

# PPSZ is better than you think

Received Jul 25, 2022  
Revised Oct 19, 2023  
Accepted Feb 4, 2024  
Published Mar 13, 2024

**Key words and phrases**  
Satisfiability, NP-complete problems, randomized algorithm

---

Dominik Scheder<sup>a</sup>  

<sup>a</sup> Hochschule Zittau/Görlitz,  
Germany

**ABSTRACT.** PPSZ, for long time the fastest known algorithm for  $k$ -SAT, works by going through the variables of the input formula in random order; each variable is then set randomly to 0 or 1, unless the correct value can be inferred by an efficiently implementable rule (like small-width resolution; or being implied by a small set of clauses).

We show that PPSZ performs exponentially better than previously known, for all  $k \geq 3$ . We achieve this through an improved analysis and without any change to the algorithm itself. The core idea is to pretend that PPSZ does not process the variables in uniformly random order, but according to a carefully designed distribution. We write “pretend” since this can be done while running the original algorithm, which does use a uniformly random order.

## 1. Introduction

Satisfiability is a central problem in theoretical computer science. One is given a Boolean formula and asked to find a *satisfying assignment*, that is, setting the input variables to 0 and 1 to make the whole formula evaluate to 1. Or rather, determine whether such an assignment exists. A particular case of interest is CNF-SAT, when the input formula is in conjunctive normal form—that is, the formula is an AND of clauses; a clause is an OR of literals; a literal is either a variable  $x$  or its negation  $\bar{x}$ . If every clause contains at most  $k$  literals, the formula is called a  $k$ -CNF formula and the decision problem is called  $k$ -SAT.

Among worst-case algorithms for  $k$ -SAT, two paradigms dominate: local search algorithms like Schönig’s algorithm [17] and random restriction algorithms like PPZ (Paturi, Pudlák, and Zane [9]) and PPSZ (Paturi, Pudlák, Saks, and Zane [8]). Both have a string of subsequent improvements: Hofmeister, Schönig, Schuler, and Watanabe [6], Baumer and Schuler [1], and

---

An extended abstract of this work has already been published [12], and a full version is publicly accessible at [13].

Liu [7] improve Schöning’s algorithm. Hertli [5] and Hansen, Kaplan, Zamir, and Zwick [3] improve upon PPSZ.

For large  $k$ , both paradigms achieve a running time of the form  $2^{n(1-c/k+o(1/k))}$ , where  $c$  is specific to the algorithm ( $c = 1$  for PPZ;  $c = \log_2(e) \approx 1.44$  for Schöning;  $c = \pi^2/6 \approx 1.64$  for PPSZ). Interestingly, the running time of completely different approaches like the polynomial method (Chan and Williams [2]) is also of this form. This gave rise to the Super-Strong Exponential Time Hypothesis (Vyas and Williams [18]), which conjectures that the  $c/k$  in the exponent is optimal; for instance, that a running time of  $2^{n(1-\log(k)/k)}$  is impossible.

This paper presents an improvement of PPSZ. However, it is not an improvement of the algorithm but of its *analysis*. We show that the same algorithm performs exponentially better than previously known. Informally, PPSZ works by going through the variables in random order  $\pi$ ; inspecting each variable  $x$ , it tosses an unbiased coin to determine which value to assign, unless there is a set of at most  $w$  clauses that implies a certain value for  $x$ . Take  $w = 1$  and this is exactly PPZ; take  $w = \omega(1)$  and this is PPSZ (the exact rate by which  $w$  grows turns out to be immaterial for all currently known ways to analyze the algorithm). Our idea is to pretend that the ordering  $\pi$  is not chosen uniformly but from a carefully designed distribution  $D$ . This increases the success probability of PPSZ by some “bonus”, which depends on  $D$ . It seems surprising that this can be done without actually changing the algorithm, but it turns out to be just a straightforward manipulation, which we formally explain below in (3). There is a price to pay in terms of how much  $D$  differs from the uniform distribution: the success probability incurs a penalty of  $2^{-\text{KL}(D||U)}$ , where  $\text{KL}(D||U)$  is the Kullback-Leibler divergence from the uniform distribution  $U$  to  $D$ . We focus on Unique- $k$ -SAT, where the input formula has exactly one satisfying assignment. A “lifting theorem” by Steinberger and myself [14] shows that improving PPSZ for Unique- $k$ -SAT automatically yields a (smaller) improvement for general  $k$ -SAT problem (without changing the algorithm).

The idea of analyzing PPSZ assuming some non-uniform distribution  $D$  on permutations and paying a price in terms of  $\text{KL}(D||U)$  is not new. It is explicit in [14] and implicit in [4] and [8]. However, all previous applications use this to deal with the case that  $\text{sat}(F)$ , the set of satisfying assignments, contains multiple elements; furthermore, in [14, 4, 8], the distribution  $D$  is defined solely in terms of  $\text{sat}(F)$  and ignores the syntactic structure of  $F$  itself. In particular, in the special case that  $F$  has a unique solution,  $D$  reverts to the uniform distribution. This paper is the first work that exploits the structure of  $F$  itself to define a distribution  $D$  on permutations, and uses this to prove a better success probability for the Unique-SAT case.

## 1.1 Analyzing PPSZ: permutations and forced variables

We will now formally describe the PPSZ algorithm. Let  $F$  be a formula,  $x$  a variable, and  $b \in \{0, 1\}$ . A formula  $F$  *implies* ( $x = b$ ) if every satisfying assignment of  $F$  sets  $x$  to  $b$ . For example,

$(x \vee y) \wedge (x \vee \bar{y})$  implies  $(x = 1)$  but neither  $(y = 0)$  nor  $(y = 1)$ . For an integer  $w$ , we say  $F$   $w$ -implies  $(x = b)$  if there is a set  $G$  of at most  $w$  clauses of  $F$  such that  $G$  implies  $(x = b)$ .

**The PPSZ algorithm with strength parameter  $w$ .** Let  $w = w(n)$  be some fixed, slowly growing function. Given a CNF formula  $F$  and a permutation  $\pi$  of its variable set  $V$ , we define  $\text{ppsz}(F, w, \pi)$  as follows: go through the variables  $x_1, \dots, x_n$  in the order prescribed by  $\pi$ . In each step, when handling a variable  $x$ , check whether  $(x = b)$  is  $w$ -implied by  $F$  for some  $b \in \{0, 1\}$ . If so, set  $x$  to  $b$  (i.e., replace every occurrence of  $x$  in  $F$  by  $b$ ). Otherwise, set  $x$  randomly to 0 or 1, with probability  $1/2$  each. We define  $\text{ppsz}(F, w)$  to first choose a uniformly random permutation  $\pi$  and then call  $\text{ppsz}(F, w, \pi)$ .

---

```

1:  procedure ppsz( $F, w, \pi$ )
2:       $V :=$  the set of variables in  $F$ 
3:       $\beta :=$  the empty assignment on  $V$ 
4:      for  $x \in V$  in the order of  $\pi$  do
5:          if there is  $b \in \{0, 1\}$  such  $(x = b)$  is  $w$ -implied by  $F$  then
6:               $\beta(x) := b$ 
7:          else
8:               $\beta(x) :=$  a uniformly random bit in  $\{0, 1\}$ 
9:               $F := F|_{x=\beta(x)}$ 
10:         if  $F$  has been satisfied then
11:             return  $\beta$ 
12:         else
13:             return failure

```

**Algorithm 1.** PPSZ with fixed permutation  $\pi$

---

It should be noted that Paturi, Pudlák, Saks, and Zane in [8] formulated a stronger version of PPSZ, which tries to infer  $(x = b)$  using bounded-width resolution. A close look at their proof shows that they never use properties of resolution beyond those already possessed by  $w$ -implication. We assume that  $\alpha = (1, \dots, 1)$  is the unique satisfying assignment of  $F$ . This is purely for notational convenience.

**DEFINITION 1.1.** Let  $\pi$  be a permutation of  $V$  and  $x$  a variable. Let  $A \subseteq V$  be the set of variables coming before  $x$  in  $\pi$ , and let  $F' := F|_{A \mapsto 1}$  be the restricted formula obtained from  $F$  by setting every variable  $y \in A$  to 1. If  $F'$   $w$ -implies  $(x = 1)$  then we say  $x$  is *forced under  $\pi$*  and

---

```

14:   procedure ppsz( $F, w$ )
15:        $\pi :=$  a random permutation of the variables  $V$ 
16:       return ppsz( $F, w, \pi$ )

```

**Algorithm 2.** PPSZ with random permutation  $\pi$

---

write  $\text{Forced}(x, \pi) = 1$ ; otherwise we say  $x$  is *guessed under  $\pi$*  and write  $\text{Forced}(x, \pi) = 0$ . Let  $\text{Forced}(\pi) := \sum_{x \in V} \text{Forced}(x, \pi)$ .

Of course,  $\text{Forced}(x, \pi)$  depends on  $F$  and  $w$ , as well, so to be formally correct, we should write  $\text{Forced}(F, w, x, \pi)$ . Since  $F$  and  $w$  are fixed throughout, we prefer the less formal notation  $F(x, \pi)$ .

**OBSERVATION 1.2 ([8]).** *Suppose we run PPSZ with a fixed permutation  $\pi$ . Then  $\text{ppsz}(F, w, \pi)$  succeeds, i.e., finds  $\alpha$ , with probability exactly  $2^{-n+\text{Forced}(\pi)}$ .*

Taking  $\pi$  to be a random permutation we get

$$\begin{aligned} \Pr[\text{ppsz}(F) \text{ succeeds}] &= \mathbb{E}_{\pi} \left[ 2^{-n+\text{Forced}(\pi)} \right] \\ &\geq 2^{-n+\mathbb{E}_{\pi}[\text{Forced}(\pi)]}, \end{aligned} \tag{1}$$

which follows from Jensen's inequality applied to the convex function  $t \mapsto 2^t$ . We are now in a much more comfortable position:  $\mathbb{E}[\text{Forced}(\pi)] = \sum_x \Pr[\text{Forced}(x, \pi) = 1]$ , and we can analyze this probability for every variable individually. Indeed, this is what Paturi, Pudlák, Saks, and Zane [8] did: they showed that  $\Pr[\text{Forced}(x, \pi) = 1] \geq s_k - o(1)$  if  $F$  is a  $k$ -CNF formula with exactly one satisfying assignment. Here  $s_k$  is a number defined by the following experiment: let  $T_{k-1}^{\infty}$  be the complete rooted  $(k-1)$ -ary tree; pick  $\pi : V(T_{k-1}^{\infty}) \rightarrow [0, 1]$  at random and delete every node  $u$  with  $\pi(u) < \pi(\text{root})$ , together with all its descendants. Let  $\mathbf{T}$  be the resulting tree. Then

$$s_k := \Pr[\mathbf{T} \text{ is finite}] . \tag{2}$$

The  $o(1)$ -term converges to 0 as  $w$  tends to infinity; thus, the growth rate of  $w$  only influences how fast this  $o(1)$  error term vanishes, but (as far as we know) does not materially influence the success probability of PPSZ. We conclude:

**THEOREM 1.3 ([8]).** *If  $F$  is a  $k$ -CNF formula with a unique satisfying assignment, then*

$$\Pr[\text{ppsz}(F) \text{ succeeds}] \geq 2^{-n+s_k n - o(n)}.$$

Furthermore,  $s_k = \frac{\pi^2}{6k} + o(1/k)$ .

## 1.2 Previous improvements

**Multiple satisfying assignments.** The analysis of Paturi, Pudlák, Saks, and Zane runs into trouble if  $F$  contains multiple satisfying assignments. In their original paper [8] they presented a workaround; unfortunately, this is quite technical and, for  $k = 3, 4$ , exponentially worse than the bound of Theorem 1.3. It was a breakthrough when Hertli [4] gave a very general analysis of PPSZ showing that the “Unique-SAT bound” also holds in the presence of multiple satisfying assignments. Curiously, his proof takes the result “ $\Pr[\text{Forced}(x, \pi)] = s_k - o(1)$ ” more or less as a black box and does not ask how such a statement would have been obtained. Steinberger and myself [14] later simplified Hertli’s proof and obtained a certain unique-to-general lifting theorem that is also important for this work:

**THEOREM 1.4 (Unique-to-General lifting theorem [8]).** *If the success probability of PPSZ is at least  $2^{-n+s_k n + \epsilon n}$  on  $k$ -CNF formulas with a unique satisfying assignment, for some  $\epsilon > 0$ , then it is at least  $2^{-n+s_k n + \epsilon' n}$  on  $k$ -CNF formulas with multiple solutions, too, for some (smaller)  $\epsilon' > 0$ .*

**Improved algorithms.** Concerning the Unique-SAT case, Hertli [5] designed an algorithm that is a variant of PPSZ and achieves a success probability of  $2^{-n+s_3 n + \epsilon n}$  for 3-CNF formulas with a unique satisfying assignment. Unfortunately, the concrete value of  $\epsilon$  is so tiny that Hertli did not even bother to determine it, and his approach is extremely specific to 3-SAT, with no clear path how to generalize it to  $k$ -SAT. A result by Qin and Watanabe [11] strengthened Hertli’s result somewhat. More recently, Hansen, Kaplan, Zamir, and Zwick [3] published an algorithm called *biased-PPSZ*, a version of PPSZ in which some guessed variables are decided by a biased coin; which variables and how biased, that depends on the structure of the underlying formula. In contrast to Hertli’s, their improvement is “visible”: for 3-SAT, it improves the success probability from  $1.3070319^{-n}$  in Theorem 1.3 to  $1.306995^{-n}$ . Also, it works for all  $k$  (although the authors do not work out the exact magnitude of the improvement).

**Lower bounds.** Chen, Tang, Talebanfard, and myself [16] have shown that there are instances on which PPSZ has exponentially small success probability. Just *how exponentially small* has been tightened by Pudlák, Talebanfard, and myself [10]: we now know that PPSZ has success probability at most  $2^{-(1-2/k - o(1/k)) \cdot n}$  on certain instances, provided our parameter  $w$  is not too large; for *strong PPSZ*, i.e., PPSZ using small-width resolution instead of  $w$ -implication, then the same bound holds, provided your width bound is really small, like  $c \cdot \sqrt{\log \log n}$  [15].

### 1.3 Our contribution

We show that the success probability of PPSZ on  $k$ -CNF formulas is exponentially larger than  $2^{-n+s_k n}$ . In particular,

**THEOREM 1.5 (Improvement for all  $k$ ).** *For every  $k \geq 3$  there is  $\epsilon_k > 0$  such that the success probability of PPSZ on satisfiable  $k$ -CNF formulas is at least  $2^{-n(1-s_k-\epsilon_k)}$ .*

**Comparison to Hansen et al.** As already mentioned, a paper by Hansen, Kaplan, Zamir, and Zwick [3] introduces the algorithm biased-PPSZ, which also exhibits an exponentially improved running time. Which approach is better, ours or theirs? Numerically, neither paper bothers to analyze the asymptotic behavior of the improvement  $\epsilon_k$  as  $k$  grows. Conceptually, one might argue that our result subsumes theirs, because they actually have to define a new algorithm, while we simply give a better analysis of the old one. Methodologically, the two approaches are somewhat orthogonal: Hansen et al. choose the Boolean values of the variables in a non-uniform way but leave the permutation of the variables uniform; we change the permutation (more appropriately, we “pretend” to change it, because we don’t change the algorithm) but choose the Boolean values uniformly. The fact that we do not change the algorithm seems like a limitation but actually gives us greater freedom: we can exploit information gleaned from the formula, even if that information is by itself NP-hard to compute. I suspect that one can combine two approaches and get improved numbers for small  $k$ , like  $k = 3$ ; however, I fear that doing so would be extremely tedious and barely offer any additional insight.

Personally, I think it would be more fruitful to focus on both approaches individually and explore how far each can be pushed because they, in the words of [3], “only scratch the surface”. Unfortunately, both currently suffer from the same shortcoming: they do not improve the asymptotic  $\frac{\pi^2}{6}$ -factor in the behavior of the savings  $s_k$  for large  $k$ . To be more precise, the improvement  $\epsilon_k$  shrinks like  $o\left(\frac{1}{k}\right)$  for both Hansen et al. and this paper and thus becomes negligible compared to  $s_k = \frac{\pi^2}{6} \cdot \frac{1}{k} + o\left(\frac{1}{k}\right)$ .

