Non-deterministic branching programs with logarithmic repetition cannot efficiently compute small monotone CNFs

Oded Lachish and Igor Razgon
Department of Computer Science and Information Systems, Birkbeck, University of London
{oded.igor}@dcs.bbk.ac.uk

Abstract

In this paper we establish an exponential lower bound on the size of syntactic non-deterministic read $d$-times branching programs for $d \leq \log n/10^5$ computing a class of monotone CNFs with a linear number of clauses. This result provides the first separation of classes NP and co-NP for syntactic branching programs with a logarithmic repetition and the first separation of syntactic non-deterministic branching programs with a logarithmic repetition from small monotone CNFs.

1 Introduction

We study here the complexity of syntactic non-deterministic branching programs (NBPs) with bounded repetition $d$ (read-$d$-times NBPs or simply $d$-NBPs). We prove an exponential lower bound for $\log n/10^5$-NBPs computing a class of functions that are expressible as monotone CNFs with a linear number of clauses. As the complement of a small CNF is a small DNF having a linear size presentation as 1-NBP, our result separates NP and co-NP for branching programs with $\log n/10^5$ repetition.

The previous record for separating classes NP and co-NP for syntactic NBPs was for $d$-NBPs with $d$ up to $\Theta(\log n/\log \log n)$. It was achieved in 1994, by Jukna [5].

To the best of our knowledge, the paper of Jukna and Schnitger [8] (Theorem 3.3.) is the only result that separates small monotone CNFs from $d$-NBPs. It does so with $d = o(\log n/\log \log n)$. We note that they do not state this explicitly, but we can easily get to this conclusion because of the following. Their lower bound is established for a function on graphs that is true if and only if the given graph does not contain a 4-clique. This function can be presented as a CNF with 6 literals per clause, all of them negative. A simple reduction shows that if all these negative literals are replaced by positive ones then the lower bound retains for the obtained monotone CNF. We are also not aware of other results separating $d$-NBPs, with $d > 1$, from monotone functions or from non-monotone CNFs.

Prior to these two results, in 1993, Borodin, Razborov and Smolensky proved an exponential lower bound for $d$-NBPs for $d$ up to $\Theta(\log n)$ computing a class of functions
that is not known to be monotone or to have small \(d\)-NBP for its complement. The results mentioned created two natural open questions: (i) is there \(d = \Theta(\log n)\) such that classes \(\text{NP}\) and \(\text{co-NP}\) are different for syntactic branching programs with repetition up to \(d\); and (ii) is it possible to separate \(d\)-NBP for some \(d = \Theta(\log n)\) from monotone functions or from \(\text{CNFs}\) or from monotone \(\text{CNFs}\)? Our result resolves these problems by providing a positive answers to both.

As mentioned above, besides [8], we are not aware of other super-polynomial lower bounds for \(d\)-NBP with \(d > 1\) on \(\text{CNFs}\) nor on monotone functions. However, such lower bounds exist for read-once branching programs. For example, an exponential lower bound for deterministic read-once branching programs on small monotone \(\text{CNFs}\) was presented in [4]. In [10], the second author showed a parametrized \(n^{\Omega(k)}\) for 1-NBP lower bound computing monotone 2-\(\text{CNFs}\) whose underlying graph has tree-width at most \(k\). This lower bound can easily be converted into an exponential one by taking the underlying graph to be an expander.

Currently, there are no super-polynomial lower bounds for syntactic branching programs with super-logarithmic repetition, even deterministic ones. However, there is an exponential lower bound for oblivious branching programs with \(o(\log^2 n)\) repetition [2] (presented in a more general form of \(o(n \log^2 n)\) trade-off). We note that oblivious branching programs are a special case of \(\text{NBP}\)s with the same order of variable occurrences along every source-sink path.

Researchers have also investigated branching programs for functions with non-Boolean domains of variables. One purpose of considering this framework for \(\text{NBP}\)s is to obtain lower bounds for semantic rather than syntactic \(\text{NBP}s\) that bound the number of variable occurrences on consistent paths only. Such lower bounds are known for functions with non-Boolean domains [6], however, in the Boolean case, super-polynomial lower bounds have not yet been established even for semantic 1-\(\text{NBPs}\).

For a more detailed survey of research on branching programs, we refer the reader to the monographs of Jukna [7] and Wegener [11].

**Overview of result:** We construct a dedicated family of small monotone formulae. We then prove that for every member \(\varphi\) of this family every \(\log n/10^5\)-NBP \(Z\), where \(n\) is the number of variables in \(\varphi\), computing \(\varphi\) has size exponential in \(n\).

From a birds eye view the proof proceeds as follows. Given an \(\log n/10^5\)-NBP \(Z\) computing a member \(\varphi\) of the family, we show that the properties of \(\varphi\) imply that every computational path \(P\) of \(Z\) contains a special set of \(O(\log n)\) vertices, we call a determining set. Then we show that \(Z\) has an exponential number of distinct determining sets. This concludes our result since there can be this many distinct determining sets only if the size of \(Z\) is exponential in \(n\). We next describe how this is achieved.

We first reduce the problem of proving exponential lower bounds for \(d\)-NBPs to the special case of proving exponential lower bounds for uniform \(d\)-NBP, which are NBPs such that along every one of their source-sink paths, every variable occurs exactly \(k\) times. Then we look at an arbitrary computational path \(P\) in \(Z\) as a string where the variables on the path are the letters. We use a structural theorem of Alon and Maass (Theorem 1.1. [11]), to deduce that there exists a set of \(O(\log n)\) indices that split the string into a set of consecutive substrings (intervals) each between two consecutive
indices of this set. This set of intervals has a special property that there exists two large sets of specific letters (variables), where by large we mean \( n^\alpha \) for some \( \alpha > 0 \), such that the first, which we call the *odd set*, has letters that only appear in odd intervals, where the "odd" is according to their order, and the second, which we call the *even set*, has letters that only appear in even intervals. The set of vertices on the path \( P \) corresponding to the set of indices is the determining set \( X \). We next explain how we use the special property of the intervals.

In order to utilize the properties of the odd and even sets we defined the dedicated family of CNFs based on a special type of graphs we call pseudoexpanders. This construction ensures that there is a large matching \( M_X \) between members of the odd set and the even set, with each pair in the matching corresponding to a distinct clause in \( \varphi \). Based on this matching we define a formula \( \psi_X \), which is satisfied by the assignment of the computational path \( P \). We construct a probability space on the set of satisfying assignment of \( \varphi \), such that the probability that a formula \( \psi_X \) is satisfied is exponentially small in the size of \( M_X \) and, since \( M_X \) is large, is, in fact, exponentially small in the number of variables of \( \varphi \). Clearly the conjunction, over all the computational paths of \( Z \), of formulae \( \psi_X \) corresponding to the paths, is satisfied with probability 1. Thus, there must be exponentially many formulae \( \psi_X \) and in turn exponentially many distinct determining sets.

We remark, regarding the construction of CNFs, that we first prove the existence of the considered class of CNFs \( \varphi \) non-constructively, using a probabilistic method. In the Appendix, we show that there is a deterministic polynomial time procedure constructing a class of small monotone CNFs \( \varphi^* \) such that \( \varphi \) can be obtained from \( \varphi^* \) by a partial assignment to the latter. It can be observed that the exponential lower bound on \( \log n/10^5 \)-NBPs retains for \( \varphi^* \). That is, our result holds for a set of constructively created CNFs.

**Structure of the paper.** The rest of the paper is structured as follows. Section 2 contains all the preliminary definitions and notations. Section 3 proves the main result with the proofs of the auxiliary results postponed to the later sections. In particular, Section 4 establishes existence of pseudoexpanders, Section 5 proves that each computational path of the considered branching programs contains a determining set and Section 6 proves the exponentially low probability claim described above. The paper also has an appendix consisting of four sections. In the Section A we provide a poly-size simulation of \( d \)-NBPs by uniform \( d \)-NBPs. In Section B, we show that the exponential lower bound for \( O \log n/10^5 \)-NBPs applies to ‘constructively created’ CNFs (proving the theorem at the end of Section 3). In Section C we prove auxiliary lemmas for Section 5. In Section D we prove the validity of the definition of probability space used in Section 6.

## 2 Preliminaries

**Sets of literals as assignments.** In this paper by a *set of literals* we mean one that does not contain both an occurrence of a variable and its negation. For a set \( S \) of literals we denote by \( \text{Vars}(S) \) the set of variables whose literals occur in \( S \) (the \( \text{Vars} \)
notation naturally generalizes to CNFs, Boolean functions, and branching programs. A set \( S \) of literals represents the truth assignment to \( \text{Vars}(S) \) where variables occurring positively in \( S \) (i.e., whose literals in \( S \) are positive) are assigned with \( \text{true} \) and the variables occurring negatively are assigned with \( \text{false} \). For example, the assignment \( \{x_1 \leftarrow \text{true}, x_2 \leftarrow \text{true}, x_3 \leftarrow \text{false}\} \) to variables \( x_1, x_2, x_3 \) is represented as \( \{x_1, x_2, \neg x_3\} \).

**Satisfying assignments.** Let \( \varphi \) be a CNF. A set \( S \) of literals satisfies a clause \( C \) of \( \varphi \) if at least one literal of \( C \) belongs to \( S \). If all clauses of \( \varphi \) are satisfied by \( S \) then \( S \) satisfies \( \varphi \). If, in addition, \( \text{Vars}(\varphi) = \text{Vars}(S) \) then we say that \( S \) is a satisfying assignment of \( \varphi \). The notion of a satisfying assignment naturally extends to Boolean functions \( F \) meaning a truth assignment to \( \text{Vars}(F) \) on which \( F \) is true.

**Definition 1 (Non-deterministic branching programs and related notions).**

- A Non-deterministic branching program (NBP) \( Z \) is a directed acyclic multigraph (DAG) with one source and one sink with some edges labelled with literals of variables. \( \text{Vars}(Z) \) denotes the set of variables whose literals occur on the edges of \( Z \).

- A directed source-sink path \( P \) of \( Z \) is a computational path of \( Z \) if opposite literals of the same variable do not occur as labels of edges of \( P \). We denote by \( A(P) \) the set of literals labelling the edges of \( P \). We call \( A(P) \) the assignment carried by \( P \).

- The function \( F_Z \) computed by \( Z \) is a function on \( \text{Vars}(Z) \). A set of literals over \( \text{Vars}(Z) \) is a satisfying assignment if and only if there is a computational path \( P \) of \( Z \) such that \( A(P) \subseteq S \).

- The size of \( Z \), denoted by \(|Z|\) is the number of its vertices.

**Definition 2 (Read \( d \) times NBP).** A syntactic read-\( d \)-times NBP (\( d \)-NBP) is an NBP in which each variable occurs at most \( d \) times on each source-think path. A \( d \)-NBP is uniform if each variable occurs exactly \( d \) times on each source-sink path.

**Lemma 1.** For every \( d \)-NBP \( Z \), there exist a uniform \( d \)-NBP computing \( F_Z \) and having size \( O(|Z|^4d) \)

The proof of Lemma 1 appears in the Appendix.

