sum in (10) (a loose upper bound suffices).

Claim 4.9. $\pi_r(\mathsf{p}) \leq r^t (R_0 + o(1))^{r-t+1}$.

Proof. Define $g(r)$ by

$$(R_0 + g(r))^{r-t+1} = \max_{\mathbf{s} \in \mathcal{S}_r} \prod_{1 \leq i \leq t} (R_0 + f(s(i)))^{s(i)} \tag{11}$$

so that

$$\prod_{1 \leq i \leq t} (R_0 + f(s(i)))^{s(i)} \leq (R_0 + g(r))^{r-t+1}. \tag{12}$$

Hence an upper bound to the right hand side of (10) is $r^t (R_0 + g(r))^{r-t+1}$.

We claim that $g(r) = o(1)$. Indeed, take any sequence $\{\mathbf{s}_r\}$ of t-tuples, with $\mathbf{s}_r = (s_r(1), \ldots, s_r(t)) \in \mathcal{S}_r$. It suffices to show

$$\prod_{1 \leq i \leq t} (R_0 + f(s(i)))^{s(i)} \leq (R_0 + o(1))^{r-t+1}, \tag{13}$$

where $s(i) = s_r(i)$. Taking $(r-t+1)$-th root, we need to show

$$\prod_{1 \leq i \leq t} (R_0 + f(s(i)))^{s(i)/(r-t+1)} \leq R_0 + o(1). \tag{14}$$

Fix $\epsilon > 0$. There is an s_0 such that $f(s) < \epsilon$ for all $s \geq s_0$, and a B such that $f(s) \leq B$ for all $s < s_0$. Then choose r_0 such that $(R_0 + B)^{s/(r-t+1)} \leq (1+\epsilon)(R_0 + \epsilon)^{s/(r-t+1)}$ for all $s < s_0$ and $r \geq r_0$. For each i, consider its contribution to the product in (14). If $s(i) \geq s_0$, then $(R_0 + f(s(i)))^{s(i)/(r-t+1)} < (R_0 + \epsilon)^{s(i)/(r-t+1)}$. If $s(i) < s_0$, then $(R_0 + f(s(i)))^{s(i)/(r-t+1)} \leq (R_0 + B)^{s(i)/(r-t+1)} \leq (1+\epsilon)(R_0 + \epsilon)^{s(i)/(r-t+1)}$ whenever $r \geq r_0$. Hence the left hand side of (14) is at most $(1+\epsilon)^t (R_0 + \epsilon)$ for $r \geq r_0$. Since t is a constant and ϵ is arbitrary, their total contribution is $R_0 + o(1)$, and the claim follows. □

From the claim, we take r-th root and then $\limsup_r$ on both sides, getting $R(\mathsf{p}) \leq R_0 = R([w_0])$. □

Lemma 4.10. *For any vertex v in G, $\pi_r(v,v)^{1/r}$ converges to $\limsup_r \pi_r(v,v)^{1/r}$ along multiples of an integer.*

Proof. Let $a_r = \pi_r(v,v)^{1/r}$. If $a_r = 0$ for all $r > 0$, the conclusion is trivial, so assume some $a_r > 0$. Let $S = \{r \mid a_r > 0\}$. For any $m, n \in \mathbb{N}$, $\pi_{m+n}(v,v) \geq \pi_m(v,v)\pi_n(v,v)$, hence

$$a_{r+s} \geq a_r^{r/(r+s)} a_s^{s/(r+s)}. \tag{15}$$

This implies S is closed under addition. It follows that S contains all sufficiently large multiples of d, where $d = \gcd(S)$. Equation (15) also implies a_r are supermultiplicative, and by Fekete's Lemma (e.g. [41, Lemma 11.6]), a_r converges to $\limsup_r a_r$ along multiples of d. □

Proof of Theorem 4.2. The theorem is trivial if V' is not reachable from u, so let us assume this is not the case. Propositions 4.4, 4.7 and Lemma 4.8 together imply $R(u,V') = R(w,w)$ for some w lying on some path from u to V'. Lemma 4.10 asserts that $\pi_r(w,w)^{1/r}$ converges to $R(w,w)$ along some arithmetic progression $S \subseteq \mathbb{N}$. We consider paths $\mathbf{p}$ from u to w and $\mathbf{q}$ from w to V', and use the same analysis as in the first part of Proposition 4.8 to show that $\pi_r(u,V')^{1/r}$ converges to $R(u,V')$ along $S + \ell$. □

We are now ready to prove Lemma 3.16.

Proof of Lemma 3.16. Consider a constraint path $P = \langle x_0, \ldots, x_r \rangle$ with constraints chosen randomly according to $\mathcal{P}$. Suppose x_0 is allowed to take values from $\mathcal{D}_0$. Let $\mathcal{D}_i$ be the set of values that x_i can take without violating constraints in P, for $1 \leq i \leq r$. Then $\langle D_0, \ldots, D_r \rangle$ corresponds naturally to a random walk on a digraph G defined as follows. The vertex set $V(G)$ is the power set of $\mathcal{D}$. For any subdomain $\mathcal{D}' \subseteq \mathcal{D}$ and any $C \in \operatorname{supp} \mathcal{P}$, consider two canonical variables X_1, X_2 in a constraint C. Define $\theta(C, \mathcal{D}')$ to be the set of values δ_2 such that there is $\delta_1 \in \mathcal{D}'$ such that C permits $(X_1 : \delta_1, X_2 : \delta_2)$. Then we put an arc $(\mathcal{D}', \theta(C, \mathcal{D}'))$ of weight $\mathcal{P}(C)$ in G.

Now let $\mathcal{D}_0 = \{\delta\}$ and $V' = \{\mathcal{D}' \subseteq \mathcal{D} \mid \gamma \notin \mathcal{D}'\}$. It is easy to see that $\pi_r^{\mathcal{P}}(\mathcal{D}_0, V')$ is the probability over $\mathcal{P}$ that a constraint path of length r on $\operatorname{supp} \mathcal{P}$ is (δ, γ)-forbidding. The result follows by Theorem 4.2. □

We close this section by proving Proposition 3.10.

Proof of Proposition 3.10. If $\delta \sim_{\mathcal{C}} \gamma$, $\pi_r^{\mathcal{P}}(\delta, \gamma) = 0$ for all sufficiently large r, hence $\beta^{\mathcal{P}}(\delta, \gamma) = 0$. If $\delta \not\sim_{\mathcal{C}} \gamma$, let P be a (δ, γ)-forbidding path over $\mathcal{C}$ of length at least $2^{|\mathcal{D}|}$. Consider the digraph G defined in the proof of Lemma 3.16. P corresponds to a walk $\mathbf{w}$ from $\{\delta\}$ in the digraph. Since G has only $2^{|\mathcal{D}|}$ vertices, $\mathbf{w}$ visits some vertex (at least) twice, say w. Then the portion of $\mathbf{w}$ between the two visits is a circuit from w to itself, say of length $s \geq 1$. Hence $\pi_s(w,w) > 0$. The numbers $a_r = \pi_r(w,w)^{1/r}$ are supermultiplicative by (15), hence $R(w,w) \geq \pi_s(w,w)^{1/s} > 0$. If $u = \{\delta\}$ and $V' = \{\mathcal{D}' \subseteq \mathcal{D} \mid \gamma \notin \mathcal{D}'\}$, then $R(u,V') \geq R(w,w)$ by Propositions 4.4, 4.7 and Lemma 4.8. Hence $\beta^{\mathcal{P}}(\delta, \gamma) = R(u,V') > 0$. □

5 Incomplete Closures

In this section, we turn to constraint sets whose closures are incomplete (Theorems 1.2 and 1.4).