### 1.4 The case $k = 3$

The case  $k = 3$  is the most visible and exhibits the fiercest competition. The full version of Hansen et al. and the ECC version of this result [13] invest considerable energy to hammer out a concrete numerical result how much they can improve over  $s_3$ . And although the  $k = 3$  part of [13] follows roughly the same approach as the general- $k$  case in this paper, it introduces several new concepts and methods that are not needed for Theorem 1.5. Furthermore, it is highly technical, and the set of people interested in it is most likely a clear subset of those interested in the general- $k$  case. Finally, the analysis for  $k = 3$  in [13] does not hit any natural wall, and therefore a simple tightening of inequalities and a better choice of constants and functions

would already yield a better bound. We therefore decided not to include the  $k = 3$  part in this paper.

## 1.5 Organization of the paper

We outline our general idea, analyzing PPSZ under some non-uniform distribution on permutations, in Section 2. Section 3 introduces the notions of critical clause trees and “cuts” in those trees. This is mainly a review of critical clause trees as defined in [8]; however, since we will manipulate these trees extensively, we introduce a more abstract and robust version, called “labeled trees” and cuts therein. Section 4 contains the proof of Theorem 1.5, our main result. We strive for succinctness above all else and took no effort to optimize the magnitude of our improvement.

## 2. Brief overview of our method

### 2.1 Working with a make-belief distribution on permutations

Our starting point is to take a closer look at the application of Jensen’s inequality:

$$\mathbb{E}_{\pi} \left[ 2^{-n+\text{Forced}(\pi)} \right] \geq 2^{-n+\mathbb{E}_{\pi}[\text{Forced}(\pi)]} .$$

This would be tight if  $X := \text{Forced}(\pi)$  was the same for every permutation  $\pi$ . But maybe certain permutations are “better” than others. The idea is to define a new distribution  $D$  on permutations, different from the uniform distribution, under which “good” permutations have larger probability, thus  $\mathbb{E}_{\pi \sim D}[X] > \mathbb{E}_{\pi \sim U}[X]$ . Sadly, we have no control over the distribution of permutations: firstly, we promised not to change the algorithm; secondly, and more importantly, defining  $D$  will require some information that is itself NP-hard to come by. There is a little trick dealing with this. Generally speaking, if we want to bound the expression  $\mathbb{E}_Q[2^X]$  from below but the obvious bound from Jensen’s inequality,  $2^{\mathbb{E}_Q[X]}$ , is not good enough for our purposes, we can replace  $Q$  by our favorite  $P$  but have to pay a price. Formally:

$$\begin{aligned} \mathbb{E}_Q[2^X] &= \sum_{\omega \in \Omega} Q(\omega) 2^{X(\omega)} = \sum_{\omega \in \Omega} P(\omega) \cdot \frac{Q(\omega)}{P(\omega)} 2^{X(\omega)} \\ &= \mathbb{E}_{\omega \sim P} \left[ 2^{X(\omega) - \log_2 \frac{P(\omega)}{Q(\omega)}} \right] \\ &\geq 2^{\mathbb{E}_P[X] - \mathbb{E}_{\omega \sim P} \left[ \log_2 \frac{P(\omega)}{Q(\omega)} \right]} \\ &= 2^{\mathbb{E}_P[X] - \text{KL}(P||Q)} . \end{aligned} \tag{3}$$

The term  $\text{KL}(P||Q) := \sum_{\omega} P(\omega) \log_2 \left( \frac{P(\omega)}{Q(\omega)} \right)$  is known as the Kullback-Leibler divergence from  $Q$  to  $P$ . If  $Q$  and  $P$  are continuous distributions (over  $\Omega = [0, 1]^n$ , for example) with density functions  $f_Q$  and  $f_P$ , then (3) still holds, for  $\text{KL}(P||Q) := \int_{\Omega} f_P(\omega) \log_2 \left( \frac{f_P(\omega)}{f_Q(\omega)} \right)$ . This trick is not

new: it plays a crucial role in [14], and, if you look close enough, also in Hertli [4]; it appears, in simpler form, already in [8]. However, in [14, 4, 8], the distribution  $P$  is defined only to make “liquid variables” (variables  $x$  for which  $F|_{x=0}$  and  $F|_{x=1}$  are both satisfiable) come earlier in  $\pi$  and do not take the syntactic structure of  $F$  into account: they define  $P$  purely in terms of  $\text{sat}(F)$ , the space of solutions, whereas our  $P$  will depend heavily on the structure on  $F$  as a  $k$ -CNF formula. Our work is the first to apply this method to improving PPSZ on formulas with a unique satisfying assignment.

## 2.2 Good make-belief distributions for PPSZ—a rough sketch

How can we apply this idea to the analysis of PPSZ? The challenge is to find a distribution  $D$  under which  $\mathbb{E}_{\pi \sim D}[\text{Forced}(\pi)]$  is larger than under the uniform distribution. Since we assume that  $F$  has the unique satisfying assignment  $\alpha = (1, \dots, 1)$ , we can find, for every variable  $x$ , a *critical clause* of the form  $(x \vee \bar{y} \vee \bar{z})$ .<sup>1</sup> Critical clauses play a crucial role in [8] and [3] as well. Imagine we change the distribution on permutations such that  $y$  tends to come a bit earlier than under the uniform distribution. It is easy to see that this can only decrease  $\mathbb{E}[\text{Forced}(y, \pi)]$  (which is bad) and only increase  $\mathbb{E}[\text{Forced}(a, \pi)]$  for all other variables. In particular, it usually increases  $\mathbb{E}[\text{Forced}(x, \pi)]$  (which is good). Now assume the literal  $\bar{y}$  appears in a disproportionately large number of critical clauses. Then the beneficial effect of pulling  $y$  to the front of  $\pi$  outweighs its adverse effect. Thus, if there is a set  $V' \subseteq V$  of variables with  $|V'| = \Omega(n)$ , and each  $y \in V'$  appears in a large number of critical clauses, we can define a new distribution  $P$  on permutations  $\pi$  under which variables  $V'$  tend to come earlier than under the uniform distribution. Using  $P$  as our make-belief distribution in (3), we obtain a success probability that is exponentially larger than the baseline. This is what we call the “highly irregular case” below.

The other extreme would be the “almost regular case”, namely that almost every variable  $x$  has exactly one critical clause and that almost every negative literal  $\bar{y}$  appears in exactly two critical clauses. In this case, we find a matching  $M$ , i.e., a set of disjoint pairs of variables such that  $\{y, z\} \in M$  implies that  $(x \vee \bar{y} \vee \bar{z})$  is a critical clause of  $F$ , for some variable  $x$ . We then adapt the distribution on permutations such that the location of  $y$  and  $z$  is positively correlated—either they both tend to come late or they both tend to come early. This will have both (easily quantifiable) beneficial effects and (more difficult to quantify) adverse effects. However, we will see that the adverse effects can only be large if  $\mathbb{E}_{\pi \sim U}[\text{Forced}(\pi)]$  is already larger than  $s_k n$  under the uniform distribution.

---

1 Our informal outline assumes  $k = 3$  to keep notation simple.

### 3. Critical clause trees, labeled trees, and cuts

This section introduces the key notions of critical clause trees and cuts, which were already defined in [8]. We introduce a more general notion that we call *labeled trees* and of *cuts* therein. We will perform extensive “tree surgery”, and this new terminology will allow us to state our results in a concise, rigorous, and readable fashion.

#### 3.1 The Critical Clause Tree

All notions and results in this subsection already appear in [8], although we might phrase certain things a little different. We assume that  $\alpha = (1, \dots, 1)$  is the unique satisfying assignment of our input  $k$ -CNF formula  $F$ . That means that for every variable  $x$ , we can find a clause of the form  $(x \vee \bar{y}_2 \vee \dots \vee \bar{y}_k)$ . This is called a *critical clause of  $x$* . For an integer  $h \in \mathbb{N}$ , a *critical clause tree of  $x$  of height  $h$*  is a rooted tree  $T_x$  of height at most  $h$  with a bunch of additional information: every node  $u$  of  $T_x$  has a *variable label*  $\text{varlabel}(u)$ ; if the depth of  $u$  is less than  $h$ , it has a *clause label*  $\text{clauselabel}(u)$ . The tree is constructed as follows:

- Initialize  $T_x$  as consisting of a single root node, and set  $\text{varlabel}(\text{root}) = x$ .
- While some node  $u$  of  $T_x$  of depth less than  $h$  does not have a clause label yet:
  1. Let  $\alpha_u$  be the assignment arising from  $\alpha$  by setting to 0 all the variables  $y$  that appear as variable labels on the path from the root to  $u$  (including both the label of the root, which is  $x$ , and the label of  $u$ ). Let  $a := \text{varlabel}(u)$ . In particular,  $\alpha_u(a) = 0$ .
  2. Pick a clause  $C$  that is violated by  $\alpha_u$  (this exists since  $\alpha$  is the unique satisfying assignment), and set  $\text{clauselabel}(u) := C$ .
  3. For each negative literal  $\bar{z} \in C$ , create a new child of  $u$  and give it variable label  $z$ . Note that  $u$  has at most  $k - 1$  children.

The construction depends on the choice of  $h$ , so we should write  $T_x^{(h)}$  instead of  $T_x$ ; however, the number  $h$  will be fixed throughout, so we simply write  $T_x$  for brevity.<sup>2</sup> This tree is central to the analysis in [8] and also [3] (but curiously is completely absent in [4] and [14]). The *depth* of a node  $u$  in a tree  $T$  is the length of the path from the root to  $u$ ; we abbreviate it as  $d_T(u)$  or simply  $d(u)$  if  $T$  is understood.  $T_x$  is a  $(k - 1)$ -ary tree: every node has at most  $k - 1$  children, as Step 3 creates a child for every *negative* literal in  $C$ ; since  $\alpha = (1, \dots, 1)$  satisfies  $C$ , there are at most  $k - 1$  negative literals in  $C$ .

**OBSERVATION 3.1 ([8]).** Suppose  $C = (y_1 \vee \dots \vee y_i \vee \bar{z}_1 \vee \dots \vee \bar{z}_j)$  is the clause label of a node  $u$ . Then

---

2 Indeed, strictly speaking,  $T_x$  also depends on the formula  $F$  and the satisfying assignment  $\alpha$ , so one should write  $T_x^{(F, \alpha, h)}$ . However, since  $F$ ,  $\alpha$ , and  $h$  always refer to the same formula, assignment, and integer parameter, there is no gain in explicitly listing this dependency.

1. each variable among  $y_1, \dots, y_i$  appears as the clause label of some (not necessarily proper) ancestor of  $u$ ;
2.  $u$  has  $j$  children (note that  $j = 0$  might happen) whose variable labels are  $z_1, \dots, z_j$ .
3. If  $v$  is a proper descendant of  $u$  in  $T_x$  then  $\text{varlabel}(u) \neq \text{varlabel}(v)$ .

**PROOF.** Point 1 and 2 follow immediately from the construction process. To show Point 3, let  $a := \text{varlabel}(u)$  and  $b := \text{varlabel}(v)$ . We have to show that  $a \neq b$ . Let  $w$  be the parent of  $v$ , so  $w$  is a (not necessarily proper) descendant of  $u$ , and let  $C = \text{clauselabel}(w)$ . By construction, in particular Point 3 above,  $C$  contains the negative literal  $\bar{b}$ . Since  $\alpha_w$  violates  $C$ , by choice of  $C$  in Point 2, we conclude that  $\alpha_w(b) = 1$ . By definition of  $\alpha_w$  in Point 1, we observe that  $\alpha_w(a) = 0$  and therefore  $a \neq b$ . ■

A *down-path* in a rooted tree is a sequence  $u_0, \dots, u_t$  where each  $u_i$  is the parent of  $u_{i+1}$ . A *root-path* is a path starting at the root. Note that a root path *always* is a down-path. By Observation 3.1, no variable can appear twice or more on a down-path in  $T_x$ . Critical clause trees are important because of the following lemma:

**LEMMA 3.2 ([8]).** Suppose  $w \geq (k-1)^{h+1}$ , where  $w$  is the strength parameter of PPSZ as in Algorithm 13. Let  $x \in V$ ,  $\pi$  a permutation of  $V$ , and denote by  $A$  the set of variables coming before  $x$  in  $\pi$ . If every path from the root of  $T_x$  to a leaf at depth  $h$  contains a node  $u$  with  $\text{varlabel}(u) \in A$  then  $\text{Forced}(x, \pi) = 1$ .

**PROOF.** Call a node  $u$  of  $T_x$  *dead* if  $\text{varlabel}(u)$  comes strictly before  $x$  in  $\pi$ ; call a node  $v$  *reachable* if the path from the root to  $v$  contains no dead nodes. Note that the root itself is reachable. By the assumption in the lemma, no leaf at depth  $h$  is reachable, and therefore every reachable node  $v$  has a clause label. Let  $G$  be the set of clause labels of all reachable nodes. Note that  $|G| \leq 1 + (k-1) + (k-1)^2 + \dots + (k-1)^h \leq \frac{(k-1)^{h+1} - 1}{k-2} \leq w$ .

When the loop of  $\text{PPSZ}(F, \pi, w)$  arrives  $x$ , the formula  $F$  has already been reduced to  $F' := F|_{A=1}$ , i.e., all variables in  $A$  have been set to 1. We have to show that  $F'$  implies  $x = 1$ . Indeed, we show something stronger, namely that  $G' := G|_{A=1}$  implies  $x = 1$ .

To show that  $G'$  implies  $(x = 1)$ , let  $\gamma'$  be a total assignment to the remaining variables  $V \setminus A$  with  $\gamma'(x) = 0$ . We have to show that  $\gamma'$  violates  $G'$ . Equivalently, we have to show that every total assignment  $\gamma$  on  $V$  that sets  $\gamma(x) = 0$  and  $\gamma(a) = 1$  for all  $a \in A$  violates  $G$ . For such a  $\gamma$ , find a maximal root path  $u_0, u_1, \dots, u_t$  of nodes in  $T_x$  such that  $\gamma(\text{varlabel}(u_i)) = 0$  for all  $u_i$  on that path. Such a path is non-empty because  $\gamma(\text{varlabel}(u_0)) = \gamma(\text{varlabel}(\text{root})) = \gamma(x) = 0$  by assumption. Can the last node  $u_t$  be a leaf of height  $h$ ? Obviously not: by assumption, such a path from root to leaf would contain node, let us say  $u_i$ , with  $a := \text{varlabel}(u_i)$  coming before  $x$ , and thus  $a \in A$  and  $\gamma(a) = 1$ . We conclude that  $u_t$  is not a leaf of height  $h$  and therefore has a clause label  $C$ . We write  $C = (y_1 \vee \dots \vee y_i \vee \bar{z}_1 \vee \dots \vee \bar{z}_j)$  with  $i + j = k$ . Note that  $i \geq 1$

since  $\alpha = (1, \dots, 1)$  satisfies  $C$ , but  $j = 0$  is possible. By Point 1 of Observation 3.1, each of the variables  $y_1, \dots, y_i$  is the variable label of one of the  $u_0, \dots, u_t$ , and thus  $\gamma(y_1) = \dots = \gamma(y_i) = 0$ ; by Point 2 of Observation 3.1, each of the  $z_1, \dots, z_j$  is the variable label of a child  $v$  of  $u_t$ . By maximality of the path  $u_0, \dots, u_t$ , we have  $\gamma(\text{varlabel}(v)) = 1$ , and thus  $\gamma(z_1) = \dots = \gamma(z_j) = 1$ . In other words,  $\gamma$  violates  $C$  and thus  $G$ . ■

From now on, we take  $h = h(n)$  to be the largest integer such that  $w \geq (k-1)^{h+1}$ . Note that  $\lim_{n \rightarrow \infty} h(n) = \infty$  because  $\lim_{n \rightarrow \infty} w(n) = \infty$ .

### 3.2 The canonical critical clause tree

In Point 2 of the construction process for critical clause trees, we might have several violated clauses to choose from. For example, if  $x$  has more than one critical clause, then there are several different critical clause trees  $T_x$ . In this section, we make things unique by introducing the concept of *canonical clauses* and *canonical clause trees*.