It follows from Lemma 1 that an exponential lower bound obtained for a uniform \( d \)-NBP applies to a \( d \)-NBP in general. Therefore, in the rest of the paper (except, obviously, for the proof of Lemma 1), we assume (without stating it explicitly) that all the considered \( d \)-NBPs are uniform. For the purpose of establishing the lower bound, the advantage of considering uniform NBPs is that for any two paths of an NBP having the same initial and final vertices, the sets of variables labelling these paths are the same, as proved in the next lemma.

**Lemma 2.** Let \( Z \) be a \( d \)-NBP. Let \( P_1 \) and \( P_2 \) be two paths of \( Z \) having the same initial and final vertices. Then a variable occurs on \( P_1 \) if and only if it occurs on \( P_2 \).
Proof. Let \( u \) and \( v \) be the starting and ending vertices of \( P_1 \) and \( P_2 \). Denote by \( \text{start} \) and \( \text{end} \) the source and sink vertices of \( Z \), respectively. Let \( P_0 \) be a \( \text{start} \rightarrow u \) path of \( Z \) and \( P_3 \) be a \( v \rightarrow \text{end} \) path of \( Z \). Due to the acyclicity of \( Z \), both \( Q_1 = P_0 + P_1 + P_3 \) (the concatenation of \( P_0, P_1, P_3 \)) and \( Q_2 = P_0 + P_2 + P_3 \) are source-sink paths of \( Z \). Suppose there is a variable \( x \) occurring on \( P_1 \) but not on \( P_2 \). Then, to supply \( d \) occurrences of \( x \) on \( Q_2 \) required because \( Z \) is uniform, \( x \) occurs \( d \) times in \( P_1 \cup P_3 \). It follows that on \( Q_1 \), \( x \) occurs at least \( d + 1 \) times in contradiction to the definition of a \( d \)-NBP.

Now, we introduce terminology related to graphs and trees and we will need for definition of the CNF class.

**Definition 3.** A rooted tree is called extended if none of its leaves has a sibling.

**Definition 4 (Binary tree based graphs (BTB)).** A graph \( H \) is a binary tree based graph if:

1. \( H \) is an edge-disjoint union of extended trees \( T(H) = \{T_1, \ldots, T_m\} \) with roots \( t_1, \ldots, t_m \), which we call the root vertices of \( H \) and denote by \( \text{Roots}(H) \). The set of all leaves of \( T_1, \ldots, T_m \) is denoted by \( \text{Leaves}(H) \).

2. Each vertex of \( u \in \text{Leaves}(H) \) is a leaf of exactly two trees of \( T(H) \).

3. Any two trees of \( T(H) \) have at most one vertex in common. This common vertex is a leaf in both trees.

In what follows we denote \(|V(H)|\) by \( n \) and \(|\text{Roots}(H)|\) by \( m \).

**Definition 5 (Adjacency of trees in BTB).** Let \( H \in \text{BTB} \) with \( T(H) = \{T_1, \ldots, T_m\} \) and \( \text{Roots}(H) = \{t_1, \ldots, t_m\} \) where each \( t_i \) is the root of \( T_i \).

- \( T_i \) and \( T_j \) are adjacent if and only if they share a leaf in common. The common leaf of \( T_i \) and \( T_j \) is denoted by \( \ell_{i,j} \).
- The unique path connecting roots of \( T_i \) and \( T_j \) in \( T_i \cup T_j \) is denoted by \( P_{i,j} \).
- The path connecting \( t_i \) and \( \ell_{i,j} \) is in \( T_i \) is denoted by \( P_{i \rightarrow j}^{1/2} \) and the path connecting \( t_j \) and \( \ell_{i,j} \) is in \( T_j \) is denoted by \( P_{j \rightarrow i}^{1/2} \). (The \( 1/2 \) in the subscript says that these paths are ‘halves’ of \( P_{i,j} \).)

**Definition 6 (Pseudodegrees and pseudomatchings).** Let \( H \in \text{BTB} \) and \( U, V \) be two disjoint subsets of \( \text{Roots}(H) \).

- A pseudodegree of \( H \) is a pair \( \{t_i, t_j\} \) of roots such that \( T_i \) and \( T_j \) are adjacent. The pseudodegree of \( t_i \) is the number of pseudodegrees containing \( t_i \).
- A pseudomatching of \( H \) is a set of pseudodegrees of \( H \) that do not share common ends.
- A pseudomatching \( M \) is between \( U \) and \( V \) if for each \( e \in M \), \(|e \cap U| = |e \cap V| = 1\).
For a pseudomatching $M$, we denote $\bigcup_{e \in M} e$ by $\hat{M}$.

**Definition 7 (Underlying graph for a graph in BTB).** The underlying graph $U(H)$ is a graph whose vertices are $\text{Roots}(H)$ and two vertices in $U(H)$ are adjacent if and only if their corresponding trees are adjacent in $H$.

### 3 The main result

**Strategy of the proof.** Fix $Z$ to be a $d$-NBP that computes a $\varphi$. The goal is to show that $Z$ is exponentially large in the number $n$ of variables of $\varphi$. To do so we prove that every computational path $P$ of $Z$ has a set of variables $X$, of size logarithmic in $n$ that has specific properties which enable us to prove the following: $Z$ contains an exponential in $n$ number of such distinct sets. Since the size of these sets is logarithmic in $n$, the lower bound on the size of $Z$ easily follows. The specific properties of $X$ are listed next.

1. There is a special formula $\psi_X$ defined on the variables of $\varphi$ such that for any path $Q$ containing $X$, the assignment $A(Q)$ carried by $Q$ satisfies $\psi_X$.

2. There is a probability space on the set of satisfying assignments of $\varphi$, so that the probability that an assignment drawn from the space satisfies $\psi_X$ is exponentially small in $n$.

Now, since every computational path contains a set $X$, every satisfying assignment $S$ of $\varphi$ satisfies some $\psi_X$. This, together with the last statement above, implies that the number of formulae $\psi_X$ must be exponentially large in $n$, and hence also the number of sets $X$.

The CNF for which we prove the lower bound is based on a special class of graphs we call pseudoeexpanders and defined next and afterwards we define the actual CNFS. Recall that $m$ denotes the number of roots of $H \in \text{BTB}$.

**Definition 8 (Pseudoeexpanders).** $H \in \text{BTB}$ is a pseudoeexpander if the following two conditions hold.

1. Small height property. The height (the largest number of vertices of a root-leaf path) of each $T_i$ is at most $(\log m)/4.9 + 3$.

2. Large pseudomatching property. For any two disjoint subsets $U, V$ of $\text{Roots}(H)$ of size at least $m^{0.999}$ each, there is a pseudomatching of $H$ between $U$ and $V$ of size at least $m^{0.999}/3$.

The the class of all pseudoeexpanders is denoted by $\text{PE}$.

**Definition 9.** For $H$, $T(H)$ and $\text{Roots}(H)$ as in Definition 6 a CNF $\phi_H$ is defined as follows.

The variables of $\phi_H$ are the vertices of $H$. The clauses $C_{i,j}$ of $\phi_H$ are in bijective correspondence with the pseudoeedges of $H$ and the literals of $C_{i,j}$ are $V(P_{i,j})$.

The class $\{\phi_H | H \in \text{PE}\}$ is denoted by $\Phi(\text{PE})$. 
We are going to prove that $O(\log n)$-NBPds require an exponential size to compute the class $\Phi(\text{PE})$. Note that Definition 8 does not obviously imply that even one pseudoexpander exists, while, for the purpose of the proof we need an infinite number of them. This is established in the following theorem.

**Theorem 1.** For each sufficiently large $m$ there is a pseudoexpander with $m$ roots and max-degree at most $m^{1/4.9}$. That is, there are infinitely many $n$ for which there is a pseudoexpander with $n$ vertices and the pseudodegree of each root being at most the number of roots to the power of $1/4.9$.

Theorem 1 is proved in Section 4 using probabilistic method.

**Corollary 1.** The elements of $\Phi(\text{PE})$ are monotone CNFs with linear number of clauses and there is an unbounded number of them. In particular, there is an unbounded number of $\phi_H$ where $H$ is a pseudoexpander with $m$ roots such that the pseudodegree of each root is at most $m^{1/4.9}$.

We next define the formulas that will be associated with the computational paths as described in the overall intuition.

**Definition 10 (Matching CNF).** Let $M$ be a pseudomatching of $H \in \text{BTB}$. A matching CNF w.r.t. $M$ consists of clauses $C_{i,j}^{1/2}$ for each $(t_i, t_j) \in M$, where $C_{i,j}^{1/2} \in \{V(t_i^{1/2}), V(t_j^{1/2})\}$.

The set of all matching CNFs w.r.t. $M$ is denoted by $\text{CNF}(M)$.

**Definition 11 (Matching OR-CNF).** An matching OR-CNF w.r.t. a pseudomatching $M$ of $H \in \text{BTB}$ has the form $\text{Conj}(S_1) \land \phi_1 \lor \cdots \lor \text{Conj}(S_q) \land \phi_q$, where $\{S_1, \ldots, S_q\}$ is the family of all sets of literals over $V(H) \setminus \bigcup M$, $\text{Conj}(S_i)$ is the conjunction of literals of $S_i$ and $\phi_i \in \text{CNF}(M)$ for each $1 \leq i \leq q$. The set of all matching OR-CNFs w.r.t. $M$ is denoted by $\text{ORCNF}(M)$.

We now define the special set of variables that appears in every computational path. Recall first that $n$ and $m$ denote the number of vertices of roots of $H$, respectively.

**Definition 12 (Determining set).** Let $Z$ be an NBP implementing $\phi_H$ for $H \in \text{BTB}$. A set $X \subseteq V(Z)$ is called an $(a, b)$-determining set if $|X| \leq a$ and there are a pseudomatching $M$ of $H$ of size at least $b$ and $\psi_X \in \text{ORCNF}(M)$ such that for any computational path $Q$ of $Z$ containing $X$, $A(Q)$ satisfies $\psi_X$. We call $\psi_X$ a $b$-witnessing formula for $X$.

**Theorem 2.** Let $Z$ be an $d$-NBP implementing $\phi(H)$ for $H \in \text{PE}$ with $d \leq \log n / 10^5$. Then each computational path of $Z$ contains a $(\log n / 4000, m^{0.999/3})$-determining set.

Theorem 2 is proved in Section 5. The following theorem is based on the low probability of satisfaction of $\psi_X$ for a $(\log n / 4000, m^{0.999/3})$-determining set $X$ as described in the strategy of the proof.

**Theorem 3.** There are constants $\alpha > 1$ and $0 < \gamma < 1$ such that the following holds. Let $H$ be a pseudoexpander with a sufficiently large number of vertices and let $\psi_1, \ldots, \psi_q$ be OR-CNFs for which the following properties hold.
For each $\psi_i$, there is a pseudomatching $M_i$ of $H$ of size at least $m^{0.999}/3$ such that $\psi_i \in \text{ORCNF}(M_i)$.

Every satisfying assignment of $\phi_H$ satisfies $\psi_1, \ldots, \psi_q$.

Then $q \geq \alpha^n \gamma$.

Theorem 3 is proved in Section 6.