Consider any distribution $\mathcal{P}$ where $\mathcal{C} = \operatorname{supp} \mathcal{P}$ has an incomplete closure. By Corollary 2.9, some nonempty subdomain $\mathcal{D}' \subseteq \mathcal{D}$ is null-constraining. In other words, there is some integer t

such that all constraint paths over $\mathcal{C}$ with length at least t $\mathcal{D}'$-permit all $(\delta, \gamma) \in \mathcal{D}' \times \mathcal{D}'$. This implies, in particular, that all cycles of length at least t are $\mathcal{D}'$-satisfiable.

In subsection 5.1, We prove that a.s. there is no subexponential resolution proof that the random CSP is not $\mathcal{D}'$-satisfiable unless it contains a cycle that is not $\mathcal{D}'$-satisfiable. The constraint hypergraph will w.u.p.p. have girth at least t and hence have no such cycle. This implies Theorems 1.2 and 1.4(a).

That part of the proof follows a standard approach: There is some $\alpha > 0$ such that a.s. every non-cyclic sub-CSP of size at most αn has either (i) a constraint where all but one variable has degree one or (ii) a path of length at least t in which all internal variables do not lie in any other constraint. If that sub-CSP is not $\mathcal{D}'$-satisfiable then neither is the sub-CSP formed by deleting the constraint of (i) or the constraints of the path of (ii). It follows recursively that the sub-CSP is indeed $\mathcal{D}'$-satisfiable unless it contains a $\mathcal{D}'$-unsatisfiable cycle. Furthermore, if it has size at least $\frac{1}{2}\alpha n$ then there must be $\Theta(n)$ such constraints and/or paths and they can serve as the *boundary*; this allows us to apply the Width Lemma of [12].

We prove Theorem 1.4(b) in subsection 5.2.

5.1 Exponential Complexity

Since $\mathcal{C} = \operatorname{supp} \mathcal{P}$ has an incomplete closure, Definition 2.6 and Corollary 2.9 imply that we can choose some $\mathcal{D}' \subseteq \mathcal{D}$ and $t > 0$ such that:

Fact 5.1. *Every constraint path over $\mathcal{C}$ with length at least t $\mathcal{D}'$-permits all $(\delta, \gamma) \in \mathcal{D}' \times \mathcal{D}'$. Note this implies that every cyclic CSP of length at least t whose constraints are from $\mathcal{C}$ is $\mathcal{D}'$-satisfiable.*

We denote by $\Omega(\operatorname{supp} \mathcal{P})$ the set of all CSP instances whose constraints are drawn only from $\operatorname{supp} \mathcal{P}$. As in many other results concerning exponential resolution complexity, we define certain sub-CSP's to be *boundaries*.

Definition 5.2. Let $\mathcal{I} \in \Omega(\operatorname{supp} \mathcal{P})$.

1. The *first boundary* of $\mathcal{I}$, denoted $\mathcal{B}^1(\mathcal{I})$, is the set of constraints of $\mathcal{I}$ which contain at most one variable of degree greater than 1.
2. The *second boundary* of $\mathcal{I}$, denoted $\mathcal{B}^2(\mathcal{I})$, is the set of pendant paths of length t in $\mathcal{I}$.
3. The *boundary* of $\mathcal{I}$ is $\mathcal{B}(\mathcal{I}) = \mathcal{B}^1(\mathcal{I}) \cup \mathcal{B}^2(\mathcal{I})$.

Recall that a pendant path in a CSP is defined (in Section 2) to be the set of constraints corresponding to a pendant path in the underlying hypergraph. Thus, every element of the boundary on $\mathcal{I}$ is a set of constraints of $\mathcal{I}$.

Lemma 5.3. *Suppose $\mathcal{I} \in \Omega(\operatorname{supp} \mathcal{P})$ and $X \in \mathcal{B}(\mathcal{I})$. Any satisfying $\mathcal{D}'$-assignment α of $\mathcal{I} - X$ can be extended to a satisfying $\mathcal{D}'$-assignment of $\mathcal{I}$.*

Proof. Suppose $X \in \mathcal{B}^1(\mathcal{I})$, and let x_0 be the variable in X with maximum degree (all other variables have degree 1). No variable other than x_0 is assigned a value by α. It is easy to see that for any value δ, there is a satisfying $\mathcal{D}'$-assignment for X with $x_0 = \delta$; such an assignment extends α to a $\mathcal{D}'$-assignment of $\mathcal{I}$. To see this, note that otherwise one can construct an arbitrarily long path, whose constraints are all isomorphic to X, that does not permit (δ, γ) for any γ (indeed the

first constraint cannot be satisfied if the first endpoint recieves δ); this contradicts the fact that $\mathcal{D}'$ is null-constraining.

Suppose $X \in \mathcal{B}^2(\mathcal{I})$. The lemma follows from Fact 5.1. □

We next prove a lemma (Lemma 5.8) which will imply exponential resolution complexity. It says that if every small subproblem is satisfiable and every subproblem with non-negligible size has many boundaries, then the CSP has large resolution complexity. This lemma is of a standard type for proving exponential resolution complexity, and its proof is essentially identical to that of Lemma 7 in [39] (which in turn follows Mitchell's framework [36]).

There is a slight twist, however. Instead of simply requiring all small subproblems to be satisfiable, we require them to be $\mathcal{D}'$-satisfiable. The idea is that any resolution refutation for the unsatisfiability of $\mathcal{I}$ also proves that $\mathcal{I}$ is $\mathcal{D}'$-unsatisfiable. Therefore exponential resolution complexity of $\mathcal{I}$ follows from exponential resolution complexity of $\mathcal{I} \restriction_{\mathcal{D}'}$, the CSP derived from $\mathcal{I}$ with its domain restricted to $\mathcal{D}'$ (i.e. variables can take values only from $\mathcal{D}'$).

Recall our definition of CNF($\mathcal{I}$) from Section 1.3.

Definition 5.4. For a CNF-clause C and a variable x, the *restriction of C on x = false*, denoted $C \restriction_{x=\text{false}}$, is defined as follows: If the *literal* $\overline{x}$ appears in C, it is $\{\text{true}\}$, the clause which is always true. Otherwise, it is $C \setminus x$, the clause obtained from C by removing the *variable* x. For a formula ϕ and a variable x, the *restriction of ϕ on x = false* is $\{C \restriction_{x=\text{false}} | \ C \in \phi\}$.

Definition 5.5. CNF($\mathcal{I}$) $\restriction_{\mathcal{D}'}$ is the restriction of CNF($\mathcal{I}$) on $x_i : \delta = \text{false}$ for all x_i and all $\delta \notin \mathcal{D}'$. It is easy to see that CNF($\mathcal{I} \restriction_{\mathcal{D}'}$) = CNF($\mathcal{I}$) $\restriction_{\mathcal{D}'}$.

A useful property of restriction is that it distributes over disjunction: $(C_1 \vee C_2) \restriction_{\mathcal{D}'} = C_1 \restriction_{\mathcal{D}'} \vee C_2 \restriction_{\mathcal{D}'}$.

We now make formal the fact that a resolution refutation for unsatisfiability also proves $\mathcal{D}'$-unsatisfiability. This is achieved by restricting resolution proofs on some variables and values.

Definition 5.6. For a derivation $\pi = \langle C_1, \ldots, C_s \rangle$ of CNF($\mathcal{I}$), the *$\mathcal{D}'$-restriction of π* is $\pi \restriction_{\mathcal{D}'} = \langle C'_1, \ldots, C'_s \rangle$, where $C'_i = C_i \restriction_{\mathcal{D}'}$ for $1 \leq i \leq s$.

In the restricted derivation $\pi \restriction_{\mathcal{D}'}$, it will be convenient to introduce a *weakening rule* in which a clause C'_i may be derived from an earlier clause C'_j, $j < i$, by applying: $C'_i = C'_j \vee C$ for some arbitrary clause C. Given any resolution proof using the weakening rule, it is straightforward to transform it to a shorter one without using the weakening rule, so the use of weakenings does not reduce resolution complexity.