**DEFINITION 3.3 (Canonical critical clause).** Among all critical clauses of  $x$ , we choose one and call it the *canonical critical clause of  $x$* .

This choice is arbitrary but considered fixed from now on. Every variable has exactly one canonical critical clause.

**DEFINITION 3.4 (Canonical critical clause tree, CCCT).** The *canonical critical clause tree* of a variable  $x$  of height  $h$  is the critical clause tree  $T_x$  that is constructed as above, but additionally adhering to the following tie-breaking rule in Point 2: if the canonical critical clause of  $\text{varlabel}(u)$  is violated by  $\alpha_u$ , pick it as  $\text{clauselabel}(u)$ ; otherwise, pick the lexicographically first violated clause.

In particular, if  $T_x$  is the CCCT of  $x$ , then the clause label of its root is the canonical critical clause of  $x$ . We distinguish between canonical and non-canonical nodes:

**DEFINITION 3.5 (Canonical nodes).** In the canonical critical clause tree  $T_x$ , we call a node  $v$  *canonical* if, for every node  $u$  on the path from the root to  $v$  (including root and  $v$ ),  $\text{varlabel}(u)$  has exactly one critical clause, and this clause is  $\text{clauselabel}(u)$ .

Suppose  $u$  is a maximal non-canonical node in  $T_x$  (i.e., either  $u$  is the root itself or the parent of  $u$  is canonical), and write  $a = \text{varlabel}(u)$ . There are two reasons why  $u$  can be non-canonical: (1) the variable  $a$  has two or more critical clauses; (2) the clause label  $C := \text{clauselabel}(u)$  is not a critical clause (it has at least two positive literals, like  $(x \vee a \vee \bar{b})$ , for example). In the latter case,  $u$  has at most  $k-2$  children in  $T_x$ .

**Remark.** For the improvement for general  $k$  as presented in this paper, the notions of canonical clause trees and canonical nodes therein are not strictly necessary, and we could prove some improvement without introducing it. However, since I feel that this is the correct way of thinking about critical clause trees and since the  $k = 3$  case makes heavy use of these notions, I decided to include it in this paper, for future reference.

### 3.3 Labeled trees and cuts

We will perform certain manipulations on critical clause trees. The resulting trees will not be critical clause trees in any meaningful sense any more (for example, they sometimes become infinite). Therefore, we introduce the more general notion of a *labeled tree*. We note that certain labeled trees appear implicitly in [8]; however, our work is the first that explicitly defines and uses them.

**DEFINITION 3.6 (Labeled trees).** We assume some countably infinite set  $L$  of labels with  $V \subseteq L$ . A *labeled tree* is a rooted tree  $T$ , possibly infinite, in which

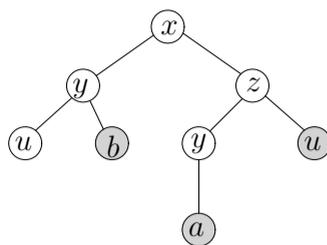
1. each node  $u$  has a label  $\text{varlabel}(u) \in L$ ;
2. no label appears twice on a down-path; that is, if  $u$  is a proper ancestor of  $v$  in  $T$ , then  $\text{varlabel}(u) \neq \text{varlabel}(v)$ ;
3. each node is marked either as *canonical* or *non-canonical*; if  $u$  is non-canonical then so are all of its children (and by induction all of its descendants);
4. each leaf of  $T$  is marked as either a *safe leaf* or an *unsafe leaf*.

A *safe path* in  $T$  is a path starting at the root that either ends at a safe leaf or is infinite. We write  $\text{Can}(T_x)$  to denote the set of canonical nodes in  $T_x$ . Furthermore, all labeled trees appearing in this paper are  $(k - 1)$ -ary: each node has at most  $k - 1$  children.

A critical clause tree of height  $h$  becomes a labeled tree by simply marking leaves at depth  $h$  as *safe leaves* and all other leaves (of smaller depth) as *unsafe leaves* and removing all clause labels.

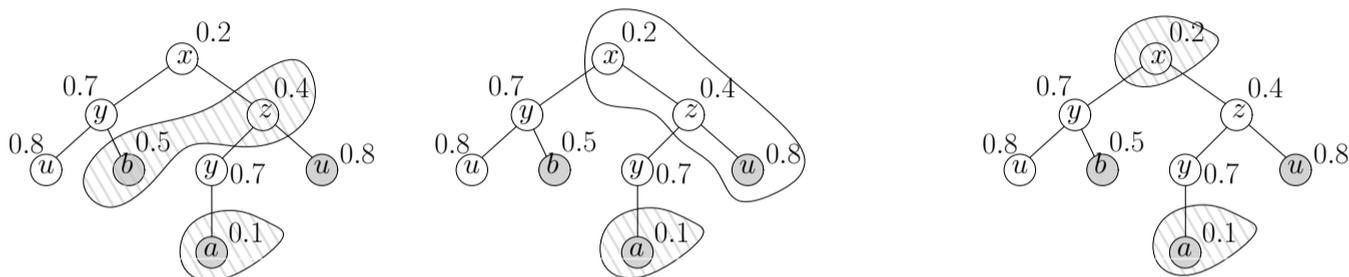
From now on, instead of viewing  $\pi$  as a permutation of the variables  $V$ , we view it as a *placement*  $\pi : L \rightarrow [0, 1]$  on the countably infinite set of labels. If  $\pi$  is sampled from some continuous distribution (for example the uniform distribution), then  $\pi$  is injective with probability 1; its restriction to  $V$  defines a permutation, by sorting the variables from low- $\pi$  to high- $\pi$ .

**DEFINITION 3.7 (Cut and  $\text{Cut}_r$  and  $\text{wCut}_r$ ).** Let  $T$  be a labeled tree,  $x$  the label of its root, and  $r \in [0, 1]$ . The event  $\text{Cut}_r(T)$  is an event in the probability space of all placements, defined as follows: mark a non-root vertex  $u$  as *dead* if  $\pi(\text{varlabel}(u)) < r$  and *alive* otherwise; mark root as alive. Then  $\text{Cut}_r(T)$  is the event that every safe path in  $T$  contains at least one dead node.  $\text{Cut}(T)$  is the event  $\text{Cut}_{\pi(x)}(T)$ , i.e., all nodes  $u$  with  $\pi(\text{varlabel}(u)) < \pi(x)$  are marked dead.



A labeled tree. The leaves with shaded circles are the safe leaves. Note that  $u$  is the label of a safe leaf and an unsafe leaf. Also, there are safe leaves at different depths. The latter situation will not arise in critical clause trees, but they are allowed under our definition of labeled trees.

Below, we choose a placement  $\pi : L \rightarrow [0, 1]$ :



$\text{Cut}_{0.6}(T)$  happens:  $b$  and  $z$  are dead and block all downpaths to a safe leaf.

$\text{Cut}_{0.3}(T)$  does not happen: there is an alive downpath to a safe leaf

$\text{wCut}_{0.3}(T)$  does happen: the root is already dead

**Figure 1.** A labeled tree, a placement, and a cut, a non-cut and a weak cut.

We define  $\text{wCut}_r(T)$  (“weak cut”) exactly as  $\text{Cut}_r(T)$ , only that we additionally mark the root as dead if  $\pi(x) < r$ .

Note that there is no corresponding event  $\text{wCut}(T)$ . Weak cuts only make sense with respect to a particular  $r \in [0, 1]$ . The following observation is simply Lemma 3.2, framed in the new terminology:

**OBSERVATION 3.8.** Suppose  $w \geq (k - 1)^{h+1}$ . If  $\text{Cut}(T_x)$  happens then  $\text{Forced}(x, \pi) = 1$ .

To better understand the statement of the observation, recall that the definition of the critical clause tree  $T_x$  depends on  $h$ , and the definition of  $\text{Forced}$  depends on  $w$ . The two definitions  $\text{Cut}_r$  and  $\text{wCut}_r$  are intimately related:

**OBSERVATION 3.9.** First, it holds that

$$\text{wCut}_r(T) = [\pi(\text{root}) < r \vee \text{Cut}_r(T)]$$

Furthermore, if  $T_1, \dots, T_l$  are the subtrees of  $T$  rooted at the children of the root, then

$$\text{Cut}_r(T) = [\text{wCut}_r(T_1) \wedge \dots \wedge \text{wCut}_r(T_l)]$$

A particularly important example of a labeled tree is  $T_{k-1}^\infty$ . This is simply an infinite complete  $(k-1)$ -ary tree: every node has  $k-1$  children, and there are no leaves. All nodes have distinct labels. If  $k$  is understood, we simply write  $T^\infty$ . If  $\pi$  is a uniformly random placement then

$$\begin{aligned} Q_r^{(k)} &:= \Pr[\text{Cut}_r(T_{k-1}^\infty)] = (\Pr[\text{wCut}_r(T_{k-1}^\infty)])^{k-1} \\ P_r^{(k)} &:= \Pr[\text{wCut}_r(T_{k-1}^\infty)] = r \vee \Pr[\text{Cut}_r(T_{k-1}^\infty)], \end{aligned}$$

where we define  $a \vee b := a + b - ab$  for  $a, b \in [0, 1]$ . It follows from the two equations above that  $Q_r^{(k)}$  and  $P_r^{(k)}$  satisfy the system of equations

$$\begin{aligned} Q &= P^{k-1} \\ P &= r \vee Q. \end{aligned}$$

By substitution, it follows that  $Q_r^{(k)}$  and  $P_r^{(k)}$  individually satisfy the following equalities, respectively:

$$\begin{aligned} Q &= (r + (1-r)Q)^{k-1} \\ P &= r \vee P^{k-1}. \end{aligned}$$

Note that  $P = 1$  and  $Q = 1$  are always solutions of these equations, but sometimes it is not the “correct” solution. Indeed, it is a well-known result from the theory of Galton-Watson branching processes that  $Q_r^{(k)}$  and  $P_r^{(k)}$  are the *smallest* solutions of these equations.

**PROPOSITION 3.10.** *For  $r \geq \frac{k-2}{k-1}$  it holds that  $Q_r^{(k)} = P_r^{(k)} = 1$ . On the interval  $[0, \frac{k-2}{k-1}]$ ,  $P_r^{(k)}$  is convex and  $r \leq P_r^{(k)} \leq \frac{k-1}{k-2} \cdot r$ . Also on that interval,  $Q_r^{(k)} \leq \left(\frac{k-1}{k-2} \cdot r\right)^{k-1} \leq e r^{k-1}$ .*

For  $k = 3$  we have explicit expression:

$$Q_r^{(3)} = \begin{cases} \left(\frac{r}{1-r}\right)^2 & \text{if } r < 1/2 \\ 1 & \text{if } r \geq 1/2 \end{cases} \quad (4)$$

and

$$P_r^{(3)} = \begin{cases} \frac{r}{1-r} & \text{if } r < 1/2 \\ 1 & \text{if } r \geq 1/2 \end{cases} \quad (5)$$

Again, if  $k$  is understood, we will simply write  $Q_r$  and  $P_r$ . Recall the number  $s_k$  as defined in (2) and observe that

$$s_k = \Pr[\text{Cut}(T_\infty^{(k-1)})]. \quad (6)$$

Paturi, Pudlák, Saks, and Zane proved the following fact:

**LEMMA 3.11 ([8]).** Let  $T_x$  be a critical clause tree of height  $h$ . Then  $\Pr[\text{Cut}_r(T_x)] \geq Q_r^{(k)} - \text{Error}(r, h)$  and  $\Pr[\text{Cut}(T_x) \geq s_k - \text{Error}(h)]$ . Here,  $\text{Error}(r, h)$  and  $\text{Error}(h)$  are functions that converge to 0 as  $h \rightarrow \infty$ .

We will give a full and overly formal proof of this lemma. It will be instructive to go through the proof in full detail because it is a simplest application of our labeled-tree-machinery, on which we will rely more heavily later on. Also, our proof is slightly different from the original proof in [8].

**PROOF OF LEMMA 3.11.** Let  $T_x$  be the critical clause tree of height  $h$  for variable  $x$ . We subject it to a sequence of transformation steps, each of which only reduces the probability  $\Pr[\text{Cut}_r]$  and makes the tree look more and more like  $T_\infty^{(k-1)}$ .

**Step 1. Completing the tree.** Add nodes to  $T_x$  to make it a complete  $(k-1)$ -ary tree of height  $h$  (i.e., while some  $v$  of depth less than  $h$  has fewer than  $k-1$  children, create a new child). Give a fresh label from  $L$  to each newly created node (by *fresh label* we mean a label in our infinite label space  $L$  that has never been used before) and mark all leaf as safe leaves (note that all leaves have height  $h$  now). Call the resulting tree  $T_0$ . Every safe path in  $T_x$  is still a safe path in  $T_0$ , and therefore  $\Pr[\text{Cut}_r(T_0)] \leq \Pr[\text{Cut}_r(T_x)]$ . Note that  $T_0$  might not a critical clause tree anymore (some non-leaves nodes do not have clause labels, some variable labels are not variables at all but fresh labels).

**Step 2. Making labels distinct.** In a sequence of steps, we want to make sure that no label appears twice in the tree. To achieve this, we create a sequence  $T_0, T_1, T_2, \dots$  as follows:

**PROPOSITION 3.12.** Let  $T$  be a labeled tree and  $v$  a node therein. Let  $a'$  be a fresh label, i.e., one that does not appear in  $T$ . Define  $T_{v \rightarrow a'}$  to be the same as  $T$  but with  $\text{varlabel}(v) := a'$  for some fresh label  $a' \notin L(T)$ . Then

$$\Pr[\text{Cut}_r(T_{v \rightarrow a'})] \leq \Pr[\text{Cut}_r(T_i)]$$

holds.

This is how we construct the sequence  $T_0, T_1, T_2, \dots$ : while some label (say  $a$ ) appears more than once in  $T_i$ , say at nodes  $u$  and  $v$ , take a fresh label  $a'$  and set  $T_{i+1} := T_{v \rightarrow a'}$ .

**Proof of Proposition 3.12.** This is the heart of the proof. Fix a partial assignment  $\tau : L \setminus \{a, a'\} \rightarrow [0, 1]$ . We will show that  $\Pr[\text{Cut}_r(T_{i+1}) \mid \tau] \leq \Pr[\text{Cut}_r(T_i) \mid \tau]$ . Here, the conditioning on  $\tau$  is a shorthand for conditioning on the event  $[\pi(l) = \tau(l) \forall l \in L \setminus \{a, a'\}]$ .

Note that both  $\text{Cut}_r(T_i)$  and  $\text{Cut}_r(T_{i+1})$  are monotone Boolean functions in the atomic events  $z_l := [\pi(l) < r]$  for  $l \in L$ . On the probability space conditioned on  $\tau$ , the event  $\text{Cut}_r(T_{i+1})$  becomes

a Boolean function  $f(z_a, z_{a'})$  in the Boolean variables  $z_a := [\pi(a) < r]$  and  $z_{a'} := [\pi(a') < r]$ . Since  $T_i$  can be obtained from  $T_{i+1}$  by replacing  $a'$  by  $a$ , we observe that conditioned on  $\tau$ , the event  $\text{Cut}_r(T_i)$  becomes the Boolean function  $f(z_a, z_a)$ . The proof of the claim now works by going through all possibilities what the monotone Boolean function  $f$  could be.

1. If  $f(z_a, z_{a'})$  is the constant 1 function, then  $\Pr[\text{Cut}_r(T_{i+1})|\tau] = 1 = \Pr[\text{Cut}_r(T_i)|\tau]$ ; similarly, if  $f \equiv 0$  then both probabilities are 0.
2. If  $f(y, z) \equiv y$  then  $[\text{Cut}_r(T_{i+1})|\tau] = [\pi(a) < r] = [\text{Cut}_r(T_i)|\tau]$ , so both events are the same.
3. If  $f(y, z) = z$  then  $[\text{Cut}_r(T_{i+1})|\tau] = [\pi(a') < r]$  and  $[\text{Cut}_r(T_i)|\tau] = [\pi(a) < r]$ ; the events are not the same, but both have probability  $r$ .
4. If  $f(y, z) = y \wedge z$  then

$$\Pr[\text{Cut}_r(T_{i+1})|\tau] = \Pr[\pi(a) < r \wedge \pi(a') < r] = r^2 \leq r = \Pr[\pi(a) < r] = \Pr[\text{Cut}_r(T_i)|\tau]$$

and the claimed inequality holds.