**Theorem 4.** There exist constants $\alpha > 1$ and $0 < \beta < 1$ such that the following holds: if $H$ is a sufficiently large pseudoexpander and $Z$ a $d$-NBP computing $\phi_H$ with $d \leq \log n/10^5$, then $|Z| \geq \alpha^n \gamma$.

**Proof.** By Theorem 2, each computational path $P$ contains $X(P)$ which is a $(\log n/4000, m^{0.999}/3)$-determining set. Let $\{X_1, \ldots, X_q\}$ be all such sets $X(P)$ and let $\psi_1, \ldots, \psi_q$ be their respective $m^{0.999}/3$-witnessing formulas. That is there are pseudomatchings $M_1, \ldots, M_q$ of $H$ of size at least $m^{0.999}/3$ each such that $\psi_i \in \text{ORCNF}(M_i)$ for $1 \leq i \leq q$.

Next, we show that the disjunction of $\psi_i$ is satisfied by every satisfying assignment of $\phi_H$. Indeed, let $S$ be a satisfying assignment of $\phi_H$. Then there is a computational path $P$ with $S = A(P)$. By the definition of $\{X_1, \ldots, X_q\}$, there is some $X_i$ contained in $P$. It follows from Definition 12 that $S$ satisfies $\psi_i$ and hence the disjunction of $\psi_1, \ldots, \psi_q$.

Thus both conditions of Theorem 3 are satisfied and it follows that $q \geq \alpha^n \gamma$, for some $\alpha > 1$ and $0 < \gamma < 1$. As each $X_i$ is of size at most $\log n/4000$, $q \leq |Z| \log n$ and thus $|Z|^\log n \geq \alpha^n \gamma$. Therefore, $|Z| \geq \alpha^n \beta$ where $\beta = 0.99 \times \gamma$. ■

The class of CNFs considered in Theorem 4 is non-constructive, its existence is established using the probabilistic method. In the following theorem, we prove existence of a poly-time deterministic procedure creating a class of monotone small CNFs such that for each sufficiently large CNF $\varphi$ of this class there is a partial assignment transforming $\varphi$ into $\phi_H$ for some pseudoexpander $H$ of essentially the same size. Thus the lower bound of Theorem 4 applies to this constructively created class of CNFs.

**Theorem 5.** There is a class of CNFs produced by a polynomial time deterministic procedure that takes an exponential size to compute by $d$-NBP for $d \leq \log n/10^5$.

The proof of Theorem 5 appears in the Appendix.

### 4 Proof of Theorem 1

In order to prove Theorem 1 we introduce a new type of graphs we call *pre-pseudoexpander*, which we define next. Afterwards we show that for each sufficiently large $m$ there is a pre-pseudoexpander with $m$ vertices. Then we prove Theorem 1 by showing that for a sufficiently large $m$ there is a pre-pseudoexpander with $m$ vertices that is the underlying graph of pseudoexpander with $m$ roots.

**Definition 13.** [Pre-pseudoexpander] An $m$ vertex graph $G = (V, E)$ is a pre-pseudoexpander if it satisfies the following:
1. the maximal degree of a vertex in $G$ does not exceed $m^{1/4.9}$, and

2. for every disjoint $V_1, V_2 \subseteq V$ such that $|V_1|, |V_2| \geq n^{0.999}$, there exists a matching in $G$, of size at least $m^{0.999}/2$, between $V_1$ and $V_2$.

The proof that for each sufficiently large $m$ there is a pre-pseudoexpander with $m$ vertices consists of two parts: in the first part, we show that the existence of a pre-pseudoexpander follows from the existence of a graph with the small degree property (Condition[1]) and a relaxation of the second property (Condition[2]) from matching to single edges; and in the second part we use the probabilistic method to prove that such graphs with $m$ vertices exist for each sufficiently large $m$.

**Lemma 3.** Let $G = (V, E)$ be a graph of $m$ vertices. If for every disjoint $U_1, U_2 \subseteq V$ such that $|U_1|, |U_2| \geq m^{0.999}/2$, there exists an edge $\{u, v\}$ such that $u \in U_1$ and $v \in U_2$, then $G$ satisfies Condition[2] of the definition of a pre-pseudoexpander.

**Proof.** Assume for the sake of contradiction that there exist disjoint subsets $U_1, U_2 \subseteq V$, each of size at least $m^{0.999}/2$, such that every matching between them is of size less than $m^{0.999}/2$. Let $W$ be the edges of a maximum matching between $U_1$ and $U_2$. Since the number of edges in $W$ is less than $m^{0.999}/2$, $|U_i \setminus V(W)| > m^{0.999}/2$ for $i \in \{1, 2\}$. Consequently, there is an edge $\{u, v\}$ such that $u \in U_1 \setminus V(W)$ and $v \in U_2 \setminus V(W)$. Thus, $W \cup \{u, v\}$ is also a matching between $U_1$ and $U_2$. Since obviously, $W \cup \{u, v\}$ is larger than $W$ we get a contradiction to $W$ being a maximum matching between $U_1$ and $U_2$.  

**Lemma 4.** There exists a constant $c$ such that, for every $m$ larger than $c$, there exists a $m$ vertex graph $G = (V, E)$ that satisfies the following two conditions:

1. the maximal degree of a vertex in $G$ does not exceed $m^{1/4.9}$, and

2. for every disjoint $U_1, U_2 \subseteq V$ such that $|U_1|, |U_2| \geq m^{0.999}/2$, there exists an edge $\{u, v\}$ such that $u \in U_1$ and $v \in U_2$.

**Proof.** We construct a random graph and prove that with strictly positive probability it has the property that, for every pair of subsets $U_1, U_2 \subseteq V$ such that $|U_1|, |U_2| = \lceil m^{0.999}/2 \rceil$, there exists an edge $\{u, v\}$ such that $u \in U_1$ and $v \in U_2$. Thus, by the probabilistic method a graph with such a property exists. The statement of the lemma follows, because every set of size at least $m^{0.999}/2$ is a superset of a set of size exactly $\lceil m^{0.999}/2 \rceil$.

Let $G = (V, E)$ be a random graph of $n$ vertices such that, for every distinct $u, v \in V$, the edge $\{u, v\}$ is in $E$ independently with probability $n^{-4/5}$. Let $|U_1|, |U_2| = \lceil m^{0.999}/2 \rceil$.

There are $|U_1| \times |U_2| \geq m^{1.998}/9$ pairs $\{u, v\}$ such that $u \in U_1$ and $v \in U_2$ and hence, with probability at least

$$(1 - n^{-4/5})m^{1.998}/9 \leq e^{-n^{-4/5} \times m^{1.998}/9} = e^{-m^{1.198}/9} \leq e^{-m^{1.197}},$$

there does not exist a single edge in $E$ between a vertex in $U_1$ and a vertex in $U_2$. The number of pairs of disjoint subsets of $V$ each of size exactly $\lceil m^{0.999}/2 \rceil$ is at most:

$$\left( \frac{m}{m^{0.999}} \right) m^{2m^{0.999}} \leq m^{2m^{0.999}} \times 2m^{0.999} \leq 2 \log m \times 2m^{0.999} \times 2m^{0.999} \leq 2m^{0.999} \leq m^{0.999} \leq e^{m^{0.999}}.$$
Thus, by the union bound, with probability at least \(1 - e^{m^{0.999}} \cdot e^{-m^{1.197}} > 2/3\), for every pair of subsets \(U_1, U_2 \subseteq V\) such that \(|U_1|, |U_2| = \lfloor m^{0.999}/2 \rfloor\), there exists an edge \(\{u, v\}\) such that \(u \in U_1\) and \(v \in U_2\).

Let \(v\) be an arbitrary vertex in \(G\). By the Chernoff’s bound, the probability that the degree of \(v\) is greater than \(3n^{1/5}\) is at most \(2e^{-m^{1/5}}\). Thus, by the union bound, with probability at least \(1 - m \cdot 2e^{-m^{1/5}} > 2/3\), for every \(u \in V\), the degree of \(U\) is at most \(3n^{1/5}\).

Finally, again by the union bound, with probability strictly greater than 1/3, the random graph we defined has both properties required in the statement of the lemma.

\(\square\)

**Proof of Theorem 1.** According to Lemma 4, for each sufficiently large \(m\), there is a graph \(G'_m\) with \(m\) vertices, max-degree \(m^{1/4.9}\) and with an edge between any two subsets of its vertices of size at least \(m^{0.999}/2\). According to Lemma 5, \(G'_m\) is in fact pre-pseudoexpander.

We are going to construct a graph \(H_m \in \mathcal{BTB}\) with \(m\) roots such that \(G'_m\) is isomorphic to \(U(H_m)\), the underlying graph of \(H_m\) and with the height of each tree of \(T(H_m)\) being at most \(\log m/4.9 + 3\). The construction is as follows. Enumerate the vertices of \(G'_m\) by 1, \ldots, \(m\). Fix extended trees \(T_1, \ldots, T_m\) such that the number of leaves of \(T_i\) is exactly as \(d_i\), the degree of vertex \(i\) in \(G'_m\) and the height of \(T'_i\) is at most \(\log m/4.9 + 3\). It is not hard to see that such trees exist. Indeed, take a complete binary tree \(T''\) with at most \(2 \cdot m^{1/4.9}\) leaves, fix an arbitrary set \(L\) of leaves \(d_i \leq m^{1/4.9}\) leaves and let \(T''\) be the rooted tree obtained by the union of all root-leaf paths of \(T''\) ending at \(L\). The height of \(T'_i\) is at most \(\log m/4.9 + 2\). By adding one leaf to each leaf of \(T''\) we obtain an extended tree with \(d_i\) leaves and height at most \(\log m/4.9 + 3\). Next for each \(1 \leq i \leq m\) mark the leaves of \(T_i\) with the numbers assigned to the neighbour of vertex \(i\) in \(G'_m\) (each leaf is assigned with a unique number). Then for those pairs \(\{i, j\}\) where \(T_i\) has a leaf \(\ell_1\) marked with \(i\) and \(T_j\) has a leaf \(\ell_2\) marked with \(j\), identify \(\ell_1\) and \(\ell_2\). Let \(H_m\) be the resulting graph. A direct inspection shows that \(U(H_m)\) is isomorphic to \(G'_m\).

We next show that \(H_m\) will satisfy the large pseudomatching property. Indeed, let \(U_1, U_2\) be two disjoint subsets of roots of \(H_m\) of size at least \(m^{0.999}\) each. By the property of pre-pseudoexpanders, there is a matching of size at least \(m^{0.999}/2 \geq m^{0.999}/3\) between \(U_1\) and \(U_2\). As in \(G'_m\) there is an edge between \(u \in U_1\) and \(v \in U_2\) if and only if in \(H_m\) there is a pseudoedge between these vertices in \(H_m\), there is a pseudomatching of size at least \(m^{0.999}/3\) between \(U_1\) and \(U_2\) in \(H_m\). We conclude that \(H_m\) is a pseudoexpander with maximal pseudodegree at most \(m^{1/4.9}\), thus implying the theorem.