Proposition 5.7. *The $\mathcal{D}'$-restriction of any refutation of $\mathcal{I}$ is a refutation of $\mathcal{I} \restriction_{\mathcal{D}'}$ of no greater length.*

Proof. Let $\pi = \langle C_1, \ldots, C_s \rangle$ be a resolution refutation of $\mathcal{I}$, and let $\pi' = \pi \restriction_{\mathcal{D}'}$ be its $\mathcal{D}'$-restriction. By definition π' is not longer than π. It remains to check that π' is a valid resolution refutation of CNF($\mathcal{I} \restriction_{\mathcal{D}'}$). Consider a clause C_i in π. If $C_i \in \text{CNF}(\mathcal{I})$, then the corresponding clause $C'_i = C_i \restriction_{\mathcal{D}'}$ is in CNF($\mathcal{I}$) $\restriction_{\mathcal{D}'}$ = CNF($\mathcal{I} \restriction_{\mathcal{D}'}$). If C_i is derived from $C_j = A \vee y$ and $C_{j'} = B \vee \overline{y}$ by resolving a variable y, there are two cases depending on whether y is restricted by $\restriction_{\mathcal{D}'}$. If y is not restricted, then $C'_i = (A \vee B) \restriction_{\mathcal{D}'} = A \restriction_{\mathcal{D}'} \vee B \restriction_{\mathcal{D}'}$ is a valid resolution step, because it comes from earlier clauses $C'_j = (A \vee y) \restriction_{\mathcal{D}'} = A \restriction_{\mathcal{D}'} \vee y$ and $C'_{j'} = (B \vee \overline{y}) \restriction_{\mathcal{D}'} = B \restriction_{\mathcal{D}'} \vee \overline{y}$. If y is restricted under $\restriction_{\mathcal{D}'}$, then y must be restricted to be false, and the clause $C'_i = A \restriction_{\mathcal{D}'} \vee B \restriction_{\mathcal{D}'}$ is derived from the

earlier one $C'_j = (A \vee y) \restriction_{\mathcal{D}'} = A \restriction_{\mathcal{D}'}$ via a weakening step. Finally, the last clause $C'_s = C_s \restriction_{\mathcal{D}'}$ is the empty clause. Thus $\pi \restriction_{\mathcal{D}'}$ is indeed a resolution refutation of CNF($\mathcal{I} \restriction_{\mathcal{D}'}$). □

Lemma 5.8. *Consider any $\mathcal{I} \in \Omega(\operatorname{supp} \mathcal{P})$ on n variables. If for some $\alpha, \xi > 0$, we have*

(a) every subproblem on at most αn variables is $\mathcal{D}'$-satisfiable, and

(b) every subproblem $\mathcal{I}'$ on v variables where $\frac{1}{2}\alpha n \leq v \leq \alpha n$ has $|\mathcal{B}(\mathcal{I}')| \geq \xi n$,

then ***C-RES***$(\mathcal{I}) \geq 2^{\Omega(n)}$.

Proof. Consider any resolution refutation of CNF($\mathcal{I} \restriction_{\mathcal{D}'}$). Mitchell ([36], Lemma 1) proves that hypothesis (a) implies there must be a clause C in the refutation and a subproblem $\mathcal{J}$ of $\mathcal{I} \restriction_{\mathcal{D}'}$ on between $\frac{1}{2}\alpha n$ and αn variables, such that $\mathcal{J}$ minimally implies C in the following sense: (i) Every satisfying $\mathcal{D}'$-assignment of $\mathcal{J}$ satisfies C, and (ii) for any subproblem $\mathcal{J}'$ of $\mathcal{J}$, there is a satisfying $\mathcal{D}'$-assignment of $\mathcal{J}'$ that does not satisfy C.

We will prove that C must have at least $\xi n/t$ variables (where t is defined at the beginning of this section, and is the length of the pendant paths in $\mathcal{B}^2$). Using this, the standard "width lemma" of Ben-Sasson and Wigderson ([12], Corollary 3.6) implies that **C-RES**($\mathcal{I} \restriction_{\mathcal{D}'}$) $\geq 2^{\Omega(n)}$. Propostion 5.7 implies that **C-RES**$(\mathcal{I}) \geq 2^{\Omega(n)}$ as well.

Consider any clause $X \in \mathcal{B}^1(\mathcal{J})$; we will show that some variable of X appears in C. To see this, consider any $\mathcal{D}'$-assignment α which satisfies $\mathcal{J} - X$ but not C. By Lemma 5.3, it is possible to extend α to a $\mathcal{D}'$-satisfying assignment α' of $\mathcal{J}$, and since $\mathcal{J}$ implies C, α' satisfies C. Thus, there is some variable of C that is assigned a value in α' but not α. This variable must be in X.

A similar argument shows that C contains a non-endpoint variable of every member of $\mathcal{B}^2(\mathcal{J})$; we use the fact that every $\mathcal{D}'$-assignment to the endpoints of a member X of $\mathcal{B}^2(\mathcal{J})$ can be completed to a satisfying $\mathcal{D}'$-assignment of X. No variable can be a non-endpoint variable of more than t members of $\mathcal{B}^2(\mathcal{J})$. So $|C| \geq |\mathcal{B}^1(\mathcal{J})| + |\mathcal{B}^2(\mathcal{J})|/t$. Now by hypothesis (b), $\mathcal{B}(\mathcal{J}) \geq \xi n$, we have $|C| \geq |\mathcal{B}(\mathcal{J})|/t$, as required. □

We need the following two lemmas from [39].

Lemma 5.9 (Lemma 11, [39])**.** *Let H be a non-empty k-uniform hypergraph on n vertices and m edges that does not have any component which is a cycle. Let B_1 be the set of edges which have at most one vertex of degree greater than 1, and B_2 be the set of pendant paths of length t. If $|B_1| + |B_2| \leq n/(72t^2k^3)$, then $m \geq n(\frac{1+\delta}{k-1})$ for $\delta = \frac{1}{3tk^2}$.*

Lemma 5.10 (Lemma 10, [39])**.** *Let $c > 0$ and $k \geq 2$, and let H be the random k-uniform hypergraph with $n > 0$ vertices and $m = cn$ edges. Then for any $\delta > 0$, there exists $\alpha = \alpha(c, k, \delta) > 0$ such that a.s. H has no subgraph with $0 < h \leq \lfloor \alpha n \rfloor$ vertices and at least $(\frac{1+\delta}{k-1})h$ edges.*

We can finally prove Theorem 1.2. The proof closely resembles that of Theorem 1 from [39].

Proof of Theorem 1.2. Recall t from Fact 5.1. It is well known that the probability that the constraint hypergraph of $\mathsf{CSP}_{n,M}(\mathcal{P})$ has no cycle of length less than t is at least some $\epsilon > 0$ depending only on c and t. (See eg. Theorem 3.19 of [26]).

We will show that condition (a) of Lemma 5.8 holds with probability at least $\epsilon + o(1)$, and that condition (b) holds a.s. with $\alpha = \alpha(c, k, \delta = 1/(3tk^2))$ from Lemma 5.10 and $\xi = \min\{1/(72t^2k^3), \alpha/3k\}$.

We begin with condition (a). Suppose $\mathcal{J}$ is a minimally $\mathcal{D}'$-unsatisfiable subproblem of $\mathsf{CSP}_{n,M}(\mathcal{P})$. Thus, $|\mathcal{B}^1(\mathcal{J})| = |\mathcal{B}^2(\mathcal{J})| = 0$, and the constraint hypergraph of $\mathcal{J}$ is connected. Furthermore, with probability at least ϵ, $\mathsf{CSP}_{n,M}(\mathcal{P})$ is such that the constraint hypergraph of $\mathcal{J}$ cannot be a cycle of length less than t and hence cannot be a cycle at all since, by Fact 5.1, every cycle of length greater than t is $\mathcal{D}'$-satisfiable. Therefore Lemma 5.9 applies to the constraint hypergraph of $\mathcal{J}$ and so $\mathcal{J}$ has clause-variable ratio at least $(1+\delta)/(k-1)$. Thus Lemma 5.10 implies that with probability at least $\epsilon + o(1)$, $\mathsf{CSP}_{n,M}(\mathcal{P})$ has no minimally unsatisfiable subproblems of size at most αn. Equivalently, all subproblems of size at most αn are satisfiable.