5. If  $f(y, z) = y \vee z$  then

$$\Pr[\text{Cut}_r(T_{i+1})|\tau] = \Pr[\pi(a) < r \vee \pi(a') < r] = 2r - r^2 > r = \Pr[\pi(a) < r] = \Pr[\text{Cut}_r(T_i)|\tau]$$

and the claimed inequality does *not* hold. But here is the thing: this cannot happen! Indeed, for  $[\text{Cut}_r(T_{i+1})|\tau]$  to become  $[\pi(a) < r \vee \pi(a') < r]$ , the two nodes  $u$  and  $v$  would have to be ancestors of each other, which by Observation 3.1 is impossible. To be more precise, suppose  $[\text{Cut}_r(T_{i+1})|\tau]$  is indeed  $[\pi(a) < r \vee \pi(a') < r]$ . Now set  $\pi(a) = \pi(a') = 1$  (and keep  $\pi(l) = \tau(l)$  for all other labels  $l$ ) so the event does not happen. By definition of  $\text{Cut}_r$ , this means that there is a safe path  $p$  in  $T_{i+1}$   $\pi(\text{varlabel}(w)) \geq r$  for all nodes  $w \in p$ . If we change  $\pi$  to  $\pi'$  by setting  $\pi'(a) = 0$  then  $[\pi'(a) < r]$  holds<sup>3</sup> and thus  $\text{Cut}_r$  does happen; this means that the variable label  $a$  appears on the path  $p$ . By an analogous argument, the label  $a'$  appears on the path  $p$ , too. In  $T_i$ , however, both those nodes have label  $a$ , which contradicts Point 3 of Observation 3.1.<sup>4</sup>

Note that the claimed inequality holds in all possible cases. This completes the proof of Proposition 3.12. ◆

The sequence  $T_0, T_1, T_2, \dots$  terminates because each application of Proposition 3.12 introduces a new label to the tree, and the number of labels will never exceed the number of nodes, which is  $1 + (k-1) + (k-1)^2 + \dots + (k-1)^h$ . Once it terminates, we obtain a tree  $T_s$  in which all labels are distinct and  $\Pr[\text{Cut}_r(T_x)] \geq \Pr[\text{Cut}_r(T_s)]$ . The event  $\text{Cut}_r(T_s)$  can easily be phrased

3 Here we silently assume  $0 < r$ . However, for the case  $r = 0$  it is clear that none of the atomic events  $[\pi(l) < r]$  happen and thus  $\text{Cut}_0(T_{i+1})$  and  $\text{Cut}_0(T_i)$  are either both the empty event or both the whole probability space and thus happen with equal probability.

4 Or rather it contradicts the definition of a labeled tree; the condition that no ancestors-descendant pair have the same labels is baked into Definition 3.6, and the reader should convince themselves that our transformations within this proof keep this property.

in terms of  $T_\infty$ : for an integer  $t$ , let  $C_t$  be the event (in the probability space of placements  $L \rightarrow [0, 1]$ ) that all paths  $p$  in  $T_\infty$  starting at the root and having length  $t$  contain a non-root node  $u$  with  $\pi(\text{varlabel}(u)) < r$ . Note that  $T_s$  is, in a sense, isomorphic to the first  $h + 1$  levels of  $T_\infty$ , and therefore  $\Pr[\text{Cut}_r(T_s)] = \Pr[C_h]$ .

For given  $\pi$ , let  $\mathbf{T}$  be the subtree of  $T_\infty$  obtained by deleting every non-root node  $u$  with  $\pi(\text{varlabel}(u)) < r$ , together with all ancestors. Note that  $\mathbf{T}$  is a random variable over our probability space of placements  $\pi$ . Thus,  $C_t$  is the event that  $\mathbf{T}$  has no root-path of length  $t$ , and  $\text{Cut}_r(T_\infty)$  is the event that  $\mathbf{T}$  has no infinite root-path. Note that in the latter case,  $\mathbf{T}$  is indeed finite and thus has a finite longest root-path, which in turn means

$$\text{Cut}(T_\infty) = \bigcup_{t \in \mathbb{N}} C_t .$$

The  $C_t$  form an increasing sequence of events and thus, by the Monotone Convergence Theorem, it holds that

$$\Pr[\text{Cut}_r(T_\infty)] = \Pr \left[ \bigcup_{t \in \mathbb{N}} C_t \right] = \lim_{t \rightarrow \infty} \Pr[C_t] . \quad (7)$$

Define  $\text{Error}(r, t) := \Pr[\text{Cut}_r(T_\infty)] - \Pr[C_t]$ . Then (7) states that  $\lim_{t \rightarrow \infty} \text{Error}(r, t) = 0$ . We conclude that

$$\begin{aligned} \Pr[\text{Cut}(T_x)] &\geq \Pr[\text{Cut}(T_0)] && \text{(by the transformation in Step 1)} \\ &\geq \Pr[\text{Cut}(T_s)] && \text{(by the transformations in Step 2)} \\ &= \Pr[C_h] \\ &= \Pr[\text{Cut}_r(T_\infty)] - \text{Error}(r, h) . \end{aligned}$$

This concludes the proof of the lemma. ■

The important part of the proof, in particular with regard to what comes below, is Proposition 3.12. In words, the fact that multiple labels can only make things better. Paturi, Pudlák, Saks, and Zane [8] use the FKG inequality to prove this. Our proof above does not use the FKG inequality. Indeed, from our proof technique one could extract a stand-alone proof of the FKG inequality (at least for the special case of monotone Boolean function and all variables being independent) that does not use top-down induction but works by replacing variables by fresh copies, step by step, and showing that each step satisfies the desired inequality.

## 4. An Exponential Improvement

### 4.1 The highly irregular case

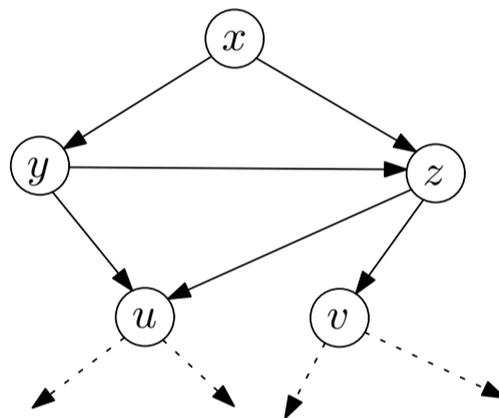
In our outline in Section 2.2 we mentioned that we will address two cases separately: the “highly irregular case” and the “almost regular case”. The highly irregular case means that there is

a set of  $\Omega(n)$  variables each of which appears as a negative literal in extraordinarily many critical clauses. This is the case that we treat in this section. Recall that every variable  $x$  has one *canonical critical clause* that looks like  $(x \vee \bar{y}_1 \vee \dots \vee \bar{y}_{k-1})$ .

**DEFINITION 4.1 (Critical Clause Graph).** The critical clause graph (CCG) is a directed graph on vertex set  $V$  (the  $n$  variables) with  $(k-1)n$  arcs defined as follows: for each  $x \in V$ , let  $(x \vee \bar{y}_1 \vee \dots \vee \bar{y}_{k-1})$  be its canonical critical clause and add the arcs  $(x, y_1), \dots, (x, y_{k-1})$  to the graph.

In the following figure, we show the beginning of a CNF formula with the canonical critical clauses underlined and part of its critical clause graph.

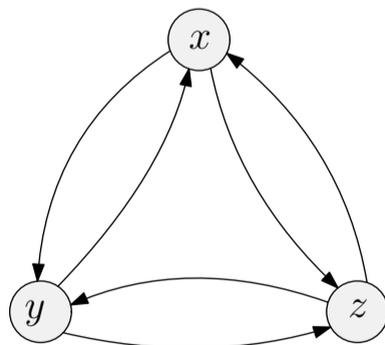
$$\underline{(x \vee \bar{y} \vee \bar{z})} \wedge (x \vee \bar{u} \vee \bar{v}) \wedge \underline{(y \vee \bar{u} \vee \bar{z})} \wedge \underline{(z \vee \bar{u} \vee \bar{v})} \wedge \dots$$



Let us give a second example. If we take the simplest 3-CNF with a unique satisfying assignment, namely

$$xyz \wedge xy\bar{z} \wedge x\bar{y}z \wedge \bar{x}yz \wedge \underline{x\bar{y}\bar{z}} \wedge \underline{\bar{x}y\bar{z}} \wedge \underline{\bar{x}\bar{y}z}$$

(where we replace  $\vee$  by juxtaposition for compactness), we get the following critical clause graph:



We see that the CCG might have anti-parallel edges  $(x, y), (y, x)$  but no self-loops. The latter follows since  $(x \vee \bar{x} \vee \dots)$  always evaluates to 1, and we assume no clause of  $F$  has this form. In the CCG, every vertex has out-degree  $k-1$ , thus it has a total of  $(k-1)n$  arcs, and the *average* in-degree is  $k-1$ , too.

**DEFINITION 4.2.** For  $x \in V$  let  $\text{indeg}(x)$  denote the in-degree of  $x$  in the CCG. For a set  $X \subseteq V$  we define

$$\text{indeg}(X) := \sum_{x \in X} \text{indeg}(x) .$$

An arc  $(x, y)$  with  $x, y$  both in  $X$  counts towards  $\text{indeg}(X)$  as well.

For  $l \in \mathbb{N}$ , we call a variable  $x$  *l-heavy* if  $\text{indeg}(x) \geq l$ . Let  $\text{HEAVY}(l)$  be the set of all *l-heavy* variables. If  $l$  is understood from the context, we might simply write  $\text{HEAVY}$ .

**THEOREM 4.3.** For every  $k \in \mathbb{N}$ , there is some  $l = l(k) \in \mathbb{N}$  and real number  $HB(k) > 0$  (with  $HB$  standing for heavy-bonus) such that

$$\Pr[\text{PPSZ succeeds}] \geq 2^{-n+s_k n+HB(k) \cdot \text{indeg}(\text{HEAVY}(l))-o(n)} .$$

**PROOF.** We define a new distribution  $D$  on placements  $\pi : V \rightarrow [0, 1]$ , in which  $\pi(x)$  for  $x \in \text{HEAVY}$  is slightly biased towards smaller values. For this, we fix some continuous differentiable  $\gamma : [0, 1] \rightarrow \mathbf{R}_0^+$  such that  $\gamma(0) = \gamma(1) = 0$  and let  $\phi := \gamma'$  be its derivative. For sufficiently small  $\epsilon > 0$ , it holds that  $1 + \epsilon\phi(r) \geq 0$  for all  $r \in [0, 1]$ . This implies that  $\int_0^r (1 + \epsilon\phi(x)) dx = r + \epsilon\gamma(r) =: \Phi(r)$  is monotonically increasing,  $\Phi(0) = 0$  and  $\Phi(1) = 1$ . Thus,  $\Phi$  is the cumulative distribution function of a distribution on  $[0, 1]$ .

**DEFINITION 4.4.** Let  $D_\epsilon^\gamma$  be the distribution on  $[0, 1]$  with the cumulative probability distribution function  $r + \epsilon\gamma(r)$  and density function  $1 + \epsilon\phi(r)$ .

In this proof, we will state all intermediate results in terms of a general function  $\gamma$  satisfying the above requirements. If we want to show *some* improvement for  $k$ -SAT, rather than concrete numerical results, the precise shape of  $\gamma$  does not matter. We might just as well define  $\gamma(r) = r(1 - r)$  and run with it. In the following, we ask the reader to interpret  $\gamma$  as the general function described above; however, since we will be juggling several constants, which depend on  $k$  and our choice of  $\gamma$ , it will be advantageous sometimes to think of  $\gamma$  as a very concrete function, not depending on anything, for example like  $\gamma(r) = r(1 - r)$ . For this choice, we have  $\phi(r) = 1 - 2r$ , and  $D_\epsilon^\gamma$  is a distribution on  $[0, 1]$  for all  $0 \leq \epsilon \leq 1$ .

Let  $D$  be the distribution on placements  $\pi : V \rightarrow [0, 1]$  that samples  $\pi(x) \in [0, 1]$  uniformly for all  $x \notin \text{HEAVY}$  and  $\pi(x) \sim D_\epsilon^\gamma$  for all  $x \in \text{HEAVY}$ . For heavy  $x$  it holds that  $\Pr[\pi(x) < r] = r + \epsilon\gamma(r) \geq r$ . By using the indicator notation  $[\text{STATEMENT}]$  that gives 1 if  $\text{STATEMENT}$  holds and 0 otherwise, we can concisely write

$$\Pr_{\pi \sim D} [\pi(x) < r] = r + [x \in \text{HEAVY}] \epsilon \gamma(r) .$$

Loosely speaking, heavy variables  $x$  tend to come *earlier* in  $\pi \sim D$  than non-heavy ones. Consequently, for heavy  $x$ , we expect  $\Pr[\text{Forced}(x, \pi)]$  to be *smaller* under  $D$  than under the uniform distribution  $U_{[0,1]}$ . The following lemma bounds the magnitude of this “penalty”:

**LEMMA 4.5.** *If  $x \in \text{HEAVY}$  then  $\Pr_{\pi \sim D}[\text{Forced}(x, \pi)]$  is at least*

$$\int_0^1 Q_r^{(k)}(1 + \epsilon\phi(r)) dr - o(1) = s_k - \epsilon \cdot \text{HP}(k) - o(1)$$

for  $\text{HP}(k) := -\int_0^1 Q_r^{(k)}\phi(r)dr$ , which depends only  $k$  and on  $\gamma$ , but not on  $k'$  nor on  $\epsilon$ . The acronym  $\text{HP}$  stands for heavy-penalty.

Remark: the minus sign in the definition of  $\text{HP}$  might look surprising, but the integral  $\int_0^1 Q_r^{(k)}\phi(r)dr$  is indeed at most 0, which can be proved by basic calculus or simply seen from the context (but this proof is not necessary for the correctness of the lemma).

**Proof.** Under  $\pi \sim D$ , the values  $\pi(y)$  are independent for all variables  $y$ . Let  $U$  denote the uniform distribution on placements. Observe that we can define a coupling  $(\pi_D, \pi_U)$  such that  $\pi_D \sim D$  and  $\pi_U \sim U$  and  $\pi_D(y) \leq \pi_U(y)$  hold always and for all variables  $y$ . We could, for example, achieve this by sampling  $u \in [0, 1]$  uniformly at random, let  $t$  be the unique number in  $[0, 1]$  for which  $t + \epsilon\gamma(t) = u$ , and set  $\pi_U(y) = u$  and  $\pi_D(y) = t$ . Then  $\pi_U(y)$  is uniform over  $[0, 1]$  and  $\pi_D(y)$  is distributed according to  $D_\epsilon^\gamma$ . Furthermore,  $t \leq u$  holds.