\(\square\)

5 Proof of Theorem 2

In this section, we prove that each computational path of a \(d\)-NBP \(\mathcal{Z}\) with \(d \leq \log n/10^5\) computing \(\phi_H\) for \(H \in \mathcal{PE}\), contains a \((\log n/4000, m^{0.999}/3)\)-determining set \(X\).

We first show, using the structural theorem of Alon and Maass (Theorem 1.1. of [1]), that on each computational path \(P\), there is a set \(X\) of size at most \(\log n/4000\) that
separates two large disjoint sets $Y_1$ and $Y_2$ of roots of $H$ in the following sense. We call $Y_1$ the odd set and $Y_2$ the even set. The set $X$ naturally partitions $P$ into subpaths enumerated along $P$ so that the elements of the $Y_1$ occur as labels only on edges of odd numbered subpaths and elements of $Y_2$ occur only on even numbered subpaths. We then show that $X$ is in fact a $(\log n/4000, m^{0.999}/3)$-determining set.

That is, we show the existence of a formula $\psi_X \in \text{ORCNF}(M)$, where $M$ is a pseudomatching of size at least $m^{0.999}/3$ such that every assignment carried by a computational path containing $X$ satisfies $\psi_X$.

We fix a pseudomatching $M$ between $Y_1$ and $Y_2$ such that $|M| \geq m^{0.999}/3$. The existence of $M$ is guaranteed by the large pseudomatching property of pseudoexpanders. We consider an arbitrary assignment $S$ to all the variables of $\phi_H$ except $\bigcup M$. We show that for each $\{t_i, t_j\} \in M$, there is a 'half clause' $C'_{i,j} \in \{V(P_{i-j}^{1/2}), V(P_{j-i}^{1/2})\}$ satisfied by the assignment carried by every computational path $P$ containing $X$ with $S \subseteq A(P)$. We prove this by showing that otherwise there are computational paths $Q'$ and $Q''$ both containing $X$ and with $S \subseteq A(Q') \cap A(Q'')$ such that $Q'$ falsifies one of these clauses and $Q''$ falsifies the other one. Then, because $Z$ is a uniform $d$-$\text{NBP}$, $Q'$ and $Q''$ can be 'combined' into a computational path $Q'$ falsifying $C'_{i,j}$, in contradiction to the definition of $Z$. It follows that every assignment carried by a computational path $P$ containing $X$ with $S \subseteq A(P)$ satisfies $\varphi \in \text{CNF}(M)$ consisting of the half clauses $C'_{i,j}$ for $\{t_i, t_j\} \in M$.

In the final stage of the proof, we set $\psi_X = \text{Conj}(S_1) \land \varphi_1 \lor \cdots \lor \text{Conj}(S_q) \land \varphi_q,$ where $S_1, \ldots, S_q$ are all possible sets of literals assigning all the variables except $\bigcup M$ and $\varphi \in \text{CNF}(M)$ is determined regarding $S_i$ as specified in the previous paragraph. We show that $\psi_X$ is a $m^{0.999}/3$-witnessing formula for $X$. Indeed, let $P$ be a path containing $X$. Due to the uniformity, there is $S_i \in A(P)$ for some $1 \leq i \leq q$ (that is $A(P)$ satisfies $\text{Conj}(S_i)$). Then, by definition of $\varphi_i$, $A(P)$ satisfies $\varphi_i$ and hence $A(P)$ satisfies $\psi_X$, concluding the proof.

**Definition 14 (Partition of a path and a generating set).** Let $P$ be a path of $Z$. A sequence $P_1, \ldots, P_c$ of subpaths of $P$ is a partition of $P$ if $P_i$ is a prefix of $P$ for each $1 \leq i \leq c$, the first vertex of $P_i$ is the last vertex of $P_{i-1}$, and $P_c$ is a suffix of $P$.

We say that the set $E$ of ends of $P_1, \ldots, P_{c-1}$ generates $P_1, \ldots, P_c$ on $P$.

**Definition 15 (Variable separators).** Let $X \subseteq V(Z)$ not containing the source nor the sink vertex of $Z$.

We say that $X$ separates two disjoint subsets $Y_1$ and $Y_2$ of $\text{Vars}(Z)$ if there is a computational path $P$ of $Z$ passing through all the vertices of $X$ such that one of the following two statements is true regarding the partition $P_1, \ldots, P_{|X|+1}$ generated by $X$ on $P$.

1. Elements of $Y_1$ occur only on paths $P_i$ with odd $i$ and elements of $Y_2$ occur only on paths $P_i$ with even $i$.
2. Elements of $Y_1$ occur only on paths $P_i$ with even $i$ and elements of $Y_2$ occur only on paths $P_i$ with odd $i$.

The proofs of the following two lemmas is provided in the Appendix.
Lemma 5. On each computational path $P$ of $Z$ there is a set $X$ of vertices size at most $\log n/4000$ that separates two disjoint subsets of Roots$(H)$ of size at least $m^{0.999}$.

Lemma 6. Let $Z$ be a $d$-NBP and let $X$ be a subset of its vertices separating two disjoint subsets $Y_1$ and $Y_2$ ofVars$(Z)$. Let $Y_1' \subseteq Y_1$ and $Y_2' \subseteq Y_2$. Let $S$ be an assignment toVars$(Z) \setminus (Y_1 \cup Y_2)$ and $Q'$ and $Q''$ be two computational paths of $Z$ containing $S$ such that $S \subseteq A(Q') \cap A(Q'')$. $Q'$ assigns negatively all the variables of $Y_1'$ and $Q''$ assigns negatively all the variables of $Y_2'$. Then there is a computational path $Q^*$ of $Z$ containing $X$ with $S \subseteq A(Q^*)$ that assigns negatively all the variables of $Y_1' \cup Y_2'$.

Lemma 7. Let $Z$ be a $d$-NBP computing $\phi_H$. Let $X \subseteq V(Z)$ be a subset of vertices of $Z$ separating subsets $Y_1$ and $Y_2$ of Roots$(H)$ and let $M$ be a pseudomatching of $H$ between $Y_1$ and $Y_2$.

Then for each set $S$ of literals over $V(H) \setminus \bigcup M$ there is $\varphi \in \text{CNF}(M)$ such that for each path $P$ containing $X$ and with $S \subseteq A(P)$, $\varphi$ is satisfied by $A(P)$.

**Proof.** Fix a set $S$ of literals over $V(H) \setminus \bigcup M$. For \{ti, tj\} $\in M$, let $C_{i,j}^{1/2}$ be clauses whose sets of literals are, respectively, $V(P_{i\rightarrow j})$ and $V(P_{j\rightarrow i})$.

We first show that for each \{ti, tj\} $\in M$, there is $C'_{i,j} \in \{C_{i,j}^{1/2}, C_{j,i}^{1/2}\}$ such that for each path $P$ containing $X$ with $S \subseteq A(P)$, $C'_{i,j}$ is satisfied by $A(P)$.

Indeed, suppose this is not true. This means that there are two computational paths $Q'$ and $Q''$ containing $X$ with $S \subseteq A(Q') \cap A(Q'')$ such that $Q'$ falsifies $C_{i,j}^{1/2}$ and $Q''$ falsifies $C_{j,i}^{1/2}$ (it is necessary that $Q' \neq Q''$ because otherwise $Q'$ falsifies both $C_{i,j}^{1/2}$ and $C_{j,i}^{1/2}$ and hence the whole $C_{i,j}$, a contradiction). All the non-root variables of $C_{i,j}$ belong to $V(H) \setminus \bigcup M$ and hence occur in $S$. As $Q'_i$ falsifies $C_{i,j}^{1/2}$, all the non-root variables of $C_{i,j}^{1/2}$ occur negatively in $S$. Symmetrically, all the non-root variables of $C_{j,i}^{1/2}$ occur negatively in $S$. Therefore, all the non-root variables of $C_{i,j}$ occur negatively in $S$. Assume w.l.o.g. that $ti \in Y_1$ and $tj \in Y_2$. Then, by Lemma 6 there is a computational path $Q^*$ of $Z$ passing through $X$ with $S \subseteq A(Q)$ and both $ti$ and $tj$ occurring negatively. That is $Q^*$ falsifies $C_{i,j}$ in contradiction to the definition of $Z$ computing $\phi$. Thus $C''_{i,j}$, as specified in the previous paragraph indeed exists.

It follows that each computational path $P$ of $Z$ containing $X$ and with $S \subseteq A(P)$ satisfies the CNF $\varphi$ whose clauses are $C'_{i,j}$, for every \{ti, tj\} $\in M$. Clearly $\varphi \in \text{CNF}(M)$.

**Proof of Theorem 2** Let $P$ be a computational path of $Z$. By Lemma 5, $P$ contains a set $X$ of vertices of size at most $\log n/4000$ that separates two disjoint subsets $Y_1$ and $Y_2$ of Roots$(H)$ of size at least $m^{0.999}$ each.

By the large pseudomatching property of pseudoexpanders, there is a pseudomatching $M$ of $H$ between $Y_1$ and $Y_2$ with $|M| \geq m^{0.999}/3$.

Let $S_1, \ldots, S_q$ be all sets of literals over $V(H) \setminus \bigcup M$. By Lemma 7 for each $S_i$ there is $\varphi_i \in \text{CNF}(M)$ that is satisfied by $A(Q)$ for each computational path $Q$ containing $X$ provided that $S_i \subseteq A(Q)$.

Let $\psi = (\text{Conj}(S_1) \land \varphi_1) \lor \cdots \lor (\text{Conj}(S_q) \land \varphi_q)$. Clearly $\psi \in \text{ORCNF}(M)$. It remains to show that $\psi$ is satisfied by the assignment carried by each computational...
path \(Q\) containing \(X\). For this notice that \(A(Q)\) necessarily contains some \(S_i\) (since \(Z\) is a uniform \(d\)-NBP, all the variables occur in \(A(Q)\), hence this \(S_i\) is just the projection of \(A(Q)\) to \(V(H) \setminus \bigcup M\)). Therefore, by the previous paragraph, \(A(Q)\) satisfies \(\varphi_i\) and hence also \(\text{Conj}(S_i \land \varphi_i)\). Consequently, \(A(Q)\) satisfies \(\psi\). ■

6 Proof of Theorem 3

Recall that Theorem 3 states that if \(M_1, \ldots, M_r\) are large pseudomatchings of \(H \in \text{PE}\) and \(\psi_1, \ldots, \psi_r\) are formulae such that \(\psi_i \in \text{ORCNF}(M_i)\) for \(1 \leq i \leq r\) and every satisfying assignment of \(\phi_H\) satisfies one of \(\psi_1, \ldots, \psi_r\), then \(r\) is exponentially large. To prove this, we define a probability space over the set of satisfying assignments of \(\phi_H\). Then we fix a large matching \(M\) and \(\psi \in \text{ORCNF}(M)\) and show that the probability of satisfaction of \(\psi\) is exponentially small in \(|M|\). By the union bound, Theorem 3 follows.

In order to prove the small probability of satisfaction of \(\psi\) we introduce a special property of sets of literals over \(V(H) \setminus M\) and show that: (i) the probability of the satisfaction of \(\psi\) conditioned on this property is exponentially small in \(|M|\) and (ii) the probability that this property is not satisfied is exponentially small in \(|M|\). Thus the probability of satisfaction of \(\psi\) is upper bounded by the sum of these two exponentially small quantities and hence is exponentially small itself. We provide further informal explanation of the use of the above special property after it has been defined.