Next, condition (b). We will use the well-known fact that a.s. the underlying random hypergraph of $\mathsf{CSP}_{n,M}(\mathcal{P})$ has fewer than $\log n$ cycles of length at most t (this follows from a simple application of Markov's inequality, or from a proof nearly identical to that of Theorem 3.19 of [26]). Suppose, by contradiction, that $\mathcal{J}$ is a subproblem of $\mathsf{CSP}_{n,M}(\mathcal{P})$ with v variables where $\frac{1}{2}\alpha n \le v \le \alpha n$, and with $|\mathcal{B}^1(\mathcal{J})| + |\mathcal{B}^2(\mathcal{J})| \le \xi n$. Let H' be the subhypergraph obtained by removing all the cycle components from the constraint hypergraph of $\mathcal{J}$. By Lemmas 5.10 and 5.9, a.s. for every such subproblem $\mathcal{J}$, H' is empty, and so every component in the constraint hypergraph of $\mathcal{J}$ is a cycle. But any such cycle of length $\ell > t$ will contain at least ℓ members of $\mathcal{B}^2(\mathcal{J})$ and so there are at most $k\xi n$ vertices in those cycles. And, as mentioned above, there are a.s. at most $t \log n$ vertices which lie in cycles of length at most t in $\mathsf{CSP}_{n,M}(\mathcal{P})$. Since $k\xi n + t\log n < \frac{1}{2}\alpha n \le |\mathcal{J}|$, we have a contradiction. □

Consider a constraint set $\mathcal{C}$ with incomplete closures such that there are unsatisfiable cyclic CSP's of size ℓ whose constraints are from $\mathcal{C}$. Note that we can take $\ell = O(1)$ since ℓ depends on $\mathcal{C}$ and not on n. For the case $k = 2$ (i.e. graphs), Theorem 3.19 of [26] (found originally in [13] and [27]) implies that the constraint hypergraph of $\mathsf{CSP}_{n,M}$ w.u.p.p. contains a cycle of length ℓ. It is straightforward to adapt the proof of Theorem 3.19 of [26] to random hypergraphs and hence to prove that the same holds for any fixed k. Under any $\mathcal{P}$ with $\mathrm{supp}(\mathcal{P}) = \mathcal{C}$, such a cycle will receive the constraints from $\mathcal{C}$ that make it unsatisfiable w.u.p.p. If this occurs, then there will be an $O(1)$-length resolution proof that the $O(1)$-sized cyclic sub-CSP is unsatisfiable and hence that the entire CSP is unsatisfiable. Consequently, w.u.p.p. $\mathsf{CSP}_{n,M}(\mathcal{P})$ has $O(1)$ resolution complexity. Thus "with uniformly positive probability" in the statement of Theorem 1.2 cannot be changed to "almost surely".

An example of such a $\mathcal{C}$ is the following: Take $\mathcal{D} = \{1, \ldots, d\}$, $k = 2$, $C = \{(\delta, \gamma) \mid \gamma - \delta \not\equiv 1 \text{ or } 2 \pmod d\}$ and consider the constraint set $\mathcal{C}$ formed by C and its reflection. Then the cycle consisting of ℓ copies of C is unsatisfiable for any $\ell \le (d-1)/2$. On the other hand, $\mathcal{D}$ is null-constraining - consider all constraint paths of length at least $t = d - 1$.

Nevertheless, if we assume there is some null-constraining subdomain $\mathcal{D}'$ such that all cyclic CSP's are $\mathcal{D}'$-satisfiable, we can prove a.s. exponential resolution complexity for all $c > 0$; this is Theorem 1.4.

Proof of Theorem 1.4(a). The proof is the same as that of Theorem 1.2, except that the hypothesis ensures that a minimally unsatisfiable subproblem cannot be cyclic. This implies that condition (a) a.s. holds. □

Remark 5.11. Theorems 1.2 and 1.4 also hold when we measure resolution complexity by **NG-RES**. Indeed, Mitchell [35] shows that **C-RES**$(\mathcal{I}) \le \mathrm{poly}($**NG-RES**$(\mathcal{I}))$. Hence an exponential lower bound for **C-RES** translates to another for **NG-RES**.

5.2 Polylogarithmic Complexity

Let $\mathcal{C} = \text{supp}(\mathcal{P})$ and $\text{cl}(\mathcal{C})$ be the closure of $\mathcal{C}$. Suppose that for every null-constraining $\mathcal{D}' \subseteq \mathcal{D}$, there is a $\mathcal{D}'$-unsatisfiable cyclic CSP formed from $\text{cl}(\mathcal{C})$. We will prove here that if c is a sufficiently large constant then $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ w.u.p.p. has polylogarithmic resolution complexity; i.e. we will prove Theorem 1.4(b).

It will often be convenient to study the CSP formed by taking the union of several independent random CSP's. The following technical lemma will be useful:

Lemma 5.12. *Consider any constants c, t and any $M_1 + ... + M_t = M = cn$. Let $F_1, ..., F_t$ be random CSP's drawn from $\mathsf{CSP}_{n,M_1}(\mathcal{P}), ..., \mathsf{CSP}_{n,M_t}(\mathcal{P})$, and consider the CSP $F = \cup_{i=1}^t F_i$. If a property holds a.s. for F then it holds a.s. for $\mathsf{CSP}_{n,M}(\mathcal{P})$.*

Proof. Let E_1 be the event that no k-tuple of variables is selected for a constraint in two of $F_1, ..., F_t$. Thus, F conditional on E_1 has the same distribution as $\mathsf{CSP}_{n,M}$. We argue below that $\mathbb{P}(E_1) \geq \zeta$ for some constant $\zeta > 0$. It follows then that if a property holds a.s. for F then it must hold a.s. for F conditioned on E_1; i.e. for $\mathsf{CSP}_{n,M}$.

The probability that F_i does not contain any k-tuples from $F_1, ..., F_{i-1}$ is

$$\binom{\binom{n}{k} - (M_1 + ... + M_{i-1})}{M_i} \Big/ \binom{\binom{n}{k}}{M_i} > \psi,$$

for some constant $\psi > 0$ dependent on c, k. Therefore $\mathbb{P}(E_1) > \zeta = \psi^t$, as required. □

Remark: If $k \geq 3$ then in fact $\mathbb{P}(E_1) = 1 - o(1)$ and so we obtain something even stronger. But we do not use that in this paper.

We will make use of the following easy concentration bound:

Lemma 5.13. *Consider selecting s independent and uniform subsets $X_1, ..., X_s$ of $\{v_1, ..., v_n\}$, where $|X_i|$ is fixed to be x_i. Let Z be a random variable determined by $X_1, ..., X_s$ with the property that for every $i, a \in X_i, b \notin X_i$, removing a from and adding b to X_i can affect Z by at most c. Then for any $t > 0$*

$$\mathbb{P}(|Z - \mathbb{E}(Z)| > t) \leq 2e^{-t^2/2c^2 \sum x_i}.$$

The proof follows easily from, eg. Azuma's Inequality[7] (or see [6]) by choosing each X_i one element at a time, without replacement.

We will begin our proof by showing that if $\delta \not\sim_{\mathcal{C}} \gamma$ then for c sufficiently large, a.s. $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ contains many (δ, γ)-forbidding petals.