Since  $\text{Cut}_r(T_x)$  depends monotonically on the atomic events  $[\pi(y) < r]$  and since  $[\pi_U < r]$  implies  $[\pi_D < r]$ , we conclude that  $\pi_U \in \text{Cut}_r(T_x)$  also implies  $\pi_D \in \text{Cut}_r(T_x)$ ; in other words,

$$\Pr_D[\text{Cut}_r(T_x)] \geq \Pr_U[\text{Cut}_r(T_x)]. \quad (8)$$

From the previous section, in particular Lemma 3.11, we know that

$$\Pr_U[\text{Cut}_r(T_x)] \geq Q_r^{(k)} - \text{Error}(r, h). \quad (9)$$

The lemma now follows from setting  $r = \pi(x)$  and taking expectation over  $\pi(x)$ , and using the fact that  $\pi(x) \sim D$ . That is, we compute

$$\begin{aligned} \Pr_D[\text{Cut}(T_x)] &= \mathbb{E}_{r \sim D} \left[ \Pr_D[\text{Cut}(T_x)] \right] \\ &= \int_0^1 \Pr_D[\text{Cut}_r(T_x)](1 + \epsilon\phi(r)) dr && \text{(multiply by density and integrate)} \\ &\geq \int_0^1 \Pr_U[\text{Cut}_r(T_x)](1 + \epsilon\phi(r)) dr && \text{(by (8))} \\ &\geq \int_0^1 \left( Q_r^{(k)} - \text{Error}(r, h) \right) (1 + \epsilon\phi(r)) dr && \text{(by (9))} \\ &= \int_0^1 Q_r^{(k)}(1 + \epsilon\phi(r)) dr - \int_0^1 \text{Error}(r, h)(1 + \epsilon\phi(r)) dr. \end{aligned}$$

Now since  $\phi(r)$  is bounded and  $\text{Error}(r, h)$  converges to 0 as  $h$  grows, for each fixed  $r$ , the second integral in the above expression also converges to 0 as  $h$  grows, which explains the  $o(1)$  term in the statement of the lemma. ◆

If  $x \notin \text{HEAVY}$  but has a heavy out-neighbor, i.e., if there is an arc  $(x, y)$  for some  $y \in \text{HEAVY}$ , we expect  $\Pr[\text{Forced}(x, \pi)]$  to be *larger* under  $D$ . Also, if there are several heavy out-neighbors, we expect this “bonus” to be even larger. To formalize this intuition, we introduce some notation.

**DEFINITION 4.6.** For  $x \in V$  and  $Y \subseteq V$ , we define  $e(x, Y)$  to be the number of arcs  $(x, y)$  with  $y \in Y$ . For sets  $X, Y \subseteq V$  (not necessarily disjoint), we define  $e(X, Y)$  to be  $\sum_{x \in X} e(x, Y)$ .

The next lemma formally states that the “bonus” for  $x \notin \text{HEAVY}$  is proportional to its number of heavy out-neighbors:

**LEMMA 4.7.** *If  $x \notin \text{HEAVY}$  then  $\Pr_{\pi \sim D}[\text{Forced}(x, \pi)]$  is at least*

$$s_k + \epsilon \text{HCB}(k) e(x, \text{HEAVY}) - o(1) \quad (10)$$

where  $e(x, \text{HEAVY})$  is the number of arcs  $(x, y)$  with  $y \in \text{HEAVY}$  and

$$\text{HCB}(k) := \int_0^1 \gamma(r) \left(P_r^{(k)}\right)^{k-2} \left(1 - Q_r^{(k)}\right) dr.$$

The acronym HCB stands for heavy-child bonus. Note that  $\text{HCB}(k)$  only depends on  $\gamma$  and  $k$ . Furthermore,  $\text{HCB}(k) > 0$  as long as  $\gamma$  is positive somewhere in  $[0, \frac{k-2}{k-1}]$ , which our concrete choice  $r(1-r)$  certainly is.

**Proof.** We first show the last part of the lemma, namely that  $\text{HCB}(k) > 0$ . It is well-known from the theory of Galton-Watson branching processes that  $Q_r^{(k)} < 1$  for all  $r < \frac{k-2}{k-1}$  and thus  $\left(P_r^{(k)}\right)^{k-2} \left(1 - Q_r^{(k)}\right) > 0$  on the interval  $[0, \frac{k-2}{k-1}]$ . This means that  $\text{HCB}(k) > 0$  for our choice  $\gamma(r) = r(1-r)$  (and basically for any other legal choice for  $\gamma$ , provided that it is positive somewhere on the interval  $[0, \frac{k-2}{k-1}]$ ).

We now prove (10). Let  $y_1, \dots, y_{k-1}$  be the labels of the children of the root of  $T_x$  and  $T_i$  be the subtree of  $T_x$  rooted at  $y_i$ . Similar to Step 2 in the proof of Lemma 3.11, we can assume that all nodes of  $T_x$  have distinct variable labels. Formally, we need a more general version of Proposition 3.12, one that deals with non-uniform distributions:

**PROPOSITION 4.8.** *Let  $T$  be a labeled tree and  $v$  a node therein. Let  $a'$  be a fresh label, i.e., one that does not appear in  $T$ . Define  $T_{v \rightarrow a'}$  to be the same as  $T$  but with  $\text{varlabel}(v) := a'$  for some fresh label  $a'$ . Furthermore, let  $D$  be a probability distribution on placements  $\pi : L(T) \rightarrow [0, 1]$  under which the variables  $\{\pi(l)\}_{l \in L}$  are independent. Define a new distribution  $D'$  by sampling  $\pi(l) \sim D$  for all labels  $l \in L(T)$  and additionally sampling  $\pi(a')$  from the same distribution as  $\pi(a)$ . Then*

$$\Pr_{D'}[\text{Cut}_r(T_{v \rightarrow a'})] \leq \Pr_D[\text{Cut}_r(T_i)]$$

holds.

The proof of this proposition is almost the same as of Proposition 3.12. For every variable  $z \notin \{x, y_1, \dots, y_{k-1}\}$ , it holds that  $\Pr_D[\pi(z) < r] \geq r$ . This means we can assume pessimistically<sup>5</sup> that  $\pi(z)$  is uniform over  $[0, 1]$ . Since all labels are distinct, we have  $\Pr_D[\text{Cut}_r(T_x)] = \prod_{i=1}^{k-1} \Pr_D[\text{wCut}_r(T_i)]$  and

$$\begin{aligned} \Pr_D[\text{wCut}_r(T_{y_i})] &= \Pr_D[\pi(y_i) < r] \vee \Pr_D[\text{Cut}_r(T_i)] \\ &= (r + [y_i \in \text{HEAVY}]\epsilon\gamma(r)) \vee \Pr_D[\text{Cut}_r(T_i)] \\ &\geq (r + [y_i \in \text{HEAVY}]\epsilon\gamma(r)) \vee (Q_r - o(1)) \\ &= r + [y_i \in \text{HEAVY}]\epsilon\gamma(r) + (1 - r - [y_i \in \text{HEAVY}]\epsilon\gamma(r))(Q_r - o(1)) \\ &= r + (1 - r)Q_r + [y_i \in \text{HEAVY}]\epsilon\gamma(r)(1 - Q_r) - o(1) \\ &= P_r + [y_i \in \text{HEAVY}]\epsilon\gamma(r)(1 - Q_r) - o(1). \end{aligned}$$

Therefore,

$$\begin{aligned} \Pr_D[\text{Cut}_r(T_x)] &\geq \prod_{y:x \rightarrow y} (P_r + [y \in \text{HEAVY}]\epsilon\gamma(r)(1 - Q_r)) - o(1) \\ &\geq (P_r)^{k-1} + \sum_{y:x \rightarrow y} [y \in \text{HEAVY}]\epsilon\gamma(r)(P_r)^{k-2}(1 - Q_r) - o(1) \\ &= Q_r + e(x, \text{HEAVY})\epsilon\gamma(r)(P_r)^{k-2}(1 - Q_r) - o(1). \end{aligned}$$

Since  $\pi(x)$  is uniform over  $[0, 1]$ , we get  $\Pr_{\pi \sim D}[\text{Cut}(T_x)]$  by integrating the above expression over  $[0, 1]$ . This yields (10) and concludes the proof of Lemma 4.7.  $\blacklozenge$

We can prove Theorem 4.3 by summing over all variables:

$$\frac{\sum_{x \in V} \Pr_D[\text{Forced}(x, \pi)] - s_k n}{\epsilon} \tag{11}$$

$$\begin{aligned} &\geq -\text{HP}(k)|\text{HEAVY}| + \text{HCB}(k) \sum_{x \notin \text{HEAVY}} e(x, \text{HEAVY}) - o(n) \\ &\geq -\text{HP}(k)|\text{HEAVY}| + \text{HCB}(k)e(V \setminus \text{HEAVY}, \text{HEAVY}) - o(n). \end{aligned} \tag{12}$$

Next, we show that  $e(V \setminus \text{HEAVY}, \text{HEAVY})$  is large:

$$\begin{aligned} e(V \setminus \text{HEAVY}, \text{HEAVY}) &= e(V, \text{HEAVY}) - e(\text{HEAVY}, \text{HEAVY}) \\ &= \text{indeg}(\text{HEAVY}) - e(\text{HEAVY}, \text{HEAVY}) \\ &\geq \text{indeg}(\text{HEAVY}) - e(\text{HEAVY}, V) \\ &= \text{indeg}(\text{HEAVY}) - k|\text{HEAVY}|, \end{aligned}$$

---

<sup>5</sup> We will not go through a formal coupling argument from now on.

where the last equality follows because every variable, in particular every  $x \in \text{HEAVY}$ , has  $e(x, V) = \text{outdeg}(x) = k - 1$ . Combining the two, we get

$$\begin{aligned} (12) &= -\text{HP}(k)|\text{HEAVY}| + \text{HCB}(k)e(V \setminus \text{HEAVY}, \text{HEAVY}) - o(n) \\ &\geq -\text{HP}(k)|\text{HEAVY}| + \text{HCB}(k) (\text{indeg}(\text{HEAVY}) - k|\text{HEAVY}|) - o(n) \\ &= \text{HCB}(k) \text{indeg}(\text{HEAVY}) - (\text{HP}(k) + k\text{HCB}(k))|\text{HEAVY}| - o(n) \\ &\geq \left( \text{HCB}(k) - \frac{\text{HP}(k) + k\text{HCB}(k)}{l} \right) \text{indeg}(\text{HEAVY}(l)) - o(n), \end{aligned}$$

where the last inequality follows from the fact that  $\text{indeg}(\text{HEAVY}) \geq l|\text{HEAVY}|$ . We also write  $\text{HEAVY}(l)$  in the last line to emphasize that its definition involves the number  $l$ . We choose  $l = l(k)$  sufficiently large to make sure the expression in the parenthesis is some  $c = c(k) > 0$ . This shows that  $\mathbb{E}_D[\text{Forced}(\pi)] \geq s_k n - o(n) + \epsilon c(k) \text{indeg}(\text{HEAVY}(l(k)))$ . Using (3), we see that our “gain” in the exponent of the success probability is at least<sup>6</sup>

$$\log_2 \Pr[\text{PPSZ succeeds}] + n - s_k n \geq \epsilon c(k) \text{indeg}(\text{HEAVY}) - \text{KL}(D||U). \quad (13)$$

Since all values  $\pi(x)$  are independent under both  $D$  and  $U$ , the Kullback-Leibler divergence becomes additive, and  $\text{KL}(D||U) = \text{KL}(D_\epsilon^\gamma||U_{[0,1]}) \cdot |\text{HEAVY}|$ , where  $U_{[0,1]}$  is the uniform distribution on  $[0, 1]$ .

**PROPOSITION 4.9.** *Define  $\Psi := \int_0^1 \phi^2(r) dr$ . Then  $\text{KL}(D_\epsilon^\gamma||U_{[0,1]}) \leq \log_2(e) \epsilon^2 \Psi$  holds and furthermore  $\text{KL}(D||U) \leq \log_2(e) \epsilon^2 \Psi |\text{HEAVY}|$ .*

**Proof.** We abbreviate  $t := \epsilon\phi(r)$ . By definition of KL for continuous distributions, we have

$$\begin{aligned} \ln(2) \text{KL}(D_\epsilon^\gamma||U_{[0,1]}) &= \int_0^1 (1+t) \ln(1+t) dr \\ &\leq \int_0^1 (1+t)t dr && \text{(since } \ln(1+t) \leq t \text{)} \\ &= \int_0^1 t dr + \int_0^1 t^2 dr \\ &= \int_0^1 \epsilon\phi(r) dr + \int_0^1 (\epsilon\phi(r))^2 dr. && \text{(putting back } t = \epsilon\phi(r) \text{)} \end{aligned}$$

The first integral is 0 because  $\int_0^1 \phi(r) dr = \gamma(1) = 0$ ; the second is  $\epsilon^2 \Psi$  by definition of  $\Psi$ . ◆

Combining Proposition 4.9 and (13) gives

$$\begin{aligned} \log_2 \Pr[\text{PPSZ succeeds}] + n - s_k n &\geq \epsilon c(k) \text{indeg}(\text{HEAVY}(l(k))) - \log_2(e) \epsilon^2 \Psi |\text{HEAVY}(l(k))| \\ &\geq \left( \epsilon c(k) - \frac{\log_2(e) \epsilon^2 \Psi}{l(k)} \right) \text{indeg}(\text{HEAVY}(l(k))) \end{aligned}$$

---

6 ignoring the  $o(n)$  term for readability

Since none of  $c(k)$ ,  $l(k)$ ,  $k$ , and  $\Psi$  depends on  $\epsilon$ , we can choose  $\epsilon$  sufficiently small to ensure that the term in parentheses above some  $\text{HB}(k) > 0$  and thus  $\log_2 \Pr[\text{PPSZ succeeds}] \geq -n + s_k n - o(n) + \text{HB}(k) \cdot \text{indeg}(\text{HEAVY}(l(k)))$ , which proves Theorem 4.3. ■

Note that in theory we could hammer out how  $\text{HB}(k)$  depends on  $k$ . This would surely be tedious because we encountered several integrals involving  $Q_r^{(k)}$  and  $P_r^{(k)}$ , for which we don't have a closed form. In its current form, the theorem gives us no explicit bounds on  $\text{HB}(k)$ , other than that it is positive.

## 4.2 Privileged variables—when $\Pr[\text{Forced}(x, \pi)]$ is already larger

There are some abnormal cases that will interfere with our analysis below. Luckily, all those cases will imply that the variables involved already have a substantially *higher* probability of being forced.

**DEFINITION 4.10.** A variable  $x$  is called *privileged* if (1)  $x$  has at least two critical clauses or it has a critical clause tree  $T_x$  such that (2) there is a variable  $y$  that appears simultaneously at depth 1 and 2 or (3)  $T_x$  has fewer than  $(k-1)^2$  nodes at depth 2. Let PRIVILEGED be the set of all privileged variables.

**LEMMA 4.11.** *There is some value  $\text{PRIV} > 0$ , depending only on  $k$ , such that  $\Pr[\text{Forced}(x, \pi)] \geq s_k + \text{PRIV} - o(1)$  for all privileged variables  $x$ , where  $o(1)$  converges to 0 as  $w$  grows.*

We prove the lemma in Appendix A. The proof is somewhat technical, partially overlaps with proofs also found in Hansen et al. [3], but is conceptually rather straightforward.

## 4.3 The almost regular case

Theorem 4.3 already gives us an exponential improvement over the old analysis of PPSZ provided that  $\text{indeg}(\text{HEAVY})$  is large (linear in  $n$ ). In this section, we will come up with a corresponding bound that works well if  $\text{indeg}(\text{HEAVY})$  is small. The final bound will then follow from a meet-in-the-middle argument.

**LEMMA 4.12.** *There is a collection  $G$  of canonical critical clauses such that no two clauses in  $G$  share a variable and  $|G| \geq \frac{n - \text{indeg}(\text{HEAVY}(l))}{kl}$ .*

**PROOF.** Set  $C$  to be the set of all canonical critical clauses. We have  $|C| = n$  because there is exactly one critical clause for each variable. Greedily pick a clause  $C \in C$ , add it to  $G$ , and delete from  $C$  all clauses  $C'$  that share a variable with  $C$  (this obviously includes  $C$  itself). Repeat this step for as long as possible.