In this section we refer to the variables of \(\phi_H\) that are root, leaf, and the rest of vertices of \(V(H)\) as the root, leaf, and internal variables of \(\phi_H\), respectively. All sets of literals considered in this section are over subsets of variables of \(\phi_H\).

Definition 16 (Fixed set of literals). A literal \(\ell_{i,j}\) (that is, the positive literal of the variable \(\ell_{i,j}\) which is the joint leaf of trees \(T_i\) and \(T_j\)) is fixed w.r.t. a set \(S\) of literals if the following two conditions are true.

- \(\ell_{i,j} \in S\);
- the rest of variables of \(C_{i,j}\) occur negatively in \(S\).

We denote the set of fixed literals of \(S\) by \(\text{Fix}(S)\).

Probability space for the set of satisfying assignments of \(H\). Denote by \(\text{SAT}(H)\) the set of satisfying assignments of \(\phi_H\), which is the family of sets \(S\) of literals with \(\text{Var}(S) = V(H)\) that satisfy \(\phi_H\). In this section we use the probability space with the sample space \(\text{SAT}(H)\) and for each \(S \in \text{SAT}(H)\), the probability of the elementary event \(\{S\}\) is \((1/2)^{|S \setminus \text{Fix}(S)|} = (1/2)^{|V(H) \setminus \text{Fix}(S)|}\). In order to ensure that this definition is valid, we prove that the probabilities of elementary events sum up to 1.

Proposition 1 (Validity of the probability space). \(\sum_{S \in \text{SAT}(H)} (1/2)^{|S \setminus \text{Fix}(S)|} = 1\)

The proof is postponed to the appendix.

The following is the main statement to be proved in this section.
Theorem 6. There are constants $0 < \lambda < 1$ and $0 < \mu < 1$ such that for any sufficiently large $H \in \text{PE}$ with $m = |\text{Roots}(H)|$, any pseudomatching $M$ of $H$ of size at least $m^{0.999}/3$ and any $\psi \in \text{ORCNF}(M)$, the probability that a satisfying assignment of $\phi_H$ satisfies $\psi$ is at most $\lambda m^{\mu}$.

Assuming that Theorem 6 holds, we can now prove Theorem 3.

Proof of Theorem 3. Let $n = |V(H)|$. It is not hard to observe that $n \leq 3m^2$. Therefore, there is a constant $0 < \gamma < 1$ such that $\lambda m^{\mu} \leq \lambda n^{\gamma}$ and, hence, according to Theorem 6, the probability that a satisfying assignment of $\phi_H$ satisfies $\psi$ at most $\lambda n^{\gamma}$.

Let $\alpha = 1/\lambda$ and assume by contradiction that $|\Psi| < \alpha n^{\gamma}$. By the union bound, the probability that an assignment of $\phi_H$ satisfies one of the elements of $\Psi$ is at most $|\Psi| \leq \lambda n^{\gamma} < 1$, the inequality follows from the assumption that there is a satisfying assignment of $\phi_H$ that does not satisfy any element of $\Psi$, in contradiction to its definition.

In the rest of this section we prove Theorem 6.

Events. The events are subsets of $\text{SAT}(H)$.

- **Containment event**: for a set $S$ of literals, $\text{EC}(S)$ denotes the set of all elements $S' \in \text{SAT}(H)$ such that $S \subseteq S'$.

- **Satisfiability event**: for a formula $\varphi$, $\text{ES}(\varphi)$ denotes the set of all elements of $S' \in \text{SAT}(H)$ that satisfy $\varphi$.

The following lemma will allow us to move back and forth between the probability of the union of events and the sum of their probabilities.

Lemma 8. Let $S_1$ and $S_2$ be two distinct sets of literals such that $\text{Var}(S_1) = \text{Var}(S_2)$. Then $\text{EC}(S_1) \cap \text{EC}(S_2) = \emptyset$.

Proof. Suppose $S \in \text{EC}(S_1) \cap \text{EC}(S_2)$. Then the projection of $S$ to $\text{Var}(S_1)$ is both $S_1$ and $S_2$, which is impossible since $S_1 \neq S_2$.

Definition 17 (Siblings in $H$). Two internal vertices $u$ and $v$ of $H$ are siblings if they belong to the same tree $T_i$ and if $u$ and $v$ are siblings in $T_i$.

(This definition is unambiguous because each internal vertex belongs to exactly one tree of $T(H)$.)

Definition 18 (Set of literals respecting a pseudoedge). A set $S$ of literals respects a pseudoedge $\{t_i, t_j\}$ if the following two conditions hold.

1. All non-root variables of $C_{i,j}$ occur negatively in $S$.
2. The siblings of all internal variables of $C_{i,j}$ occur positively in $S$.

Definition 19 ($\eta$-comfortable set of literals). For $0 \leq \eta \leq 1$, a set $S$ of literals is $\eta$-comfortable w.r.t. a pseudomatching $M$ if the following two conditions hold.

1. $S$ does not falsify any clause of $\phi_H$
2. \( S \) respects at least \( \eta \ast |M| \) pseudoedges of \( M \).

We continue the informal explanation started in the beginning of this section. The special property on sets of assignments \( S \) over \( V(H) \setminus \bigcup M \) mentioned there is that \( S \) is \( \eta \)-comfortable w.r.t. \( M \) for \( \eta = c \ast m^{-4/4.9} \) for a specially chosen constant \( c \). We prove that (i) if \( S \) is \( \eta \)-comfortable w.r.t. \( M \) then for any \( \varphi \in \text{CNF}(M) \), \( \Pr(ES(\varphi) \in EC(S)) \leq (2/3)^{\eta \ast |M|} \) and (ii) the probability that an element of \( \text{SAT}(H) \) is not \( \eta \)-comfortable is exponentially small in \( |M| \).

The proof of (i) is, essentially, a reduction from the following: (iii) if \( S \) is an assignment of \( V(H) \setminus M \) that is 1-comfortable w.r.t. \( M \) then \( \Pr(ES(\varphi) \in EC(S)) \leq (2/3)^{|M|} \). To prove (iii), we note \( S \) assigns all the variables of \( \phi_H \) except the root variables of \( C_{i,j} \) for \( \{t_i, t_j\} \in M \). Moreover, by Definition 18 all the non-root variables of each such \( C_{i,j} \) are assigned negatively. Therefore, for each such \( C_{i,j} \), either \( t_i \) or \( t_j \) is assigned positively, thus making 3 choices per clause and the total number of possible elements of \( EC(S) \) being at most \( 3^{|M|} \). It may seem that some choices are not available because negative assignment to some \( t_i \), together with \( S \), could falsify clauses corresponding to pseudoedges that are not in \( M \). However, this does not happen since positive assignments to siblings of internal variables of \( C_{i,j} \) eliminate such a possibility, so the number of elements of \( EC(S) \) is indeed \( 3^{|M|} \).

Moreover, we show that each element of \( EC(S) \) has the same set of fixed literals and has the same probability. Therefore, proving (iii) amounts to proving that at most \( 2^{|M|} \) elements of \( EC(S) \) satisfy \( \varphi \).

In order to show this, recall that \( \varphi \in \text{CNF}(M) \) consists of ‘halves’ of clauses \( C_{i,j} \) for \( \{t_i, t_j\} \in M \). Hence every element of \( EC(S) \) satisfying \( \varphi \) must satisfy all the root variables of these halves, leaving 2 choices per clause (either positive or negative assignment of the other root variable) and the total number of choices is at most \( 2^{|M|} \) as required.

To prove statement (ii) we define for each \( \{t_i, t_j\} \in M \) a \( 1 \sim 0 \) random variable \( X_{i,j} \) which is 1 exactly on those elements of \( \text{SAT}(H) \) that respect \( \{t_i, t_j\} \). We then prove that (iv) the probability that \( X_{i,j} = 1 \) is at most \( c_1 \ast m^{-4/4.9} \) for some constant \( c_1 \) and that (v) the variables \( X_{i,j} \) are mutually independent. By Chernoff’s bound, with high probability, an element of \( \text{SAT}(H) \) respects at least \( c \ast m^{-4/4.9} \ast |M| \) edges of \( M \) where \( c \) is a constant dependent on \( c_1 \) (and we choose \( \eta = c \ast m^{-4/4.9} \)), thus implying (ii).

For statement (iv), note that because of the small height property of pseudoexpanders, each clause \( C_{i,j} \) of \( \phi_H \) has about \( 2 \log m/4.9 \) variables. Together with about the same number of siblings of internal variables of \( C_{i,j} \), this means that assignments of about \( 4 \log m/4.9 \) variables of \( \phi_H \) must be fixed in an element of \( \text{SAT}(H) \) in order to respect \( \{t_i, t_j\} \). Although, due to ‘non-uniform’ character of the probability space, some careful calculation is required, this is, essentially the reason why the probability of respecting \( t_{i,j} \) is \( O(2^{-4 \log m/4.9}) = m^{-4/4.9} \).

The underlying reason for statement (v) is that respecting of two distinct pseudoedges of a pseudomatching requires fixing assignments on disjoint sets of variables of \( \phi_H \) (though, again, in the considered probability space, this implication is not trivial and requires a careful calculation).
The statement (i) is Theorem \ref{thm:main} and it is proved in Subsection \ref{subsec:proof-thm-main}. The statements (iv) and (v) are Theorem \ref{thm:projection-chain} whose proof is provided in Subsection \ref{subsec:proof-thm-projection-chain}. We now show how Theorem \ref{thm:projection-chain} follows from (i) and (ii).

The proof of Theorem \ref{thm:projection-chain} amounts to showing that $\sum_{i=1}^q \Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i))$ is exponentially small, where $S_1, \ldots, S_q$ are all possible assignments over $V(H) \setminus M$ and $\varphi_1, \ldots, \varphi_q$ are elements of $\text{CNF}(M)$ such that $\psi = \bigvee_{i=1}^q (\text{Conj}(S_i) \land \varphi_i)$. Because of the possibility of rearrangement, we can assume w.l.o.g. that for some $r \leq q$, $S_1, \ldots, S_r$ are $\eta$-comfortable w.r.t. $M$ while $S_{r+1}, \ldots, S_q$ are not. Then $\sum_{i=r+1}^q \Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i))$ is exponentially small by (ii) and $\sum_{i=1}^r \Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i)) = \sum_{i=1}^r \Pr(\text{ES}(\varphi_i) | \text{EC}(S_i)) \ast \Pr(\text{EC}(S_i))$. All the conditional probabilities are exponentially small according to (i), therefore, the last quantity can be upper bounded by a number exponentially small in $|M|$ multiplied by $\sum_{i=1}^r \Pr(\text{EC}(S_i)) \leq \sum_{i=1}^r \Pr(\text{EC}(S_i)) = 1$. Thus as stated in the beginning of this section, $\Pr(\text{ES}(\psi))$ can be upper bounded by the sum of two exponentially small quantities.