Lemma 5.14. *Suppose that $\delta \not\sim_{\text{cl}(\mathcal{C})} \gamma$. Then there exists constants c, a such that a.s. $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ contains at least $\frac{n^2}{4}$ pairs of variables x, y that have a (δ, γ)-forbidding petal from x to y of size $O(\log^a n)$.*

Before proving this lemma, we show how it implies our theorem:

Proof of Theorem 1.4(b) By Lemma 2.13, $\mathcal{D}$ can be partitioned into $\mathcal{D}_1, ..., \mathcal{D}_t, W$ such that (i) each $\mathcal{D}_i$ is null-constraining and (ii) for every pair $\delta, \gamma \in \mathcal{D}$ not both lying in the same $\mathcal{D}_i$, we have $\delta \not\sim_{\text{cl}(\mathcal{C})} \gamma$. Label the pairs from (ii) as $(\delta_1, \gamma_1), ..., (\delta_\ell, \gamma_\ell)$ for some $\ell \leq d^2$.

We first choose ℓ random CSP's from $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$: $F_1, ..., F_\ell$, each on the same set of variables, and we let F be the CSP formed by the union of the constraints from $F_1, ..., F_\ell$. We will analyze F rather than $\mathsf{CSP}_{n,M=\ell cn}(\mathcal{P})$, using Lemma 5.12.

By Lemma 5.14 there exists c such that a.s. for each $1 \leq i \leq \ell$, F_i contains at least $\frac{n^2}{4}$ pairs of variables x, y with a (δ_i, γ_i)-forbidding petal from x to y of size $O(\log^a n)$. Since no variable lies in more than n such pairs, it follows that there is a set X_i of $\frac{n}{100\ell}$ variables, such that for each $x \in X_i$ there are at least $\frac{n}{8}$ variables y such that F_i contains a (δ_i, γ_i)-forbidding petal from x to y. Set $\alpha = \frac{1}{2}(\frac{1}{9})^\ell$.

Claim *A.s. there is a variable x, and a set of variables Z of size at least αn such that: for each $1 \leq i \leq \ell$, and for any $z \in Z$, F_i has a (δ_i, γ_i)-forbidding petal from x to z.*

Proof of Claim: For each $1 \leq i \leq \ell$ we can choose F_i by first selecting a CSP F_i' from $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ and then take a uniformly random permutation of its variables; note that this yields the correct distribution for F_i. Select all the F_i''s and suppose that they each contain a set X_i' with the properties of X_i described above (as they a.s. do). Then we expose the random permutations in parts. First map the variables of each X_i' thus yielding X_i. It is straightforward to argue that a.s. there is a variable x that lies in every X_i. Let Y_i' be the set of variables $u \notin X_i'$ such that F_i' contains a (δ_i, γ_i)-forbidding petal from x to u. Note that $|Y_i| \geq \frac{n}{8} - |X_i'| > \frac{n}{9}$. Let Z^+ denote the set of variables that do not lie in X_i for any $1 \leq i \leq \ell$; note that $|Z^+| \geq n - \sum |X_i| = \frac{99}{100}n$. Next, for each i, choose the mappings of the variables of Y_i', thus yielding Y_i. We set $Z = \cap_{i=1}^{\ell} Y_i$; the expected size of Z is at least $|Z^+| \times (\frac{1}{9})^\ell > \frac{3}{2}\alpha$. A straightforward concentration argument, using the independence of the sets Y_i, shows that a.s. $|Z| \geq \alpha n$. Note that Z meets the condition of our claim. □

Now we choose a random CSP from the distribution $\mathsf{CSP}_{M=(\ell+t)cn}(\mathcal{P})$ as follows. First pick a random $\mathsf{CSP}_{n,M=\ell cn}(\mathcal{P})$ and label it H_0. Next, for each $1 \leq j \leq t$, we let H_j be a random CSP with cn constraints whose variables are drawn uniformly and without replacement from the k-tuples of variables that do *not* form constraints in $H_0 \cup ... \cup H_{j-1}$, and whose constraints are selected from $\mathcal{P}$. Thus $H_0 \cup ... \cup H_t$ has the same distribution as $\mathsf{CSP}_{M=(\ell+t)cn}$

By our Claim, and Lemma 5.12, a.s. H_0 has a variable x and a set Z as described in the Claim.

For each $1 \leq j \leq t$, the hypothesis of our theorem states that there is a $\mathcal{D}_i$-unsatisfiable cyclic CSP formed from the constraints of $\mathcal{C}$. Note that its size is independent of n. A standard and straightforward method of moments analysis along the lines of that from, eg. the aforementioned Theorem 3.19 of [26], implies: There is an $\epsilon > 0$ such that for each $1 \leq j \leq t$, with probability at least ϵ, H_j contains a copy C_j of that cyclic CSP with its variables all lying in Z. So the probability that this holds for every $1 \leq j \leq t$ is at least ϵ^t; suppose this does indeed hold. Let z be any variable from Z. Then we define $F^* \subset F$ to be the CSP formed by:

- $x, z, C_1, ..., C_t$;
- for each $1 \leq j \leq t$ and each $\delta \in \mathcal{D}_j$ and $\gamma \notin \mathcal{D}_j$, the (δ, γ)-forbidding petals from x to each vertex in C_j;
- for each $\delta \in W$ and $\gamma \in \mathcal{D}$, the (δ, γ)-forbidding petal from x to z.

Note that F^* is not satisfiable. Indeed, if x takes a value from some $\mathcal{D}_j$ then, since $C_j \subset Z$ the petals from our Claim imply that each variable in C_j must take a value from $\mathcal{D}_j$; but C_j is $\mathcal{D}_j$-unsatisfiable. The only other possibility is for x to take a value from W, but then the petals from our Claim forbid z from taking any value. Furthermore, the fact that these petals each have size $O(\log^a n)$ easily implies that F^* has a resolution refutation of size $O(\log^a n)$, very similar to the refutation from the proof of Theorem 1.1. □

Remark: Note that when $t = 0$ the proof still holds, and in fact, we get a.s. rather than w.u.p.p. Thus, this provides an alternate proof for Theorem 1.2, albeit for a much higher value of c.

We complete the proof by proving Lemma 5.14. First, we need some definitions and lemmas.

Suppose we are given a directed graph H whose vertices are a subset of the variables of a CSP, F. Consider any constraint path $P = \beta_1, ..., \beta_h$ in F where the constraint β_i is on variables $x_{i,1}, ..., x_{i,k}$, the endpoints are $x_{1,1}, x_{h,k}$ and the other connecting variables are $x_{i,k} = x_{i+1,1}$ for $1 \leq i \leq h-1$. We define the *first* and *last* variables of β_i to be $x_{i,1}, x_{i,k}$.

Definition 5.15. We say that β_i *respects* H if there is an edge in H from its first variable to its last variable. We say that P *respects* H if every β_i respects H.

This definition will be useful in constructing petals: H will represent pairs of vertices that are joined by certain petals, and so a path respecting H can be the main path of a larger petal.

Recall that a P^i-path in a CSP is defined (in Section 2) to be the constraint path $PPP...P$ consisting of i concatenated copies of P.

Lemma 5.16. *Let H be any fixed directed graph where $V(H) \subseteq \{v_1, ..., v_n\}$ and H has minimum indegree and minimum outdegree at least $\frac{1}{2}n$. Let P be any constraint path over $\mathcal{C}$. There exists constants c, Q and $\eta > 0$ such that a.s., $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ (on variables $v_1, ..., v_n$) contains two sets of variables X, Y with $|X|, |Y| \geq \eta n$ such that for every $x \in X, y \in Y$ there is a P^i-path from x to y that respects H for some $i \leq Q \log n$.*

Proof. It will be convenient to work in the $\mathsf{CSP}_{n,p}$ model; Remark 1.6 permits us to do so.

Suppose that the constraints of P are $\beta_1, ..., \beta_h$.