How many clauses does each step remove from  $C$ ? Let  $x_1, \dots, x_k$  denote the variables of  $C$ . For sure we remove the canonical critical clauses of  $x_1, \dots, x_k$ . Additionally, we remove,

for each  $1 \leq i \leq k$ , all canonical critical clauses containing  $\bar{x}_i$ . This removes at most a total of  $k + \sum_{x \in \text{var}(C)} \text{indeg}(x)$  clauses. Thus, the total number of canonical critical clauses removed in this process is at most

$$\begin{aligned} & \sum_{C \in G} \left( k + \sum_{x \in \text{var}(C)} \text{indeg}(x) \right) \\ & \leq |G|k + \sum_{\substack{C \in G \\ x \notin \text{HEAVY}(l)}} \sum_{x \in \text{var}(C)} (l-1) + \sum_{C \in G} \sum_{\substack{x \in \text{var}(C) \\ x \in \text{HEAVY}(l)}} \text{indeg}(x) \\ & \leq |G|k + |G|k(l-1) + \text{indeg}(\text{HEAVY}(l)) = |G|kl + \text{indeg}(\text{HEAVY}(l)). \end{aligned}$$

The process ends  $C$  becomes empty. Since  $|C| = n$  in the beginning, it removes exactly  $n$  clauses, and therefore  $|G|kl + \text{indeg}(\text{HEAVY}(l)) \geq n$ . Solving for  $|G|$  proves the lemma. ■

We take a set  $G$  of canonical critical clauses as guaranteed by the lemma. We form a collection  $M'$  of disjoint pairs of variables by selecting, for each  $C \in G$ , two negative literals  $\bar{y}, \bar{z} \in C$  and adding  $\{y, z\}$  to  $M'$ .<sup>7</sup> Call  $x$  a *parent* of  $\{y, z\}$  if the canonical critical clause of  $x$  contains the literals  $\bar{y}$  and  $\bar{z}$ . Note that this name makes sense since in the canonical critical clause tree  $T_x$  of  $x$ , the root has two children with labels  $y$  and  $z$ , respectively. Every  $\{y, z\} \in M'$  has at least one parent. We form a final collection  $M \subseteq M'$  of pairs by removing each pair  $\{y, z\}$  with parent  $x$  from  $M'$  if at least one of  $x, y, z$  is in PRIVILEGED. Each privileged variable  $x$  is “responsible” for the removal of at most two elements from  $M'$ : one if the canonical clause of  $x$  happens to be in  $G$ ; one if there is some  $x'$  with  $\{x, x'\} \in M'$ . Therefore,

$$|M| \geq \frac{n - \text{indeg}(\text{HEAVY})}{kk'} - 2 |\text{PRIVILEGED}|. \quad (14)$$

We denote the set of all parents  $x$  of some  $\{y, z\} \in M$  by PARENTM.

**THEOREM 4.13.** *For every  $c > 0$  and  $k \geq 3$  there is some  $c' > 0$  such that if  $|M| \geq cn$  then  $\Pr[\text{PPSZ succeeds}] \geq 2^{-n+s_k n+c'n-o(1)}$ .*

From here, the proof of Theorem 1.5 is simple. Let  $c_1$  be a small constant, depending on  $k$ . If  $|\text{indeg}(\text{HEAVY})| \geq c_1 n$  then we can apply Theorem 4.3. If  $|\text{PRIVILEGED}| \geq c_1 n$  we can apply Lemma 4.11. Otherwise, (14) implies that  $|M| \geq c_2 n$  for some  $c_2$  depending on  $c_1$  and  $k$ , and  $c_2 > 0$  if  $c_1$  is small enough. We can now apply Theorem 4.13 and are done. This proves Theorem 1.5. It remains to prove Theorem 4.13.

The remainder of this section contains the proof of Theorem 4.13. At first reading of what follows, it might even be helpful to think of PRIVILEGED as being empty.

---

<sup>7</sup> For  $k = 3$ , the clause  $C$  contains exactly two negative literals; for larger  $k$ , we select two literals arbitrarily.

#### 4.4 Using disjoint pairs to define a distribution

We choose some  $\rho \leq \frac{k-2}{k-1}$ , to be determined later. Define  $\gamma : [0, 1] \rightarrow \mathbb{R}_0^+$  by

$$\gamma(r) := \begin{cases} r(\rho - r) & \text{if } r \leq \rho \\ 0 & \text{if } r \geq \rho. \end{cases} \quad (15)$$

Let  $\phi := \gamma'$  and extend this via  $\phi(\rho) := -\rho$ . Observe that  $\phi_{\min} := \min_{r \in [0,1]} \phi(r) = -\rho$ . We fix some  $\epsilon > 0$ . Let  $D_e^{\gamma, \square}$  be the distribution on  $[0, 1] \times [0, 1]$  whose density at  $(r, s)$  is  $1 + \epsilon\phi(r)\phi(s)$ . This really is a density, provided that  $1 + \epsilon\phi(r)\phi(s) \geq 0$  for all  $r$  and  $s$ . Let  $D$  be the distribution on placements that samples  $(\pi(y), \pi(z)) \sim D_e^{\gamma, \square}$  for each  $\{y, z\} \in M$  and samples  $\pi(x) \in [0, 1]$  uniformly for each remaining variable. All samplings are done independently. We define

$$\delta = \delta(r) := \epsilon |\phi_{\min}| \gamma(r) = \epsilon \rho \gamma(r) \quad (16)$$

**LEMMA 4.14.** *Let  $r \in [0, 1]$ . Then*

$$\Pr_D[\text{Cut}(T_x) \mid \pi(x) = r] \geq Q_r - \text{DAMAGE}(r) + \text{BENEFIT}(r) \cdot \mathbf{1}_{x \in \text{PARENTM}} - o(1),$$

where

$$\text{DAMAGE}(r) := (k-1)(1-r)P_r^{k-2}\delta Q_r' \quad (17)$$

$$\text{BENEFIT}(r) := \epsilon \gamma^2(r)(1-Q_r)^2 P_{r-\delta}^{k-3}. \quad (18)$$

and the  $o(1)$  converges to 0 as  $h$  grows.

**PROOF.** The proof works by constructing an easy-to-analyze tree and distribution that serve as a pessimistic estimate for  $\text{Cut}_r(T_x)$ . First, we make a simple but important observation about  $M$  and the labels in  $T_x$ :

**OBSERVATION 4.15.** *Let  $u$  be a node of depth 1 in  $T_x$  and  $y = \text{varlabel}(u)$ . Then  $\{x, y\} \notin M$ .*

**PROOF.** Suppose  $\{x, y\} \in M$ , for the sake of contradiction, and let  $a$  be one of their parents. So  $T_a$  contains two nodes  $X, Y$  at depth 1 with labels  $x$  and  $y$ , respectively. What is the clause label  $C_X$  of node  $X$  in  $T_a$ ? First, if it is  $C_x$ , the canonical critical clause of  $x$ , then  $X$  has a child with label  $y$ , since  $\bar{y} \in C_x$ . This means that  $y$  appears in  $T_a$  at level 1 and 2, so  $a$  is privileged. Second, if it is not  $C_x$  but is *some* critical clause, it must be a critical clause for  $x$  or  $a$ , meaning that  $x$  or  $a$  has at least two critical clauses, so at least one of them is privileged. Third, if  $C_X$  is not a critical clause, then  $C_X$  has at most  $k-2$  negative literals and  $X$  has at most  $k-2$  children, meaning  $T_a$  has fewer than  $(k-1)^2$  nodes at level 2. In either case, at least one of  $a$  and  $x$  is privileged, meaning we would have eliminated  $\{x, y\}$  from  $M$ . This is a contradiction. ■

This observation has two important consequences:

**OBSERVATION 4.16.** Suppose  $x \notin \text{PARENTM}$  and let  $y_1, \dots, y_{k-1}$  be the labels of the depth-1-nodes in  $T_x$ . Then  $\pi(x), \pi(y_1), \dots, \pi(y_{k-1})$  are independent and uniform under  $D$ .

**OBSERVATION 4.17.** Suppose  $x \in \text{PARENTM}$  is a parent of  $\{y, z\} \in M$ . Let  $y, z, v_1, \dots, v_{k-3}$  be the labels of the depth-1-nodes in  $T_x$ . Then  $\pi(x), (\pi(y), \pi(z)), \pi(v_1), \dots, \pi(v_{k-3})$  are independent under  $D$ , the pair  $(\pi(y), \pi(z))$  has distribution  $D_\epsilon^{y, \square}$ , and the other  $k - 2$  variables are uniform over  $[0, 1]$ .

The upshot is that we completely understand the distribution of  $\pi(l)$  for the labels on the level 0 and 1 of  $T_x$ . Starting from level 2 downwards, the distribution can become complicated, so we resort to a pessimistic estimate:

**OBSERVATION 4.18.** Let  $v$  be a variable and let  $\tau : V \setminus \{v\} \rightarrow [0, 1]$  be a particular placement of all other variables. Let  $r \in [0, 1]$ . Then  $\Pr_D[\pi(v) < r \mid \tau] \geq r - \delta$ , for  $\delta$  as defined in (16).

**PROOF.** If  $v$  is not contained in any pair of  $M$ , then  $\Pr_D[\pi(v) < r \mid \tau] = r$ . If  $\{v, w\} \in M$  then  $\Pr_D[\pi(v) < r \mid \tau] = \Pr_{D_\epsilon^{y, \square}}[\pi(v) < r \mid \pi(w) = \tau(w)] = r + \epsilon\gamma(r)\phi(\tau(w)) \geq r - \epsilon|\phi_{\min}|\gamma(r)$ . ■

**The two pessimistic distributions  $D_M$  and  $D_{\bar{M}}$ .** Fix  $r \in [0, 1]$  and let  $T^\infty$  be the complete infinite  $(k - 1)$ -ary tree in which all labels are distinct. Since all the labels are distinct, we take the liberty of writing  $\pi(v)$  instead of  $\pi(\text{varlabel}(v))$  for a node  $v$  in  $T^\infty$ . We specify two distributions  $D_M$  and  $D_{\bar{M}}$  on placements  $L \rightarrow [0, 1]$ . First, we set  $\Pr[\pi(v) < r] := r - \delta$  under both  $D_{\bar{M}}$  and  $D_M$  for all nodes  $v$  of depth at least 2 in  $T^\infty$ . Second, we let  $y_1, \dots, y_{k-1}$  be the nodes of depth 1 in  $T^\infty$  and sample all  $\pi(y_i) \in [0, 1]$  uniformly and independently under  $D_{\bar{M}}$ . Under  $D_M$ , we sample  $\pi(y_3, \dots, y_{k-1})$  uniformly and independently but sample  $(\pi(y_1), \pi(y_2)) \sim D_\epsilon^{y, \square}$ . This does not fully specify a distribution on placements  $L \rightarrow [0, 1]$  but it does specify the joint distribution of the events  $[\pi(v) < r]$ . Since we are only interested in  $\Pr[\text{Cut}_r(T_x)]$ , this is enough. Note that we also do not need to specify a distribution for  $\pi(\text{root})$ .

**LEMMA 4.19.** If  $x \notin \text{PARENTM}$  then  $\Pr[\text{Cut}(T_x) \mid \pi(x) = r] \geq \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] - o(1)$ . If  $x \in \text{PARENTM}$  then  $\Pr[\text{Cut}(T_x) \mid \pi(x) = r] \geq \Pr_{D_M}[\text{Cut}_r(T^\infty)] - o(1)$ .

The  $o(1)$ -term comes from the fact that  $T_x$  is a critical clause tree of height  $h$ , whereas  $T^\infty$  is an infinite tree.

**PROOF.** We mimic the proof of Lemma 3.11 by assigning each node at depth 2 or greater a fresh label. We need a third version of Proposition 3.12:

**PROPOSITION 4.20.** Let  $T$  be a labeled tree and  $v$  a node therein. Let  $a'$  be a fresh label, i.e., one that does not appear in  $T$ . Define  $T_{v \rightarrow a'}$  to be the same as  $T$  but with  $\text{varlabel}(v) := a'$  for some fresh label  $a'$ .

Let  $D$  be a probability distribution on placements  $\pi : L(T) \rightarrow [0, 1]$  satisfying the following property: for every label  $l \in L(T)$  and placement  $\tau : L(T) \setminus \{l\} \rightarrow [0, 1]$ , it holds that  $\Pr_D[\pi(l) < r \mid \tau] \geq r - \delta$ . Define a new distribution  $D'$  by sampling a placement  $\pi : L(T) \rightarrow [0, 1]$  from  $D$  and then, additionally, sampling  $\pi(a')$  such that  $\Pr_{D'}[\pi(a') < r] = r - \delta$ . Then

$$\Pr_{D'}[\text{Cut}_r(T_{v \rightarrow a'})] \leq \Pr_D[\text{Cut}_r(T_i)]$$

holds.

The proof of this proposition is almost the same as of Proposition 3.12. We apply it repeatedly, to each node of depth 2 or greater, noting that the distribution  $D$  always satisfies the requirement  $\Pr_D[\pi(l) < r \mid \tau] \geq r - \delta$ , by Observation 4.19. After this process, only the nodes at level 0 and 1 retain their old labels (and thus their original joint distribution under  $\pi$ ). We apply Observations 4.16 and 4.17 to conclude that we have indeed arrived at distributions  $D_M$  and  $D_{\bar{M}}$ , respectively. ■

**PROPOSITION 4.21.** *With  $BENEFIT(r) = \epsilon\gamma^2(r)(1 - Q_r)^2 P_{r-\delta}^{k-3}$  as defined in (18), it holds that  $\Pr_{D_M}[\text{Cut}_r(T^\infty)] \geq \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] + BENEFIT(r)$ .*

**PROOF.** Let  $T_1, \dots, T_{k-1}$  be the subtrees of  $T^\infty$  rooted at the nodes of depth 1. Let  $\tau : L \setminus \{y_1, y_2\} \rightarrow [0, 1]$  be a partial placement. Observe that the marginal distribution  $D'$  of  $\tau$  is the same under  $D_M$  and  $D_{\bar{M}}$ . We call  $\tau$  *critical* if  $\text{Cut}_r(T_1)$  and  $\text{Cut}_r(T_2)$  do *not* happen but  $w\text{Cut}_r(T_3), \dots, w\text{Cut}_r(T_{k-1})$  do happen. This can be determined by looking at  $\tau$  alone.

**CLAIM 4.22.** *If  $\tau$  is not critical then  $\Pr_{D_M}[\text{Cut}_r(T^\infty) \mid \tau] = \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty) \mid \tau]$ .*

**PROOF.** There are several reasons why  $\tau$  is not critical. First, if some of  $w\text{Cut}_r(T_3), \dots, w\text{Cut}_r(T_{k-1})$  does not happen then  $\text{Cut}_r(T^\infty)$  will not happen, whatever the values of  $\pi(y_1)$  and  $\pi(y_2)$ . Second, suppose both of  $\text{Cut}_r(T_1)$  and  $\text{Cut}_r(T_2)$  happen. Then  $w\text{Cut}_r(T_i)$  happens for all  $1 \leq i \leq k-1$  and  $\text{Cut}_r(T^\infty)$  happens for sure. Third, suppose exactly one of  $\text{Cut}_r(T_1)$  and  $\text{Cut}_r(T_2)$  happens, say  $\text{Cut}_r(T_2)$ . This means that  $\text{Cut}_r(T^\infty)$  happens if and only if  $\pi(y_1) < r$ . This event has probability  $r$  under both  $D_{\bar{M}}$  and  $D_M$ . For  $D_{\bar{M}}$  this is immediate from the definition; for  $D_M$  this follows from  $D_e^{\gamma, \square}$  having uniform marginals. ■

If  $\tau$  is critical then

$$\begin{aligned} \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty) \mid \tau] &= \Pr_{D_{\bar{M}}}[\pi(y_1) < r \wedge \pi(y_2) < r] = r^2, \\ \Pr_{D_M}[\text{Cut}_r(T^\infty) \mid \tau] &= \Pr_{D_M}[\pi(y_1) < r \wedge \pi(y_2) < r] = r^2 + \epsilon\gamma^2(r). \end{aligned}$$

To bound the probability that  $\tau$  is critical, note that  $\Pr_{D'}[\text{Cut}_r(T_i)] = Q_{r-\delta}$  since each of the atomic events  $[\pi(l) < r]$  that are relevant for  $\text{Cut}_r(T_i)$  happen with probability  $r - \delta$ . Furthermore,

$Q_{r-\delta} \leq Q_r$ . Next,  $\Pr_{D'}[\text{wCut}_r(T_i)] \geq P_{r-\delta}$  since each of the relevant atomic events happen with probability *at least*  $r - \delta$  (the event at the root of  $T_i$ , namely  $[\pi(y_i) < r]$ , happens with probability exactly  $r$ , which is at least  $r - \delta$ ). Therefore,

$$\Pr[\neg\text{Cut}_r(T_1)] \cdot \Pr[\neg\text{Cut}_r(T_2)] \cdot \prod_{i=3}^{k-1} \Pr[\text{wCut}_r(T_i)] \geq (1 - Q_r)^2 P_{r-\delta}^{k-3},$$

and therefore

$$\begin{aligned} \Pr_{D_M}[\text{Cut}_r(T^\infty)] &= \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] + \epsilon \gamma^2(r) \Pr[\tau \text{ is critical}] \\ &\geq \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] + \epsilon \gamma^2(r) (1 - Q_r)^2 P_{r-\delta}^{k-3}. \end{aligned}$$

This completes the proof of Proposition 4.21 ■

**PROPOSITION 4.23.** *Let  $\text{DAMAGE}(r) := (k - 1)(1 - r)P_r^{k-2}\delta Q'_r$  as defined in (17). Then it holds that  $\Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] \geq Q_r - \text{DAMAGE}(r)$ .*

**PROOF.** If  $r \geq \frac{k-2}{k-1}$  then  $\delta = 0$ ,  $\text{DAMAGE} = 0$ , and  $\Pr[\pi(v) < r] = r$  for every node  $v$  in  $T^\infty$ . Both sides of the inequality evaluate to 1.