**Lemma 9.** A satisfying assignment $S$ of $\phi_H$ is $\eta$-comfortable w.r.t. a pseudomatching $M$ of $H$ if and only if the projection $S'$ of $S$ to $V(H) \setminus M$ is $\eta$-comfortable w.r.t. $M$.

**Proof.** Assume $S$ is $\eta$-comfortable w.r.t. $M$. By definition, this means that $S$ contains particular occurrences of variables all of which are non-root ones. Since $\bigcup M$ consists of root variables only, these occurrences are preserved in $S'$. Therefore, $S'$ is $\eta$-comfortable w.r.t. $M$.

Conversely, if $S'$ is $\eta$-comfortable w.r.t. $M$ then the witnessing occurrences of variables remain in any superset of $S'$, in particular in $S$. As $S$ does not falsify any clause, we conclude that $S$ is $\eta$-comfortable w.r.t. $M$. ∎

**Theorem 7.** Let $M$ be a pseudomatching of $H$, $0 \leq \eta \leq 1$. Then for any set $S$ of literals over $(V(H) \setminus M)$, which is $\eta$-comfortable w.r.t. $M$ and for any $\varphi \in \text{CNF}(M)$, $\Pr(\text{ES}(\varphi) | \text{EC}(S)) \leq (2/3)^{\eta \ast |M|}$.

The proof is provided in Subsection \ref{subsec:proof-thm-main}.

**Indicator variables for pseudoeedges and their sums.** Let $\{t_i, t_j\}$ be a pseudoedge of $H$. The indicator variable $X_{i,j}$ is a $1 \ast 0$ variable such that for $S \in \text{SAT}(H)$, $X_{i,j}(S) = 1$ if and only if $S$ respects $\{t_i, t_j\}$.

For a pseudomatching $M$ of $H$, let $X_M = \sum_{(t_i, t_j) \in M} X_{i,j}$.

**Theorem 8 (Statements about indicator variables.).**

1. For any pseudoeedge $\{t_i, t_j\}$ of $H$, $\Pr(X_{i,j} = 1) \geq 3/128 \ast m^{-4/4.9}$.

2. For any pseudomatching $M$, the variables $\{X_{i,j} | \{t_i, t_j\} \in M\}$ are mutually independent.

The proof appears in Subsection \ref{subsec:proof-thm-projection-chain}.

**Lemma 10.** Let $M$ be a pseudomatching of $H$ and $\eta = 3/12800 \ast m^{-4/4.9}$. Then $\Pr(X_M < \eta \ast |M|) \leq 0.4^{100 \ast \eta \ast |M|}$.
Proof. By Theorem \[8\] the variables \( \{X_{i,j} \mid \{t_i, t_j\} \in M \} \) are mutually independent, which allows us to use Chernoff’s bounds.

For this, we first calculate a lower bound on the expected value of \( X_M \).

\[
E[X_M] = \sum_{\{t_i, t_j\} \in M} E[X_{i,j}] = \sum_{\{t_i, t_j\} \in M} Pr(X_{i,j} = 1) \geq 100 * \eta * |M| \tag{1}
\]

where the last inequality follows from Theorem \[8\] stating that \( Pr(X_{i,j} = 1) \geq 100 * \eta \).

To apply Chernoff’s bounds, we use inequality (4.4.) of Theorem 4.5. of \[9\]. We provide it here for convenience, adapted to our notation

\[
Pr(X_M \leq (1 - \delta) * E[X_M]) \leq \left( \frac{e^{-\delta}}{(1 - \delta)^{1 - \delta}} \right)^{E[X_M]} \tag{2}
\]

for any \( 0 < \delta < 1 \). Now, we obtain the following.

\[
Pr(X_M < \eta * |M|) \leq Pr(X_M \leq E[X_M]/100) = Pr(X_M \leq (1 - 99/100) * E[X_M]) \leq 0.4^{E[X_M]} \leq 0.4^{100 * \eta * |M|} \tag{3}
\]

where the first inequality follows from \( 1 \) and the obvious fact that \( Pr(X_M < E[X_M]/100) \leq Pr(X_M \leq E[X_M]/100) \), the second inequality follows from substituting 0.99 to \( \delta \) in \( 2 \) and the inequality \( \frac{e^{-0.99}}{0.99} \leq 0.4 \) verified by a straightforward calculation, the third inequality follows from \( 1 \). \( \blacksquare \)

Proof of Theorem \[6\]. By definition \( \psi = Conj(S_1) \land \varphi_1 \lor \ldots \lor Conj(S_q) \land \varphi_q \), where \( S_1, \ldots, S_q \) are sets of literals over \((V(H) \setminus \cup M)\) and \( \varphi_i \in CNF(M) \) for \( 1 \leq i \leq q \).

Observe that

\[
ES(\psi) = \bigcup_{i=1}^{q} [ES(\varphi_i) \cap EC(S_i)] \tag{4}
\]

Indeed, assume \( S \in ES(\psi) \). Then \( S \) satisfies one of the conjuncts of \( \psi \). Hence, there is \( 1 \leq i \leq q \) such that \( S \in ES(Conj(S_i) \land \varphi_i) \). Therefore, \( S \) satisfies both \( Conj(S_i) \) and \( \varphi_i \). Thus, \( S \in ES(Conj(S_i)) \cap ES(\varphi_i) \). To satisfy a conjunction of literals, \( S \) must contain all these literals, as a result \( S \in EC(S_i) \). Consequently, \( S \in ES(\varphi_i) \cap EC(S_i) \) and hence \( S \in \bigcup_{i=1}^{q} [ES(\varphi_i) \cap EC(S_i)] \).

Conversely, assume that \( S \in \bigcup_{i=1}^{q} [ES(\varphi_i) \cap EC(S_i)] \). Then \( S \in ES(\varphi_i) \cap EC(S_i) \) for some \( 1 \leq i \leq q \). Since \( S \) contains all the literals of \( S_i \), \( S \) satisfies the conjunction of these literals. Therefore, \( S \in ES(Conj(S_i)) \). Since \( S \) satisfies both \( \varphi_i \) and \( Conj(S_i) \), \( S \) also satisfies their conjunction. Consequently, \( S \in ES(Conj(S_i) \land \varphi_i) \). Finally, since \( Conj(S_i) \land \varphi_i \) is a disjunct of \( \psi \), we conclude that \( S \in ES(\psi) \).

Set \( \eta = 3/12800 * m^{-4/9} \). Assume w.l.o.g., because of the possibility of rearrangement, that \( S_1, \ldots, S_r \) are the \( \eta \)-comfortable w.r.t. \( M \) and \( S_{r+1}, \ldots, S_q \) are not \( \eta \)-comfortable.
Then

\[ Pr(\text{EC}(\psi)) = Pr\left(\bigcup_{i=1}^{q} [\text{ES}(\varphi_i) \cap \text{EC}(S_i)] \right) = \sum_{i=1}^{q} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i)) = \sum_{i=1}^{r} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i)) + \sum_{i=r+1}^{q} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i)) \quad (5)\]

where the first equality follows from Lemma 3. For the second equality, notice that by Lemma 3, \(\text{EC}(S_i)\) for \(1 \leq i \leq q\) are mutually disjoint. Therefore, \(\text{ES}(\varphi_i) \cap \text{EC}(S_i)\) for \(1 \leq i \leq q\) are also mutually disjoint and hence the probability of their union can be replaced by the sum of their probabilities. The third equality is correct because the right-hand part of its is a regrouping of items in the left-hand part.

We are now, going to show that both \(\sum_{i=1}^{r} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i))\) and \(\sum_{i=r+1}^{q} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i))\) are exponentially small from where the theorem will immediately follow.

For the former,

\[ \sum_{i=1}^{r} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i)) = \sum_{i=1}^{r} [Pr(\text{ES}(\varphi_i) \mid \text{EC}(S_i)) \ast Pr(\text{EC}(S_i))] \leq \sum_{i=1}^{r} [(2/3)^{\eta_1} \ast Pr(\text{EC}(S_i))] = (2/3)^{\eta_1} \ast Pr(\bigcup_{i=1}^{r} \text{EC}(S_i)) \leq (2/3)^{\eta_1} \ast |M| \quad (6)\]

where the first equality follows from the definition of conditional probability and the first inequality follows from Theorem 7. The second equality is the result of moving \((2/3)^{\eta_1} \ast |M|\) out of the brackets and replacing the sum of probabilities by the probability of union, the replacement enabled by Lemma 3 (see the explanation to (5) for the detailed justification). Finally, the second inequality holds because \(Pr(\bigcup_{i=1}^{r} \text{EC}(S_i)) \leq 1\).

In order to establish an exponentially small lower bound on \(\sum_{i=r+1}^{q} Pr(\text{ES}(\varphi_i) \cap \text{EC}(S_i))\), the key observation is the following.

\[ \bigcup_{i=r+1}^{q} \text{EC}(S_i) = \{ S' \mid (S' \in \text{SAT}(H)) \wedge (X_M(S') < \eta \ast |M|) \} \quad (7)\]

Indeed, assume that \(S \in \bigcup_{i=r+1}^{q} \text{EC}(S_i)\). Then, \(S_i \subseteq S\) for some \(r + 1 \leq i \leq q\). By our assumption, \(S_i\) is not \(\eta\)-comfortable w.r.t. \(M\). Hence, as \(S\) is a satisfying assignment of \(\phi_H\) \(S\) is not \(\eta\)-comfortable w.r.t. \(M\) according to Lemma 9.

As \(S\) is a satisfying assignment of \(\phi_H\), the only way for \(S\) to be not \(\eta\)-comfortable is to respect less than \(\eta \ast |M|\) pseudoedges \(\{t_i, t_j\}\) of \(M\). Consequently, less than \(\eta \ast |M|\) corresponding variables \(X_{i,j}\) equal 1 on \(S\) and hence \(X_M(S) < \eta \ast |M|\). That is, \(S \in \{ S' \mid (S' \in \text{SAT}(H)) \wedge (X_M(S') < \eta \ast |M|) \}\).

Conversely, assume that \(S \in \{ S' \mid (S' \in \text{SAT}(H)) \wedge (X_M(S') < \eta \ast |M|) \}\) That is, \(X_M(S) < \eta \ast |M|\). Hence less than \(\eta \ast |M|\) variables \(\{X_{i,j}\} \{t_i, t_j\} \in M\) are 1 on \(S\), that is, in turn, \(S\) respects less than \(\eta \ast |M|\) pseudoedges of \(M\), hence \(S\) is not \(\eta\)-comfortable w.r.t. \(M\). As \(S\) is a satisfying assignment of \(\phi_H\), the projection \(S^*\) of \(S\) to
$V(H) \setminus \cup M$ is not $\eta$-comfortable by Lemma 9. By assumption, there is $r + 1 \leq i \leq q$ such that $S_i = S^*$. Hence, $S \in EC(S_i)$ as required.