We use a variation of a standard branching process argument (see eg. Section 5.2 of [26], or [28] where such arguments were introduced to random graph theory). We explore the random CSP via two parallel breadth-first type searches from an arbitrary vertex $v \in V(H)$, in which we search for P^i-paths starting at v or ending at v. We initialize $L_1 = \{v\}$, $R_1 = \{v\}$ and $T = \emptyset$. For the first iteration, we expose all β_1 constraints respecting H in which v is the first variable, add those constraints to T, and let L_2 be the set of last vertices from these constraints. We also expose all β_h constraints respecting H in which v is the last variable, add those constraints to T, and let R_2 be the set of first vertices from these constraints. During iteration i: for each $u \in L_i$ (one-at-a-time) we expose all $\beta_{i(\mod h)}$ constraints whose first variable is u and whose other variables do not lie in any constraints of T. We add all of those constraints to T and we add the last vertices to L_{i+1}. We then process each $u \in R_i$ and build R_{i+1} in the obvious analogous way. We repeat until $|T| \geq \frac{n}{3}$ or $L_i = \emptyset$ or $R_i = \emptyset$. In the first case we halt. In each of the last two cases, we pick another vertex $u \in V(H) \setminus V(T)$ and set $i = 1$, $L_1 = R_1 = \{u\}$ and $L_j = R_j = \emptyset$ for all $j > 1$; i.e. we restart the process except that we do not remove anything from T.

We define the *height* of a variable in T to be its distance, in T, to the root of its component; i.e. the variable that was selected to initiate that component. We set $L = \cup_{i \geq 0} L_i, R = \cup_{i \geq 0} R_i$.

Consider processing a particular variable $u \in L_i$. H has minimum indegree and outdegree at least $\frac{1}{2}n$, and $|T| \leq \frac{1}{3}n$. Therefore, there are at least $\frac{1}{2}n - \frac{1}{3}n$ choices for the last variable of a constraint from u, and at least $\binom{\frac{2}{3}n}{k-2}$ choices for the other variables of the constraint. Thus there are $O(n^{k-1})$ potential constraints, and so if c is sufficiently large, then the expected number of new vertices added to L is larger than one, say at least two, and similarly for R. It follows that the number of unexplored vertices in L, R after j vertices have been processed is at least as high, in

distribution, as two variables ℓ_j, r_j following the random walks ($Po(2)$ denotes a Poisson variable with mean 2):

$$\ell_0 = r_0 = 1; \qquad \ell_{j+1} = \ell_j - 1 + Po(2); \qquad \ell_{j+1} = \ell_j - 1 + Po(2);$$

and if either $\ell_j = 0$ or $r_j = 0$ then both $\ell_{j+1} = r_{j+1} = 1$. At each restart $\ell_{j+1} = 1$, since the drift is positive, the probability that ℓ does not return to zero is a positive constant, and the probability that neither ℓ nor r returns to zero is the square of that constant. So a.s. after $O(1)$ expected restarts, our process will continue until $|T| = \frac{n}{3}$, and a.s. the component being exposed at that point will have $\Theta(n)$ unexplored vertices in each of L, R. Once the number of unexplored vertices becomes linear, the size of each successive level L_i, R_i becomes highly concentrated (as it is essentially a binomial variable). The sizes of these levels tend to grow by a constant factor, and so we a.s. reach $|T| \geq \frac{n}{3}$ before reaching level $i = Q \log n$ for some constant Q. It follows that for some constants $Q, \eta > 0$ a.s.

(i) at least ηn vertices in R have a height that is divisible by h and is at most $Q \log n$;

(ii) at least ηn vertices in L have a height that is divisible by h and is at most $Q \log n$;

The two groups (i) and (ii) form the sets X, Y required by our lemma. □

We now build on Lemma 5.16 to obtain the following strengthening:

Lemma 5.17. *Let H be any fixed directed graph where $V(H) \subseteq \{v_1, ..., v_n\}$ and H has minimum outdegree and minimum indegree at least $\frac{1}{2}n$. Let P_A, P_B, P_C be any constraint paths over $\mathcal{C}$. For any constant $\varphi > 0$, there exists c, Q such that $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ (on variables $v_1, ..., v_n$) a.s. contains a set of variables Z with $|Z| \geq (1-\varphi)n$ such that for each $z \in Z$:*

(i) there are at least $(1-2\varphi)n$ variables x such that there is an H-respecting constraint path from x to z of the form $P_A P_B^i P_C$ for some $i \leq Q \log n$; and

(ii) there are at least $(1-2\varphi)n$ variables y such that there is an H-respecting constraint path from z to y of the form $P_A P_B^i P_C$ for some $i \leq Q \log n$.

Proof. We will first form a random CSP F by taking the union of three random CSP's F_A, F_B, F_C each distributed like $\mathsf{CSP}_{n,M=c_1 n}$, for some sufficiently large c_1.

First, we note that for some $\xi > 0$, there are a.s. at least ξn disjoint copies of P_A in F_A. One way to see this is: the expected number of components of the underlying hypergraph which are isomorphic to P_A is easily determined to be linear in n (indeed, this is true of components isomorphic to any constant-sized sub-CSP whose underlying hypergraph is a hypertree). The number of such components is concentrated around its mean by a very simple second moment argument. The same argument shows that there are a.s. at least ξn disjoint copies of P_C in F_C.

Lemma 5.16, with $P = P_B$ and H equal to the complete directed graph, implies that for some constants $Q, \eta > 0$: F_B a.s. contains X, Y with $|X|, |Y| = \eta n$ such that for every $x \in X, y \in Y$ there is a P_B^i-path from x to y for some $i \leq Q \log n$.

Expose F_A, F_B, F_C and assume that the a.s. events described above all hold. Then take uniformly random permutations of the variables of F_A, F_B, F_C; note that this does not affect the distribution of these CSP's and hence of F.

We wish to lower-bound the number of P_A paths in F_A that end at a variable in X. The vertices of X are determined by F_B. Upon taking the random permutation of variables in F_A, the endpoints of those P_A paths form a random subset of the variables, of size at least ξn. Since $|X| \geq \eta n$, the expected number of those endpoints that are in X is at least $\xi\eta n$. Lemma 5.13 implies that number is a.s. at least $\frac{1}{2}\xi\eta n$. Similarly, a.s. at least $\frac{1}{2}\xi\eta n$ of the P_C paths in F_C begin at a variable in Y. Each of the $\frac{1}{4}\xi^2\eta^2 n^2$ pairs of such paths can be completed to a $P_A P_B^i P_C$ path for some $i \leq Q \log n$. Letting X' be the variables at the beginnings of those P_A paths and Y' be the variables at the ends of those P_C paths, we have $|X'|, |Y'| \geq \frac{1}{2}\xi\eta n$ and every pair $x \in X', y \in Y'$ is joined by a path of the form $P_A P_B^i P_C$ for some $i \leq Q \log n$.

Set $\eta' = \frac{1}{2}\xi\eta$; we can assume that η' is a sufficiently small positive constant. Set $s = 100/(\eta')^3$, and form a random CSP F^* by taking the union of s different random CSP's $F_1, ..., F_s$ each distributed as F, above; thus we have $F_i = F_{i,A} \cup F_{i,B} \cup F_{i,C}$, as above. As above, we expose every $F_{i,A}, F_{i,B}, F_{i,C}$ and then take independent random permutations of the variables in each; note that this does not affect the distribution of F^*. Let X_i', Y_i' denote the sets that we proved above to a.s. exist in F_i.

For a variable v, the number of sets X_i' containing v is at least as high, in distribution, as the binomial distribution $BIN(s, \eta')$. This is because the random permutations of the variables imply that $\Pr(v \in X_i') = |X_i'|/n \geq \eta'$, and these events are independent for each i. That binomial has mean $s\eta' = 100/(\eta')^2$. For sufficiently small η', the probability that this binomial is at least $1/(\eta')^2$ is greater than $1 - \varphi/4$. Since the X_i''s are independent subsets of the variables, Lemma 5.13 yields that a.s. at least $(1 - \varphi/2)n$ variables lie in at least $1/(\eta')^2$ sets X_i'; let X^* denote this set of variables.