Otherwise, let  $T_1, \dots, T_{k-1}$  be the subtrees of  $T^\infty$  rooted at the depth-1-nodes of  $T^\infty$ . Since  $\Pr_{D_{\bar{M}}}[\pi(v) < r] = r - \delta$  for all nodes  $v$  of  $T^\infty$  of depth 2 or greater, it holds that  $\Pr_{D_{\bar{M}}}[\text{Cut}_r(T_i)] = Q_{r-\delta}$  for  $1 \leq i \leq k - 1$ . Since  $Q_r$  is convex on  $[0, \frac{k-2}{k-1}]$ , this is at least  $Q_r - \delta Q'_r$ . Next,

$$\begin{aligned} \Pr_{D_{\bar{M}}}[\text{wCut}_r(T_i)] &= r \vee \Pr_{D_{\bar{M}}}[\text{Cut}_r(T_i)] \geq r \vee (Q_r - \delta Q'_r) \\ &= r + (1 - r)(Q_r - \delta Q'_r) = P_r - (1 - r)\delta Q'_r \\ &= P_r \left( 1 - \frac{(1 - r)\delta Q'_r}{P_r} \right), \end{aligned}$$

and

$$\begin{aligned} \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] &= \prod_{i=1}^{k-1} \Pr_{D_{\bar{M}}}[\text{wCut}_r(T_i)] \\ &\geq P_r^{k-1} \left( 1 - \frac{(1 - r)\delta Q'_r}{P_r} \right)^{k-1} \\ &\geq P_r^{k-1} \left( 1 - \frac{(k - 1)(1 - r)\delta Q'_r}{P_r} \right) \\ &= Q_r - (k - 1)(1 - r)P_r^{k-2}\delta Q'_r. \end{aligned}$$

This completes the proof. ■

From here on, we estimate for  $x \notin \text{PARENTM}$ :

$$\begin{aligned} \Pr_D[\text{Cut}(T_x) \mid \pi(x) = r] &\geq \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] - o(1) && \text{(by Observation 4.19)} \\ &\geq Q_r - \text{DAMAGE}(r) - o(1) && \text{(by Proposition 4.23)} \end{aligned}$$

and for  $x \in \text{PARENTM}$ :

$$\begin{aligned} \Pr_D[\text{Cut}(T_x) \mid \pi(x) = r] &\geq \Pr_{D_M}[\text{Cut}_r(T^\infty)] - o(1) && \text{(by Observation 4.19)} \\ &\geq + \Pr_{D_{\bar{M}}}[\text{Cut}_r(T^\infty)] + \text{BENEFIT}(r) - o(1) && \text{(by Proposition 4.21)} \\ &\geq Q_r - \text{DAMAGE}(r) + \text{BENEFIT}(r) - o(1) . && \text{(by Proposition 4.23)} \end{aligned}$$

This concludes the proof of Lemma 4.14. ■

We obtain a lower bound on  $\Pr_D[\text{Cut}(T_x)]$  by integrating the bound in Lemma 4.14 over  $r$ :

$$\Pr[\text{Cut}(T_x)] \geq s_k - \text{DAMAGE} + \text{BENEFIT} \cdot \mathbf{1}_{x \in \text{PARENTM}} - o(1) , \quad (19)$$

where  $\text{DAMAGE} = \int_0^1 \text{DAMAGE}(r) dr$  and  $\text{BENEFIT} = \int_0^1 \text{BENEFIT}(r) dr$ . To simplify the integration, we will first give an upper bound on  $\text{DAMAGE}(r)$  and a lower bound on  $\text{BENEFIT}(r)$ . Recall our definition of  $\gamma : [0, 1] \rightarrow \mathbb{R}_0^+$  in (15) as used in  $D_e^{\gamma, \square}$ :

$$\gamma(r) := \begin{cases} r(\rho - r) & \text{if } r \leq \rho \\ 0 & \text{if } r \geq \rho. \end{cases}$$

**PROPOSITION 4.24.** *The following bounds hold:*

$$\text{DAMAGE} \leq O(\epsilon \rho^{2k}) \quad (20)$$

$$\text{BENEFIT} \geq \Omega(\epsilon \rho^{k+2}) , \quad (21)$$

where the  $O$  hides factors depending solely on  $k$  and terms of order  $\rho^a$  for  $a \geq 2k + 1$ , and the  $\Omega$  hides factors depending solely on  $k$  and terms of order  $\rho^b$  for  $b \geq k + 2$ .

**PROOF.** We remind the reader of the definitions of  $\text{DAMAGE}$  and  $\text{BENEFIT}$  in (17) and (18):

$$\begin{aligned} \text{DAMAGE}(r) &= (k-1)(1-r)P_r^{k-2}\delta Q'_r \\ \text{BENEFIT}(r) &= \epsilon \gamma^2(r)(1-Q_r)^2 P_{r-\delta}^{k-3} \end{aligned}$$

Both  $\text{BENEFIT}(r)$  and  $\text{DAMAGE}(r)$  vanish for  $r \geq \rho$ . Thus, we can replace  $\int_0^1$  by  $\int_0^\rho$ . We will first bound  $\text{BENEFIT}(r)$ . On the interval  $[0, \rho]$ ,  $\gamma(r) = r(\rho - r)$ , and  $1 - Q_r \geq 1 - Q_\rho$ , and  $P_{r-\delta} \geq r - \delta = r(1 - \epsilon\rho(\rho - r))$ . Therefore,

$$\begin{aligned} \text{BENEFIT}(r) &\geq \epsilon r^2 (\rho - r)^2 (1 - Q_\rho)^2 r^{k-3} (1 - \epsilon\rho(\rho - r))^{k-3} \\ &\geq \frac{1}{2} \epsilon r^{k-1} (\rho - r)^2 , \end{aligned}$$

for sufficiently small  $\epsilon$  and  $\rho$  (smaller than a value depending solely on  $k$ ). Integrating this over  $r \in [0, \rho]$  shows (21).

Next, we bound  $\text{DAMAGE}(r)$  from above. It holds that  $r \leq P_r \leq \frac{k-1}{k-2}r$ , where the first inequality follows immediately from  $P_r = r \vee Q_r$ , and the second follows from the fact that  $P_r$  is convex on  $[0, \frac{k-2}{k-1}]$ ,  $P_0 = 0$ , and  $P_{\frac{k-2}{k-1}} = 1$ . Third, we compute  $Q'_r = (P_r^{k-1})' = (k-1)P_r^{k-2}P'_r \leq (k-1) \left(\frac{k-1}{k-2}\right)^{k-2} r^{k-2} P'_r$ . To bound  $P'_r$ , observe again that  $P'_r \leq P'_{\frac{k-2}{k-1}}$ , where  $P'_{\frac{k-2}{k-1}}$  is the *left derivative* since  $P_r$  is not differentiable at  $r = \frac{k-2}{k-1}$ . To determine the left derivative, recall that  $P = P_r$  satisfies the equation

$$P = P^{k-1} + (1 - P^{k-1})r$$

and therefore

$$r = r(P) = \frac{P - P^{k-1}}{1 - P^{k-1}}.$$

For  $P \rightarrow 1$ , the derivative of  $P \mapsto r(P)$  converges to  $\frac{k-2}{2(k-1)}$  (compute the derivative of  $r(P)$  and apply l'Hôpital's rule twice; then substitute  $P = 1$ ). Thus, for  $r \rightarrow \frac{k-2}{k-1}$ , the derivative  $P'_r$  converges to the inverse thereof, to  $\frac{2(k-1)}{k-2}$ . Thus,  $Q'_r \leq (k-1) \left(\frac{k-1}{k-2}\right)^{k-2} r^{k-2} \frac{2(k-1)}{k-2}$ . Putting things together and plugging in the definition of  $\delta$  in (16), we get

$$\begin{aligned} \text{DAMAGE}(r) &= (k-1)(1-r)P_r^{k-2}\delta Q'_r \\ &\leq (k-1)(1-r) \left(\frac{k-1}{k-2}\right)^{k-2} r^{k-2} \epsilon \rho r (\rho - r) (k-1) \left(\frac{k-1}{k-2}\right)^{k-2} r^{k-2} \frac{2(k-1)}{k-2} \\ &= C_k \epsilon (1-r) r^{2k-3} \rho (\rho - r) \end{aligned}$$

for some constant  $C_k$  depending only on  $k$ . Integrating over  $r \in [0, \rho]$  yields (20). ■

Combining (19) with the bounds (20) and (21) and summing over all  $x \in V$ , we obtain

$$\sum_{x \in V} \Pr_D[\text{Cut}(T_x)] \geq s_k n - O(\epsilon \rho^{2k}) n + \Omega(\epsilon \rho^{k+2}) |M| - o(n). \quad (22)$$

Finally, to bound the success probability of PPSZ using the distribution  $D$ , we need to bound  $\text{KL}(D||U)$  from above. By additivity of KL, we see that  $\text{KL}(D||U) = |M| \cdot \text{KL}(D_\epsilon^{y, \square} || U^\square)$ , where  $U^\square$  denotes the uniform distribution on  $[0, 1] \times [0, 1]$ . The density of  $D_\epsilon^{y, \square}$  at  $r, s$  is  $1 + \epsilon \phi(r) \phi(s)$ ,

and therefore

$$\begin{aligned}
 KL(D_\epsilon^{y,\square}||U^2) &= \frac{1}{\ln(2)} \int_{[0,1]^2} (1 + \epsilon\phi(r)\phi(s)) \ln(1 + \epsilon\phi(r)\phi(s)) ds dr \\
 &\leq \frac{1}{\ln(2)} \int_{[0,1]^2} (1 + \epsilon\phi(r)\phi(s))\epsilon\phi(r)\phi(s) ds dr \\
 &= \frac{1}{\ln(2)} \int_{[0,1]^2} \epsilon^2\phi^2(r)\phi^2(s) ds dr \\
 &= \frac{\epsilon^2}{\ln(2)} \left( \int_0^1 \phi^2(r) dr \right)^2 \\
 &= \frac{\epsilon^2\rho^3}{3 \ln(2)}.
 \end{aligned}$$

Thus, using (3), we conclude that the success probability of PPSZ is  $2^{-n+s_k n+\text{gain}}$  where

$$\text{gain} \geq \Omega(\epsilon\rho^{k+2})|M| - O(\epsilon^2\rho^3)|M| - O(\epsilon\rho^{2k})n.$$

Choosing  $\epsilon = \rho^{k-3}$ , this is  $\Omega(\rho^{2k-1})|M| - O(\rho^{3k-3})n$ . Note that  $2k - 1 \leq 3k - 3$  for  $k \geq 3$ . Thus, if  $|M| \geq cn$ , we can choose a sufficiently small  $\rho$ , depending on  $k$  and  $c$ , ensuring that  $\text{gain} \geq c'n$ , for some constant  $c'$  depending on  $c$  and  $k$ . This concludes the proof of Theorem 4.13.

## Acknowledgments

I am grateful to Robin Moser and Timon Hertli, with whom I intensively discussed the PPSZ algorithm, and some ideas used in this paper came from our discussions. I am grateful to Navid Talebanfard for fruitful discussions and helpful comments on this work. Finally, my thanks go to the anonymous reviewers, both of the conference version and of this version, which greatly helped me to improve the paper.

---

## References

- [1] Sven Baumer and Rainer Schuler. Improving a probabilistic 3-sat algorithm by dynamic search and independent clause pairs. *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 150–161. Springer, 2003. [DOI](#) (1)
- [2] Timothy M. Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: quickly derandomizing razborov-smolensky. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1246–1255. SIAM, 2016. [DOI](#) (2)
- [3] Thomas Dueholm Hansen, Haim Kaplan, Or Zamir, and Uri Zwick. Faster  $k$ -sat algorithms using biased-PPSZ. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 578–589. ACM, 2019. [DOI](#) (2, 5, 6, 8, 9, 24, 33)
- [4] Timon Hertli. 3-SAT faster and simpler—unique-SAT bounds for PPSZ hold in general, 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011, pages 277–284. IEEE Computer Soc., Los Alamitos, CA, 2011. [DOI](#) (2, 5, 8, 9)

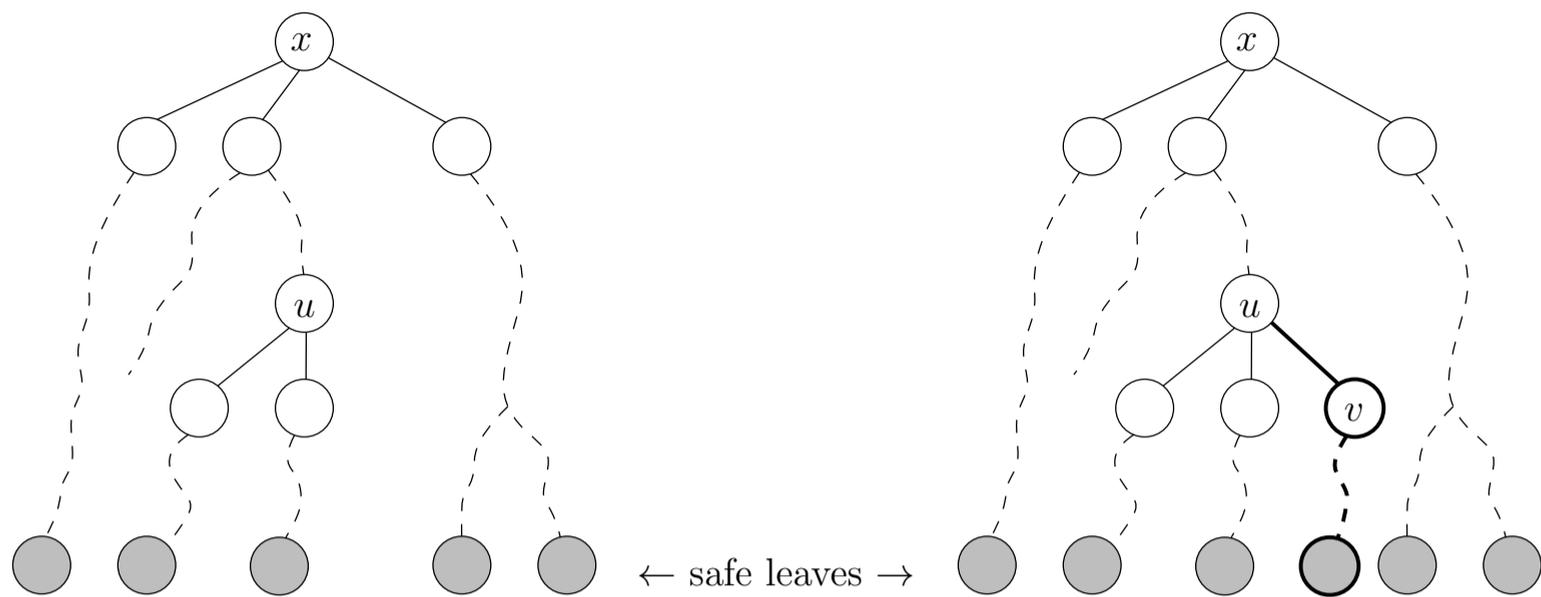
- [5] Timon Hertli. Breaking the PPSZ Barrier for Unique 3-SAT. *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 600–611. Springer, 2014. DOI (2, 5)
- [6] Thomas Hofmeister, Uwe Schöning, Rainer Schuler, and Osamu Watanabe. A probabilistic 3-sat algorithm further improved. *STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes - Juan les Pins, France, March 14-16, 2002, Proceedings*, volume 2285 of *Lecture Notes in Computer Science*, pages 192–202. Springer, 2002. DOI (1)
- [7] Sixue Liu. Chain, generalization of covering code, and deterministic algorithm for k-sat. *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPICs*, 88:1–88:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. DOI (2)
- [8] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for  $k$ -SAT. *J. ACM*, 52(3):337–364 (electronic), 2005. DOI (1–5, 7–10, 12, 15, 17, 35)
- [9] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. *Chicago J. Theoret. Comput. Sci.* Article 11, 19 pp. (electronic), 1999. DOI (1)
- [10] Pavel Pudlák, Dominik Scheder, and Navid Talebanfard. Tighter hard instances for PPSZ. *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, 85:1–85:13, 2017. DOI (5)
- [11] Tong Qin and Osamu Watanabe. An improvement of the algorithm of Hertli for the unique 3SAT problem. *Theor. Comput. Sci.* 806:70–80, 2020. DOI (5)
- [12] Dominik Scheder. PPSZ is better than you think. *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 205–216. IEEE, 2021. DOI (1)
- [13] Dominik Scheder. PPSZ is better than you think. *Electron. Colloquium Comput. Complex.* 69, 2021. URL (1, 6)
- [14] Dominik Scheder and John P. Steinberger. PPSZ for General  $k$ -SAT - making Hertli's analysis simpler and 3-SAT faster. *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, 9:1–9:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. DOI (2, 5, 8, 9)
- [15] Dominik Scheder and Navid Talebanfard. Super strong ETH is true for PPSZ with small resolution width. *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPICs*, 3:1–3:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. DOI (5)
- [16] Dominik Scheder, Bangsheng Tang, Shiteng Chen, and Navid Talebanfard. Exponential lower bounds for the PPSZ  $k$ -sat algorithm. *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1253–1263. SIAM, 2013. DOI (5)
- [17] Uwe Schöning. A probabilistic algorithm for  $k$ -SAT and constraint satisfaction problems, *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 410–414. IEEE Computer Society, Los Alamitos, CA, 1999. DOI (1)
- [18] Nikhil Vyas and R. Ryan Williams. On super strong ETH. *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, pages 406–423. Springer, 2019. DOI (2)