Now, we obtain the following:

$$\sum_{i=r+1}^{q} Pr(ES(\varphi_i) \cap EC(S_i)) \leq \sum_{i=r+1}^{q} Pr(EC(S_i)) = Pr(\bigcup_{i=r+1}^{q} EC(S_i)) = Pr(\{S' \in SAT(H) \wedge (X_M(S') < \eta * |M|)\}) = Pr(X_M < \eta * |M|) \leq 0.4^{100*\eta*|M|} \quad (8)$$

For the first inequality, note that for each $r + 1 \leq i \leq q$, $ES(\varphi_i) \cap EC(S_i) \subseteq EC(S_i)$, hence the probability of the event on the left-hand side does not exceed the probability of the event on the right-hand side. The first equality follows from Lemma 8 (see the explanation to (5) for the detailed justification). The second equality is just (7). The third equality is just an effect of changing notation for the probability of the same event. The fourth inequality follows from Lemma 10.

By substituting (6) and (8) into the last item of (5) and then, by substituting the value of $\eta$ and the assumed lower bound of $|M|$, we obtain the following.

$$Pr(EC(\psi)) \leq (2/3)^{\eta * |M|} + 0.4^{100*\eta*|M|} \leq (2/3)^{3/12800+s+m^{-4/4.9}*sm^{0.999}/3} + (0.4)^{100*3/12800+s+m^{-4/4.9}*sm^{0.999}/3} \quad (9)$$

Let $\mu = 0.999 - 4/4.9$. Clearly, $0 < \mu < 1$ and there is $\lambda$ such that, for a sufficiently large $m$, $Pr(EC(\psi)) \leq \lambda^m$.

### 6.1 Proof of Theorem 7

Referring to the informal explanation provided after Definition 19, Theorem 7 is statement (i). The proof strategy outlined there is implemented in this section as follows.

Using the next lemma we will be able to conclude that if $S$ is a partial assignment on $V(H) \setminus \cup M$ that is 1-comfortable for $M$, then all $3^{|M|}$ elements of $EC(S)$ are satisfying assignments of $\phi_H$. Next, Lemma 12 states that all these elements have the same probability. After that, in Lemma 13 we establish the claim of Theorem 7 for $\eta = 1$ and then, in the actual proof of Theorem 7 we do ‘reduction’ to an arbitrary $\eta$.

**Lemma 11.** Let $M$ be a pseudomatching and let $S$ be a partial assignment on $V(H) \setminus \cup M$ that is 1-comfortable for $M$ (that is, $S$ does not falsify any clause of $\phi_H$ and respects all the pseudoedges of $M$). Let $\{t_i, t_j\} \in M$ and let $\{t_i, t_j, t_l\} \neq \{t_i, t_j\}$ be another pseudoedge of $H$ (not contained in $M$ due to having a joint end with $\{t_i, t_j\}$). Then $S$ assigns positively at least one internal variable of $C_{i,j}$.

**Proof.** The root $t_i$ of $T_i$ is a common ancestor of $\ell_{i,j}$ and of $\ell_{i,j'}$. Therefore, we can identify the lowest common ancestor $u$ of $\ell_{i,j}$ and $\ell_{i,j'}$ in $T_i$. Then $u$ has two children $v$ and $w$ (if it had one child that child would be a common ancestor of $\ell_{i,j}$ and $\ell_{i,j'}$ in contradiction to being $u$ the lowest one. Assume w.l.o.g. that $v$ is an ancestor of $\ell_{i,j}$.}
Then \( v \) belongs to \( P_{i,j} \), hence \( v \) is a variable of \( C_{i,j} \). As \( v \) is not a root variable (due to having a parent), \( v \) occurs negatively in \( S \). Notice that \( v \neq \ell_{i,j} \) because \( v \) has a sibling in \( T_i \) while \( \ell_{i,j} \) does not. Therefore, \( v \) is an internal variable of \( C_{i,j} \) and hence its sibling \( w \) occurs positively in \( S \).

Observe that \( w \) is an ancestor of \( \ell_{i,j'} \). Indeed, otherwise, \( v \) is an ancestor of \( \ell_{i,j'} \) in contradiction to \( u \) being the lowest common ancestor of \( \ell_{i,j} \) and \( \ell_{i,j'} \). This means that \( w \) belongs to \( P_{i,j'} \) and hence \( w \) is a variable of \( C_{i,j'} \). Furthermore, \( w \neq i,j \) because \( w \) has a sibling \( v \) while \( \ell_{i,j'} \) does not have siblings. As \( w \neq t_i \) due to having a parent, we conclude that \( w \) is an internal variable of \( C_{i,j'} \) assigned positively by \( S \), as required. \( \blacksquare \)

**Lemma 12.** Let \( M \) be a pseudomatching and let \( S \) be a partial assignment on \( V(H) \setminus \bigcup M \) that is 1-comfortable for \( M \). Then for every \( S' \in EC(S) \), \( Pr(S') = 1/2^{V(H) \setminus Fix(S)} \).

**Proof.** By definition, \( Pr(S') = 1/2^{V(H) \setminus Fix(S')} \). We are going to show that \( Fix(S') = Fix(S) \).

Assume that \( \ell_{i,j} \in Fix(S) \). This means that \( S \) occurs positively in \( S \) and the rest of variables of \( C_{i,j} \) occur negatively in \( S \). As \( S \subseteq S' \), all these occurrences are preserved in \( S' \) and hence \( \ell_{i,j} \in Fix(S') \). That is \( Fix(S) \subseteq Fix(S') \).

Conversely, assume that \( \ell_{i,j} \in Fix(S') \). Then \( \{t_i,t_j\} \notin M \). Indeed, otherwise, as \( S \) respects \( \{t_i,t_j\} \), \( \ell_{i,j} \) is assigned negatively in \( S \) and hence in \( S' \) in contradiction to the assumption that \( \ell_{i,j} \in Fix(S') \).

Furthermore, it cannot happen that \( \{t_i,t_j\} \cap \bigcup M = 1 \). Indeed, otherwise, there is a pseudoedge, say \( \{t_i,t_j'\} \in M \) such that \( \{t_i,t_j\} \neq \{t_i,t_j'\} \). Then, by Lemma 11 at least one non-leaf variable of \( C_{i,j} \) is assigned positively by \( S \) and hence by \( S' \). However, this is a contradiction to \( \ell_{i,j} \in Fix(S') \) requiring all the non-leaf variables of \( C_{i,j} \) to be assigned negatively.

It remains to assume that \( \{t_i,t_j\} \cap \bigcup M = \emptyset \). But in this case all the variables of \( C_{i,j} \) are contained in \( V(H) \setminus \bigcup M \) and hence occur in \( S \). In particular, \( \ell_{i,j} \) occurs positively in \( S \) and the rest of variables of \( C_{i,j} \) occur negatively in \( S \). It follows that \( \ell_{i,j} \in Fix(S) \). That is \( Fix(S') \subseteq Fix(S) \). \( \blacksquare \)

**Lemma 13.** Let \( M \) be a pseudomatching and let \( S \) be a partial assignment on \( V(H) \setminus \bigcup M \) that is 1-comfortable for \( M \). Let \( \varphi \in CNF(M) \). Then \( Pr(ES(\varphi)|EC(S)) = (2/3)^{|M|} \).

**Proof.** Let \( SA \) be the set of all extensions of \( S \) to \( V(H) \) such that for each \( \{t_i,t_j\} \in M \), at least one of \( t_i,t_j \) occurs positively in \( S \). Observe that

\[
EC(S) = SA
\]  
(10)

Indeed, let \( S' \in EC(S) \). Then \( S' \) is a satisfying assignment of \( \phi_H \). In particular, for each \( \{t_i,t_j\} \in M \), the clause \( C_{i,j} \) is satisfied. However, all the variables of \( C_{i,j} \), except the root ones occur negatively in \( S \) and hence in \( S' \). It follows that, to satisfy \( C_{i,j} \), at least one of root variables \( t_i \) or \( t_j \) must occur positively in \( S' \). Consequently (taking into account that \( S' \) is an extension of \( S \)), \( S' \in SA \) and hence \( EC(S) \subseteq SA \).

Conversely, assume that \( S' \in SA \). To show that \( S' \in EC(S) \), we need to demonstrate that \( S' \) is a satisfying assignment of \( \phi_H \).
For each \( \{t_i, t_j\} \in M \), the clause \( C_{i,j} \) is satisfied because, by definition of \( \mathbf{SA} \), at least one of \( t_i, t_j \) occurs positively in \( S' \). Consider a clause \( C_{i,j'} \) such that \( i \in \bigcup M \) while \( j' \notin \bigcup M \). By Lemma 11, at least one variable of \( C_{i,j'} \) occurs positively in \( S \) and hence in \( S' \), satisfying \( C_{i,j'} \). Finally consider a clause \( C_{i,j} \) such that none of \( t_i, t_j \) occur in \( \bigcup M \). This means that all the variables of \( C_{i,j} \) occur in \( S \). As \( S \) is 1-comfortable for \( M \), \( S \) does not falsify \( C_{i,j} \) and, since \( S \) contains literals of all the variables of \( C_{i,j} \), \( S \) must satisfy \( C_{i,j} \) and hence so is \( S' \). Consequently, \( S' \in \mathbf{EC}(S) \) and hence \( \mathbf{SA} \subseteq \mathbf{EC}(S) \).

As \( \mathbf{SA} \) allows three independent choices per clause corresponding to a pseudoedge of \( M \), we conclude that:

\[
|\mathbf{SA}| = 3^{|M|} \tag{11}
\]

Now, we are ready to obtain an expression for the probability of \( \mathbf{EC}(S) \).

\[
\Pr(\mathbf{EC}(S)) = \sum_{S' \in \mathbf{EC}(S)} \Pr(\{S'\}) = \sum_{S' \in \mathbf{EC}(S)} \frac{1}{2}^{\left|V(H)\setminus\fix(S)\right|} = \frac{1}{2}^{\left|V(H)\setminus\fix(S)\right|} \cdot |\mathbf{EC}(S)| = \frac{1}{2}^{\left|V(H)\setminus\fix(S)\right|} \cdot 3^{|M|} \tag{12}
\]

where the second equality follows from definition of the probability of an event, the second equality follows from Lemma 12 the third equality a straightforward algebraic transformation, the fourth equality follows from (10), the fifth equality follows from (11).

Recall, that by definition of \( \mathbf{CNF}(M) \), the clauses \( C_{i,j}^{1/2} \) of \( \varphi \) correspond to the pseudoedges \( \{t_i, t_j\} \in M \). In particular, the clause \( C_{i,j}^{1/2} \) is a subclause of \( C_{i,j} \) whose variables are either \( V(P_{i+1}^{1/2}) \) or \( V(P_{j+1}^{1/2}) \). That is, either the vertices of the path from \( t_i \) to \( \ell_{i,j} \) in \( T_i \) or of the path from \( t_j \) to \( \ell_{i,j} \) in \( T_j \). By construction, \( C_{i,j}^{1/2} \) contains exactly one vertex of \( \{t_i, t_j\} \). Denote this vertex by \( \text{root}(i,j) \).