For any variable v, let $Y(v)$ denote $\cup_{i:v \in X_i'} Y_i'$. Having fixed the variables of each $F_{i,B}$, each X_i' is determined by $F_{i,A}$ and each Y_i' is determined by $F_{i,C}$. Thus, for each $v \in X^*$, $Y(v)$ contains the union of at least $1/(\eta')^2$ independent random subsets, each of size at least $\eta' n$. So $\mathbb{E}(|Y(v)|) \geq (1 - (1 - \eta')^{1/(\eta')^2})n \geq (1 - \varphi/4)n$ for η' sufficiently small, and Lemma 5.13 implies that a.s. $Y(v) \geq (1 - \varphi/2)n$ for each $v \in X^*$.

For any variable u, let $X(u)$ denote $\cup_{i:u \in Y_i'} X_i'$. The same argument shows that a.s. there is a set Y^* with $|Y^*| \geq (1 - \varphi/2)n$ where each $u \in Y^*$ has $|X(u)| \geq (1 - \varphi/2)n$. Let $Z = X^* \cap Y^*$, thus $|Z| \geq (1 - \varphi)n$. For any $z \in Z$, $|X(z) \cap Z| \geq |X(z)| - \varphi n > (1 - 2\varphi)n$ and similarly, $|Y(z) \cap Z| \geq (1 - 2\varphi)n$. This establishes that the lemma holds for F^*; Lemma 5.12 implies that it holds for $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ with $c = 3sc_1$. □

Proof of Lemma 5.14: If $\delta \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma$, then by Proposition 3.13 we can form arbitrarily long (δ, γ)-forbidding petals using the constraints of $\mathcal{C}$. Let P^* be such a petal of length greater than 2^d; so P^* is a $(\mathcal{T}_{\delta,\gamma}, \mathcal{C})$-petal for some configuration tree $\mathcal{T}_{\delta,\gamma}$. We will prove by induction on the height of this configuration tree that:

Claim 5.18. *For any $\varphi > 0$ there exists c, a such that $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ a.s. contains a set of variables Z with $|Z| \geq (1 - \varphi)n$ where for every $z \in Z$:*

(i) there are at least $(1 - 2\varphi)n$ variables x such that there is a (δ, γ)-forbidding petal from x to z of size $O(\log^a n)$; and

(ii) there are at least $(1 - 2\varphi)n$ variables y such that there is a (δ, γ)-forbidding petal from z to y of size $O(\log^a n)$.

Clearly this claim implies the lemma.

Let P be the main path of the petal. Let Δ be the set of labels of the children of the root of $\mathcal{T}_{\delta,\gamma}$. ($\Delta = \emptyset$ if the root has no children.) Thus $P \cup \Delta$ (recall Definition 3.3) is (δ,γ)-forbidding. Since $|P| > 2^d$, the proof of Corollary 2.11 implies that P must be decomposable into $P_A P_B P_C$ such that for any $i \geq 0$, $P_A P_B^i P_C \cup \Delta$ is (δ,γ)-forbidding

For height 1, $P^* = P$ and the Claim follows from Lemma 5.17 where H is the complete directed graph. For height $t > 1$, let $u_1, ..., u_s$ be the children of the root and let (δ_i, γ_i) be the label of u_i; thus $\Delta = \{(\delta_1,\gamma_1), ..., (\delta_s,\gamma_s)\}$. For any given $\varphi > 0$, we inductively apply the claim substituting φ/s for φ to obtain that (since $s = O(1)$) there exists c_1, a_1, such that $\mathsf{CSP}_{n,M=c_1 n}$ a.s. contains, for each $1 \leq i \leq s$, a set Z_i of size at least $(1-\varphi/s)n$ such that for every $z \in Z_i$:

(i) there are at least $(1-2\varphi/s)n$ variables x such that there is a (δ_i,γ_i)-forbidding petal from x to z of size $O(\log^{a_1} n)$; and

(ii) there are at least $(1-2\varphi/s)n$ variables y such that there is a (δ_i,γ_i)-forbidding petal from z to y of size $O(\log^{a_1} n)$.

We define a directed graph H as follows: Set $V(H) = \cap_{i=1}^s Z_i$, so $|V(H)| \geq (1-\varphi)n$. We direct an edge from u to v if for *every* $1 \leq i \leq s$, there is a (δ_i,γ_i)-forbidding petal from u to v of size $O(\log^{a_1} n)$.

Since $|Z_i - V(H)| < \varphi n$ for every i, it follows from (ii) above that every vertex in H has outdegree at least $n - \varphi n - s \times (2\varphi/s) > \frac{1}{2}n$. Similarly it follows from (i) that every vertex in H has indegree at least $\frac{1}{2}n$.

Note that any H-respecting constraint path of the form $P_A P_B^i P_C$ is the main path of a (δ,γ)-forbidding petal, and the size of that petal will be $O(\log^{a_1} n)$ times the length of the path. Therefore, the claim follows inductively from Corollary 5.17. □

Remark: Note that this proof yields that the exponent a in Theorems 1.1 and 1.4(b) is equal to the height of the configuration tree $\mathcal{T}_{\delta,\gamma}$.

We close by showing how to find these poly($\log n$) resolution proofs in polynomial time, when c is sufficiently large.

Proof of Theorem 1.5: By Lemma 2.13, $\mathcal{D}$ can be partitioned into $\mathcal{D}_1, ..., \mathcal{D}_t, W$ such that (i) each $\mathcal{D}_i$ is null-constraining and (ii) for every pair $\delta, \gamma \in \mathcal{D}$ not both lying in the same $\mathcal{D}_i$, we have $\delta \not\sim_{\mathcal{C}} \gamma$. By Lemma 3.15, we can find this decomposition in $O(1)$ time.

Consider any $\delta, \gamma \in \mathcal{D}$ such that $\delta \not\sim_{\mathcal{C}} \gamma$. By Lemma 3.15, in $O(1)$ time we can construct constraint paths P_A, P_B, P_C over $\mathcal{C}$ such that for every $i \geq 0$, $P_A P_B^i P_C$ is (δ,γ)-forbidding. It is straightforward to determine, in polynomial time, all pairs of variables in a CSP F that are joined by such paths. Indeed, since $|P_A| = O(1)$, it is easy to construct a directed graph G_A such that (x,y) is an edge in G_A iff there is a P_A-path from x to y in F, and similarly to construct G_B, G_C. From those graphs, construct a directed graph G^* such that (x,y) is an edge in G^* iff there exist z_1, z_2 such that: (i) $(x,z_1) \in E(G_A)$; (ii) there is a path from z_1 to z_2 in G_B; and (iii) $(z_2,y) \in E(G_C)$. $E(G^*)$ is the set of all pairs of variables that are joined by paths of the form $P_A P_B^i P_C$.

Now consider any $\delta', \gamma' \in \mathcal{D}$ such that $\mathcal{C}$ can form a (δ',γ')-forbidding petal from a configuration tree T with height 2. Using a techniques similar to that of the previous paragraph, and using the graphs G^* formed in the previous paragraph it is straightforward to find all pairs of variables linked by a petal of the type described by T. The main difference is that we need to list pairs joined by, eg. P_A-paths in which the first and last variables of each constraint are linked by a path that allows

us to build the main path up into a petal. The graphs G^* formed in the previous paragraph allows us to check quickly whether the first and last variables of a potential constraint are so linked.

Carrying on iteratively, for every $\delta, \gamma \in \mathcal{C}$ with $\delta \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma$, we can list all pairs of variables that are linked by a (δ, γ)-forbidding petal of the sort used to prove Theorem 1.4(b) above.

If $t = 0$ (i.e. if $\delta \not\sim_{\mathrm{cl}(\mathcal{C})} \gamma$ for all $\delta, \gamma \in \mathcal{D}$) then these lists will easily allow us to identify a forbidding flower - simply find a pair of variables that appears in every list. Note that this does not reveal every possible forbidding flower; only the sort whose paths are of the form used in the proof of Theorem 1.4(b). But that proof shows that for c sufficiently large, $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ will a.s. have such a flower. (In fact, a careful examination of the proof of Theorem 1.1 will show that the flowers guaranteed in that proof also have this form; so they will also be discovered by this algorithm.)