## A. Proofs from Section 3 and 4

**LEMMA 4.11. (Restated)** *There is some value  $PRIV > 0$ , depending only on  $k$ , such that  $\Pr[\text{Forced}(x, \pi)] \geq s_k + PRIV - o(1)$  for all privileged variables  $x$ , where  $o(1)$  converges to 0 as  $w$  grows.*

**PROOF.** We show that there are constants  $c_1, c_2, c_3 > 0$ , depending only on  $k$ , such that  $\Pr[\text{Forced}(x, \pi)] \geq s_k + c_i - o(1)$  whenever  $x$  is privileged due to reason (i) in Definition 4.10.

The case of privileged variables of type (1), i.e., those having at least two critical clauses, has already been addressed in the full versions of [3]. However, for the sake of completeness we will also discuss this case. We will introduce some operations on labeled trees  $T$  that never



**Figure 2.** Attaching additional descendants to  $u$  will not increase  $\Pr[\text{Cut}_r(T)]$ .

increase  $\Pr[\text{Cut}_r(T)]$ . As a most simple example, suppose  $u$  is a node in  $T$  and not a safe leaf; form  $T'$  by adding a new child  $v$  to  $u$ . Then  $\Pr[\text{Cut}_r(T)] \geq \Pr[\text{Cut}_r(T')]$ , regardless of the label of  $v$ . This follows immediately from the definition of  $\text{Cut}_r$ .

This operation allows us to reduce case (3) to case (2). Indeed, suppose  $T_x$  has fewer than  $(k-1)^2$  nodes at depth 2. Let  $Y_1, \dots, Y_{k-1}$  be the children of the root of  $T_x$  and let  $y_1, \dots, y_{k-1}$  be their respective labels. By assumption, some child  $Y_i$  has at most  $k-2$  children. Create a new node  $Z$ , attach it as an additional child to  $Y_i$ , and give it label  $y_1$ . The resulting tree  $T'$  is a labeled tree, every node has at most  $k-1$  children, and label  $y_1$  occurs at depths 1 (at  $Y_1$ ) and at depth 2 (at  $Z$ ), so  $T'$  is of type (2).

Next, we (almost) reduce case (1) to case (2). Suppose  $x$  has two critical clauses,  $C = (x \vee \bar{y}_1, \dots, \bar{y}_{k-1})$  and  $D = (x \vee \bar{z}_1, \dots, \bar{z}_{k-1})$ . Note that  $k \leq |\{y_1, \dots, y_{k-1}, z_1, \dots, z_{k-1}\}| \leq 2(k-1)$ . Suppose for the moment that it is less than  $2(k-1)$ , i.e., some variable  $y_i$  also appears in  $D$ . Without loss of generality,  $y_1 = z_1$ . Also, the two clauses are distinct, so let us assume that  $y_{k-1}$  does not appear in  $D$ . We will construct a (non-canonical) critical clause tree  $T'_x$  for  $x$  that is of type (2). Use  $C$  as clause label for the root. Note that this creates  $k-1$  nodes  $Y_1, \dots, Y_{k-1}$  at depth 1 with labels  $y_1, \dots, y_{k-1}$ . The assignment label of  $Y_{k-1}$  is  $\alpha[y_{k-1} \mapsto 0]$ , which violates  $D$ ; here we use the fact that  $D$  does not contain  $y_{k-1}$ . Thus, we can use  $D$  as clause label of node  $Y_{k-1}$ , which in turn creates  $k-1$  nodes at depth 2, one of which has label  $z_1$ . Now recall that  $y_1 = z_1$  by assumption, so this label occurs once at depth 1 and somewhere at depth 2, and  $T_x$  is of type (2).

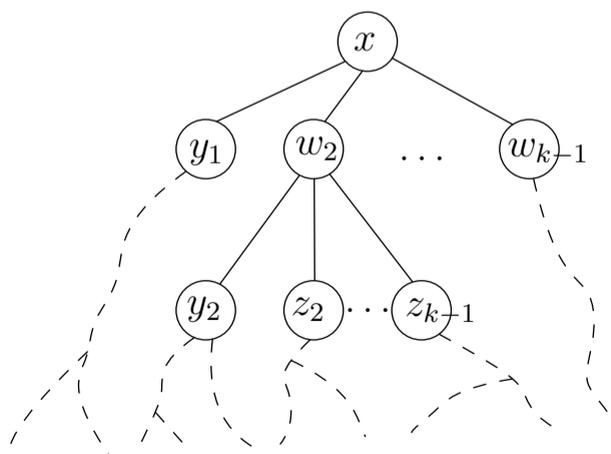
Summarizing, we are left with privileged variables of type (2) and those with two critical clauses  $C, D$  that share no variable besides  $x$ . Let us deal with type (2) first. We start with a proposition stating that assigning “fresh labels” to a node of  $T$  cannot increase  $\Pr[\text{Cut}_r(T)]$ .

This can be seen as an alternative proof of Lemma 7 in [8] that bypasses the FKG inequality for monotone Boolean functions (in fact, implicitly reproves it).

**PROPOSITION A.1.** *Let  $T$  be a labeled tree and suppose the values  $\{\pi(l)\}_{l \in L}$  are independent. Let  $u$  be a node in  $T$  with label  $z$ . Form a new tree  $T'$  by assigning  $u$  a fresh label  $z'$  and making  $\pi(z')$  follow the same distribution as  $\pi(z)$ , but independent of everything else. Then  $\Pr[\text{Cut}(T)] \geq \Pr[\text{Cut}(T')]$*

**PROOF.** We show that  $\Pr[\text{Cut}_r(T)] \geq \Pr[\text{Cut}_r(T')]$  for all  $r \in [0, 1]$ . Let  $\tau : L \setminus \{z, z'\} \rightarrow [0, 1]$  be a placement of all labels except  $z$  and  $z'$ . In fact, we claim that  $\Pr[\text{Cut}_r(T) \mid \tau] \geq \Pr[\text{Cut}_r(T') \mid \tau]$  holds for all  $\tau$ . Under the partial placement  $\tau$ , the event  $\text{Cut}_r(T')$  becomes some monotone Boolean function  $f(b, b')$  in the Boolean variables  $b := [\pi(z) < r]$  and  $b' := [\pi(z') < r]$ , and  $\text{Cut}_r(T)$  becomes  $f(b, b)$ . This holds since  $T$  can be obtained from  $T'$  by merging the labels  $z$  and  $z'$ . Now under our distribution on placements,  $\pi(z)$  and  $\pi(z')$  follow the same distribution and therefore  $\Pr[b = 1] = \Pr[b' = 1]$ . This means that  $\Pr[f(b, b') = 1] = \Pr[f(b, b) = 1]$  unless  $f$  depends on both variables; the only monotone functions depending on both  $b$  and  $b'$  are  $b \wedge b'$  and  $b \vee b'$ . If  $f(b, b') = b \wedge b'$  then  $\Pr[\text{Cut}_r(T') \mid \tau] = \Pr[b \wedge b'] \leq \Pr[b] = \Pr[\text{Cut}_r(T)]$  and our claim holds. Finally,  $f(b, b') = b \vee b'$  cannot hold: the set of nodes in  $T'$  with label  $z$  or  $z'$  form an antichain, by Point 2 of Definition 3.6. ■

Now let  $T$  be a labeled tree to which (2) applies, i.e., some variable  $y$  appears at depth 1 and 2 in  $T_x$ . Let  $Y_1, Y_2$  be those two nodes with label  $y$ . We apply the proposition to all nodes except  $Y_1$  and  $Y_2$ . Second, we add new children to nodes of depth less than  $h$  until all such nodes have exactly  $k - 1$  children, and all the  $(k - 1)^h$  nodes at depth  $h$  are safe leaves. Call this tree  $T$ . From the proposition and the discussion above, it follows that  $\Pr[\text{Cut}_r(T_x)] \geq \Pr[\text{Cut}_r(T)]$ . In a last step, give a fresh label  $y_1$  to  $Y_1$  and  $y_2$  to  $Y_2$ , and call the resulting tree  $T'$ . In  $T'$ , all nodes have distinct labels, and we know what  $\Pr[\text{Cut}_r(T')]$  is: it is  $Q_r^{(k-1)} - o(1)$ .<sup>8</sup> We also know that  $\Pr[\text{Cut}_r(T)] \geq \Pr[\text{Cut}_r(T')]$  by the above proposition. Now, however, we have to take a closer look at the proof of the proposition since we want to show that  $\Pr[\text{Cut}_r(T)]$  is significantly larger than  $\Pr[\text{Cut}_r(T')]$ .



<sup>8</sup> We will write  $Q_r$  instead of  $Q_r^{(k-1)}$  from now on since  $k$  is understood.

The tree  $T'$ : all labels are distinct.

In  $T'$ , we denote the node with label  $w_2$  by  $W_2$ ; that with label  $z_2$  by  $Z_2$ , and so on. Since  $T'$  and  $T$  has the same node set, we use this notation for the nodes in  $T$ , too. Furthermore, for a node  $u$ , we denote the subtree of  $T$  (or  $T'$ ) rooted at node  $u$  by  $T_u$  (or  $T'_u$ ). As in the proof of the proposition, we fix some partial placement  $\tau : L \setminus \{y_1, y_2, y\} \rightarrow [0, 1]$ . As we have seen in the proof,  $\Pr[\text{Cut}_r(T) \mid \tau] \geq \Pr[\text{Cut}_r(T') \mid \tau]$  holds for every such  $\tau$ . Call  $\tau$  *good* if the following holds: (1)  $\pi(w_2) \geq r$ ; (2)  $\pi(z_2), \dots, \pi(z_{k-1}), \pi(w_3), \dots, \pi(w_{k-1}) < r$ ; (3)  $\neg \text{Cut}_r(T_{Y_1})$  and  $\neg \text{Cut}_r(T_{Y_2})$ . The events described in (1–3) are independent; those in (1) and (2) happen with probability exactly  $(1-r)r^{2k-5}$ . Those in (3) happen with probability at least  $(1-Q_r)^2$ .

Under a good  $\tau$ , the  $\text{Cut}_r(T')$  becomes  $[\pi(y_1) < r \wedge \pi(y_2) < r]$  and has probability  $r^2$ , and  $\text{Cut}_r(T)$  becomes  $[\pi(y) < r]$ , which has probability  $r$ . Therefore,

$$\begin{aligned} \Pr[\text{Cut}_r(T)] - \Pr[\text{Cut}_r(T')] &\geq \Pr[\tau \text{ is good}] \cdot (r - r^2) \\ &\geq (1-r)r^{2k-5}(1-Q_r)^2(r-r^2) = (1-r)^2r^{2k-4}(1-Q_r)^2. \end{aligned}$$

Putting everything together and integrating over  $r$ , we conclude that

$$\Pr[\text{Cut}(T_x)] \geq s_k - o(1) + \int_0^1 (1-r)^2r^{2k-4}(1-Q_r)^2 dr.$$

It is clear that the integral is some positive constant depending solely on  $k$ .

We are left with the case that  $x$  has two critical clauses  $C = (x \vee \bar{y}_1, \dots, \bar{y}_{k-1})$  and  $D = (x \vee \bar{z}_1, \dots, \bar{z}_{k-1})$ , and  $y_i \neq z_j$  for all  $1 \leq i, j \leq k-1$ . It is clear that  $x$  is forced if all  $y_i$  come before  $x$  or all  $z_i$  come before  $x$ . Therefore,

$$\begin{aligned} \Pr[\text{Forced}(x, \pi) \mid \pi(x) = r] &\geq r^{k-1} + r^{k-1} - \Pr[\text{all } y_i \text{ and all } z_i \text{ come before } x] \\ &\geq 2r^{k-1} - r^{2k-2}. \end{aligned}$$

On the other hand, by focusing solely on the canonical critical clause tree of  $x$ , we can apply Lemma 3.11 and conclude that

$$\Pr[\text{Forced}(x, \pi) \mid \pi(x) = r] \geq Q_r - o(1),$$

(we write  $Q_r$  instead of  $Q_r^{(k)}$  since  $k$  is understood), and therefore

$$\begin{aligned} \Pr[\text{Forced}(x, \pi) = 1] &\geq \int_0^1 \max(2r^{k-1} - r^{2k-2}, Q_r) dr - o(1) \\ &= s_k - o(1) + \int_0^1 \max(0, 2r^{k-1} - r^{2k-2} - Q_r) dr. \end{aligned}$$

It remains to show that the latter term is positive for a substantial range of  $r \in [0, 1]$ . We claim that if  $r$  is sufficiently small,  $Q_r$  is only a tiny factor larger than  $r^{k-1}$ . Indeed, From Proposition 3.10, we know that  $Q_r \leq e r^{k-1}$ , thus  $P_r = r \vee Q_r = r + (1-r)e r^{k-1} = r(1+e(1-r)r^{k-2})$

and in turn  $Q_r = P_r^{k-1} \leq (r + e r^{k-1})^{k-1} = r^{k-1} (1 + e r^{k-2})^{k-1} < r^{k-1} e^{e(k-1)r^{k-2}}$ . We check that  $e^{e(k-1)r^{k-2}} \leq 1.5$  for all  $k \geq 3$  and  $r \leq 1/16$ . Therefore,

$$\begin{aligned} \int_0^1 \max\left(0, 2r^{k-1} - r^{2k-2} - Q_r\right) dr &\leq \int_0^{1/16} \left(2r^{k-1} - r^{2k-2} - 1.5r^{k-1}\right) dr \\ &= \int_0^{1/16} \left(\frac{1}{2}r^{k-1} - r^{2k-2}\right) dr \end{aligned}$$

and the latter is some positive constant  $c_1$  depending only on  $k$ . This concludes the proof of Lemma 4.11. ■