For $t \geq 1$, we apply Lemma 3.15 to produce a list of cyclic CSP's $C_1, ..., C_t$ such that each C_i is not $\mathcal{D}'$-satisfiable. Since each C_i has size $O(1)$, there are at most a polynomial number of them in F, and they can be found by exhaustive search in polytime. For any choice of occurences of $C_1, ..., C_t$ in F and pair of variables x, y, we can check whether they form the structure H described in the proof of Theorem 1.4(b) using the lists of variables that are joined by petals. For c sufficiently large, such an H occurs w.u.p.p. and if it does occur then this algorithm will find it. □

6 Future Work

Corollary 1.3 might, in fact, extend to the stronger statement that for every $\mathcal{P}$ and every c, with the possible exception of some "threshold values" of c, a.s. the shortest resolution refutation of $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ is either exponential or polylogarithmic. This is true for random 2-SAT and for all models studied in [39].

For those models that have property POLY, it is natural to ask for their thresholds of polynomial resolution complexity. [2] and [39] actually determine, for each of their random models for which POLY holds, a precise value c^*, above which the random CSP has a.s. polynomial resolution complexity and below which it has a.s. exponential resolution complexity. We would like to determine such a value for every $\mathsf{CSP}_{n,M=cn}(\mathcal{P})$ for which POLY holds; i.e. for which $\mathrm{cl}(\mathrm{supp}\,\mathcal{P})$ is complete. Upon reading Section 3, some readers may guess that c^* is the threshold for the appearance of the first *forbidding flower*. This is the case for random 2-SAT and for all the models in [2] and [39]. However, we have examples of other models for which it is not the case.

Quite possibly, with sufficient labour, a branching process argument along the lines of that in Section 5.2 might yield the threshold for the appearance of the first forbidding flower.

7 Acknowledgement

We would like to thank anonymous referees and Toniann Pitassi for helpful comments on an earlier draft of this paper.

References

[1] Dimitris Achlioptas, Paul Beame, and Michael Molloy. Exponential bounds for dpll below the satisfiability threshold. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 132–133, 2004.

[2] Dimitris Achlioptas, Paul Beame, and Michael Molloy. A sharp threshold in proof complexity yields lower bounds for satisfiability search. *Journal of Computer and System Sciences*, 2004. An earlier version appeared in the 33rd Annual ACM Symposium on the Theory of Computing (STOC) 2001.

[3] Dimitris Achlioptas, Arthur Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k SAT and NAE 3-SAT. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 721–722, 2001.

[4] Dimitris Achlioptas, Lefteris M. Kirousis, Evangelos Kranakis, and Danny Krizanc. Rigorous results for random $(2 + p)$-SAT. *Theoretical Computer Science*, 265(1-2):109–129, 2001.

[5] Dimitris Achlioptas and Cristopher Moore. Random k-sat: Two moments suffice to cross a sharp threshold. *SIAM Journal on Computing*, 36(3):740–762, 2006.

[6] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley-Interscience, 2nd edition, 2000.

[7] K. Azuma. Weighted sums of certain dependent random variables. *Tokuku Math. Journal*, 19:357–367, 1967.

[8] Paul Beame, Joseph Culberson, David Mitchell, and Cristopher Moore. The resolution complexity of random graph k-colorability. *Discrete Applied Mathematics*, 153(1):25–47, 2005.

[9] Paul Beame, Russell Impagiazzo, and Ashish Sabharwal. Resolution complexity of independent sets in random graphs. In *16th Annual IEEE Conference on Computational Complexity (CCC)*, pages 52–68, 2001.

[10] Paul Beame, Richard M. Karp, Toniann Pitassi, and Michael E. Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, 2002. An earlier version appeared in the 30th Annual ACM Symposium on the Theory of Computing (STOC) 1998.

[11] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 274, Washington, DC, USA, 1996. IEEE Computer Society.

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

[13] B. Bollobás. Random graphs. In *Combinatorics, Proceedings Swansea, London Math. Soc. Lecture Note Ser. 52*, pages 80–102, 1981.

[14] Vašek Chvátal and Bruce Reed. Mick gets some (the odds are on his side). In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1992.

[15] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[16] Harold Connamacher and Michael Molloy. The exact satisfiability threshold for a potentially intractable random constraint satisfaction problem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[17] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing (STOC)*. ACM, New York, 1971.

[18] Nadia Creignou and Hervé Daudé. Generalized satisfiability problems: minimal elements and phase transitions. *Theoretical Computer Science*, 302(1-3):417–430, 2003.

[19] O. Dubois and J. Mandler. The 3-XORSAT threshold. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 769–778, 2002.

[20] Martin Dyer, Alan Frieze, and Michael Molloy. A probabilistic analysis of randomly generated binary constraint satisfaction problems. *Theoretical Computer Science*, 290(3):1815–1828, January 2003.

[21] Seneta Eugene. *Non-negative Matrices and Markov Chains*. Springer-Verlag, New York, 1981.

[22] Alan M. Frieze and Michael Molloy. The satisfiability threshold for randomly generated binary constraint satisfaction problems. *Random Structures and Algorithms*, 28(3):323–339, 2006. An earlier version appeared in the International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM) 2003.

[23] Alan M. Frieze and Nicholas C. Wormald. Random k-SAT: A tight threshold for moderately growing k. *Combinatorica*, 25:297–305, 2005.

[24] Xudong Fu. *On the complexity of proof systems*. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada, 1996.

[25] Ian P. Gent, Ewan Macintyre, Patrick Prosser, Barbara M. Smith, and Toby Walsh. Random constraint satisfaction: Flaws and structure. *Constraints*, 6(4):345–372, 2001.

[26] Svante Janson, Tomasz Łuczak, and Andrzej Ruciński. *Random Graphs*. Wiley-Interscience, 1st edition, 2000.

[27] M. Karoński and A. Ruciński. On the number of strictly balanced subgraphs of a random graph. In *Graph Theory, Proceedings, Lagow, 1981, Lec. Notes in Math. 1018*, pages 79–83.

[28] R. Karp. The transitive closure of a random digraph. *Random Structures and Algorithms*, 1:73–94, 1990.

[29] D. Knuth. *The Art of Computer Programming, Vol. 1*. Addison Wesley, 1969.

[30] James Ledoux, Gerardo Rubino, and Bruno Sericola. Exact aggregation of absorbing markov processes using the quasi-stationary dsitribution. *Journal of Applied Probability*, 31:626–634, 1994.

[31] Colin McDiarmid. On the span of a random channel assignment problem. *Combinatorica*, 27(2):183–203, 2007.

[32] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812–815, August 2002.

[33] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina. Two solutions to diluted p-spin models and XORSAT problems. *Journal of Statistical Physics*, 111(3–4):505–533, May 2003.

[34] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992.

[35] David G. Mitchell. *Resolution Complexity of Constraint Satisfaction*. PhD thesis, University of Toronto, 2002.

[36] David G. Mitchell. Resolution complexity of random constraints. In *Proceedings of Principles and Practices of Constraint Programming*, 2002.

[37] Michael Molloy. Models and thresholds for random constraint satisfaction problems. *SIAM Journal of Computing*, pages 935–949, 2003.

[38] Michael Molloy. When does the giant component bring unsatisfiability? *Combinatorica*, 28(6):693–734, Nov 2008.

[39] Michael Molloy and Mohammad R. Salavatipour. The resolution complexity of random constraint satisfaction problems. *SIAM Journal of Computing*, 37(3):895–922, 2007. An earlier version appeared in the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2003.

[40] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, July 1999.

[41] Jacobus Hendricus van Lint and Richard Michael Wilson. *A Course in Combinatorics*. Cambridge University Press, 2nd edition, 2001.

[42] Ke Xu and Wei Li. Many hard examples in exact phase transitions with application to generating hard satisfiable instances. *Theoretical Computer Science*, 355:291–302, 2006.