Let \( \mathbf{SB} \) be the set of all extensions \( S' \) of \( S \) to \( V(H) \) such that for each \( \{t_i, t_j\} \in M \), \( \text{root}(i,j) \) occurs positively in \( S' \).

\[
\mathbf{EC}(S) \cap \mathbf{ES}(\varphi) = \mathbf{SB} \tag{13}
\]

Indeed, let \( S' \in \mathbf{EC}(S) \cap \mathbf{ES}(\varphi) \). This means that \( S' \) is an extension of \( S \) and satisfies \( C_{i,j}^{1/2} \) for all \( \{t_i, t_j\} \in M \). For each \( C_{i,j}^{1/2} \) all the non-root variables occur negatively in \( S \) and hence so in \( S' \). That is, the only way for \( S' \) to satisfy \( C_{i,j}^{1/2} \) is to contain the positively occurrence of \( \text{root}(i,j) \). Hence, \( S' \in \mathbf{SB} \), and therefore \( \mathbf{EC}(S) \cap \mathbf{ES}(\varphi) \subseteq \mathbf{SB} \). Conversely, suppose \( S' \in \mathbf{SB} \). Then \( S' \) is an extension of \( S \) to \( V(H) \) in which for each \( \{t_i, t_j\} \), at least one variable occurs positively, namely \( \text{root}(i,j) \). This means that \( S' \in \mathbf{SA} \) and hence \( S' \in \mathbf{EC}(S) \) according to (10). Moreover, because of the positive occurrence of \( \text{root}(i,j) \), for each \( \{t_i, t_j\} \in M \), \( S' \) satisfies all the clauses \( C_{i,j}^{1/2} \) and hence \( S' \) satisfies \( \varphi \). Consequently, \( S' \in \mathbf{ES}(\varphi) \). That is, \( S' \in \mathbf{EC}(S) \cap \mathbf{ES}(\varphi) \) and hence \( \mathbf{SB} \subseteq \mathbf{EC}(S) \cap \mathbf{ES}(\varphi) \). The correctness proof for (13) is now complete.

21
Since an element of $SB$ can be formed by two independent choices per clause corresponding to an element of $M$, we conclude that

$$|SB| = 2^{|M|} \tag{14}$$

$$Pr(\mathcal{EC}(S) \cap \mathcal{ES}(\varphi)) = \sum_{S' \in \mathcal{EC}(S) \cap \mathcal{ES}(\varphi)} Pr\{S'\} = \sum_{S' \in \mathcal{EC}(S) \cap \mathcal{ES}(\varphi)} (1/2)^{|V(H) \setminus Fix(S)|} = \frac{1}{2} 2^{|V(H) \setminus Fix(S)|} = 2^{|V(H) \setminus Fix(S)|} = 2^{|M|} \tag{15}$$

where the reasoning is analogous to (12) with (13) is used instead (10) and (14) is used instead of (11).

Now, with (12) and (15) in mind we obtain the following.

$$Pr(\mathcal{ES}(\varphi) | \mathcal{EC}(S)) = Pr(\mathcal{EC}(S) \cap \mathcal{ES}(\varphi)) / Pr(\mathcal{EC}(S)) = \frac{((1/2)|V(H)\setminus Fix(S)| * 2^{|M|})}{((1/2)|V(H)\setminus Fix(S)| * 2^{|M|})} = (2/3)^{|M|} \tag{16}$$

**Proof of Theorem 7.** Let $M'$ be a subset of $M$ consisting of at least $\eta * |M|$ pseudoedges respected by $S$. Let $\varphi'$ be the sub-CNF of $\varphi$ consisting of the clauses $C_{i,j}^{1/2}$ of $\varphi$ corresponding to all the pseudoedges $\{t_i, t_j\} \in M'$. Clearly $\varphi' \in \text{CNF}(M')$.

Let $S$ be the set of all extensions of $S$ to $V(H) \setminus M'$ that do not falsify any clause. Observe that

$$\mathcal{EC}(S) = \bigcup_{S^* \in S} \mathcal{EC}(S^*) \tag{17}$$

Indeed, assume that $S' \in \mathcal{EC}(S)$. Then the projection $S^*$ of $S'$ to $V(H) \setminus M'$ is an extension of $S$ to $V(H) \setminus M'$ that does not falsify any clauses of $\phi_H$, because $S'$ does not falsify any. Therefore, $S^* \in S$. Clearly, $S^* \in \mathcal{EC}(S^*) \subseteq \bigcup_{S^* \in S} \mathcal{EC}(S^*)$.

For the opposite direction, let $S^* \in \bigcup_{S^* \in S} \mathcal{EC}(S^*)$. Then there is $S^* \in S$ such that $S^* \in \mathcal{EC}(S^*)$. As $S \subseteq S^*$, clearly, $\mathcal{EC}(S^*) \subseteq \mathcal{ES}(S)$. Consequently, $S \in \mathcal{EC}(S)$.

Observe that each $S^* \in S$ respects each pseudoedge $\{t_i, t_j\} \in M'$. Indeed, $S$ respects $\{t_i, t_j\}$ by assumption and, since $S \subseteq S^*$, all literal occurrences witnessing that $S$ respects $\{t_i, t_j\}$ are retained in $S^*$. Since $S^*$ does not falsify any clause of $\phi_H$, $S^*$ is 1-comfortable w.r.t. $M'$. Consequently, by Lemma [13]

$$\forall S^* \in S, Pr(\mathcal{ES}(\varphi') | \mathcal{EC}(S^*)) = (2/3)^{|M'|} \tag{18}$$
Now we obtain the following.

\[
Pr(ES(\varphi) \cap EC(S)) \leq Pr(ES(\varphi') \cap EC(S)) = Pr(ES(\varphi') \cap \bigcup_{S^* \in S} EC(S^*)) = \\
Pr(\bigcup_{S^* \in S} (ES(\varphi') \cap EC(S^*))) = \sum_{S^* \in S} Pr(ES(\varphi') \cap EC(S^*)) = \\
\sum_{S^* \in S} (Pr(ES(\varphi')|EC(S^*)) \cdot Pr(EC(S^*))) = \sum_{S^* \in S} ((2/3)^{|M'|} \cdot Pr(EC(S^*))) = (2/3)^{|M'|} \cdot \sum_{S^* \in S} Pr(EC(S^*)) = \\
(2/3)^{|M'|} \cdot Pr(EC(S)) \leq (2/3)^{|S| \cdot |M'|} \cdot Pr(EC(S)) \quad (19)
\]

where the first inequality is correct because, due to \( \varphi' \) being a sub-CNF of \( \varphi \), an any satisfying assignment of \( \varphi \) is also a satisfying assignment of \( \varphi' \). The first equality follows from (17). The second equality is a standard set-theoretic transformation. The third equality follows from Lemma 8 applied to the elements of \( S \). The fourth equality follows from definition of conditional probability. The fifth equality follows from Lemma 13. The sixth equality is a result of moving \((2/3)^{|M'|}\) outside the brackets and the replacement of \( \sum_{S^* \in S} P_r(EC(S^*)) \) by \( P_r(\bigcup_{S^* \in S} EC(S^*)) \) allowed by Lemma 8. Finally, the second inequality follows from definition of \(|M'|\).

Thus (19) derives that \( P_r(ES(\varphi) \cap EC(S)) \leq (2/3)^{|S| \cdot |M'|} \cdot P_r(EC(S)) \). The theorem immediately follows from dividing both parts of this inequality by \( P_r(EC(S)) \).

\[
\blacksquare
\]

6.2 Proof of Theorem 8

Recall that Theorem 8 consists of two statements. The first one states that \( P_r(X_{i,j}) \geq 3/128 \times m^{-4/9} \), for every pseudodedge \( \{i, j\} \in H \). The second statement claims that, for every pseudomatching \( M \) of \( H \), the variables \( X_{i,j} \) for pseudodges \( \{i, j\} \in M \) are mutually independent.

For a pseudodedge \( \{i, j\} \), denote by \( S_{i,j} \) the set of literals consisting of negative literals of all the non-root variables of \( C_{i,j} \) and positive literals for all the siblings of the internal variables of \( C_{i,j} \).

It follows from the definition of \( X_{i,j} \) that

\[
\text{Observation 1. } P_r(X_{i,j} = 1) = P_r(EC(S_{i,j})�.
\]

\[
\text{Definition 20 (Guarded set of literals). A set } S \text{ of literals is guarded from a clause } C_{i,j} \text{ if one of the following three conditions holds.}
\]

1. \( \ell_{i,j} \) does not occur in \( S \).
2. A non-leaf variable of \( C_{i,j} \) occurs positively in \( S \).
3. \( \ell_{i,j} \) occurs positively in \( S \) and the rest of variables of \( C_{i,j} \) occur negatively in \( S \).

\[
\text{Lemma 14. Let } S \text{ be a guarded set if literals. Then } P_r(EC(S)) = 2^{-|S| \cdot Fix(S)}�.
\]
The proof of Lemma 14 appears in Section 6.3. We show that $Pr(EC(S_{i,j}))$ can be represented as the sum of probabilities of EC-events for guarded set of literals. Denote the set $\{S_{i,j} \cup \{t_i, t_j\}, S_{i,j} \cup \{-t_i, t_j\}, S_{i,j} \cup \{t_i, \neg t_j\}\}$ by $POSX_{i,j}$.

**Lemma 15.**
1. Each $S \in POSX_{i,j}$ is guarded with $Fix(S) = \emptyset$.
2. $Pr(X_{i,j} = 1) = Pr(\cup_{S \in POSX_{i,j}} EC(S)) = \sum_{S \in POSX_{i,j}} Pr(EC(S))$.

**Proof.** $S$ is guarded from $C_{i,j}$ because at least one of $t_i, t_j$ occurs positively and hence the second condition of Definition 20 is satisfied. No leaf variable except $\ell_{i,j}$ occurs in $S$ hence, $S$ is guarded from the rest of the clauses according to the first condition of Definition 20. As $S$ does not contain positive literals of leaf variables, it does not contain fixed literals.

The second equality of the second statement immediately follows from Lemma 8. In light of Observation 1 for the first equality it is enough to show that $EC(S_{i,j}) = \cup_{S \in POSX_{i,j}} EC(S)$. Indeed, let $S' \in EC(S_{i,j})$. As $S'$ is a satisfying assignment of $\phi_H$, at least one variable of $C_{i,j}$ must occur positively in $S'$. As all the non-root variables of $C_{i,j}$ occur negatively in $S_{i,j}$ and hence in $S'$, at least one root variable of $C_{i,j}$ (either $t_i$ or $t_j$) must occur positively in $S'$. Therefore, there is $S \in POSX_{i,j}$ such that $S \subseteq S'$. Conversely, if $S' \in \cup_{S \in POSX_{i,j}} EC(S)$, then $S' \in EC(S)$ for some $S \in POSX_{i,j}$ and therefore $S \subseteq S'$. Now as $S_{i,j} \subseteq S$, we conclude that $S_{i,j} \subseteq S'$.

The combination of Lemma 14 and Lemma 15 allow us to compute $Pr(X_{i,j} = 1)$ in terms of $|S_{i,j}|$. It remains to impose an upper bound on $|S_{i,j}|$.

**Lemma 16.** For each pseudoedge $\{t_i, t_j\}$ of $H$, $|S_{i,j}| \leq 4 * \log m/4.9 + 5$.

**Proof.** For the purpose of this lemma, the height of a rooted tree $T$ is the largest number of vertices in a root-leaf path of $T$. We denote by $\ell(T)$ the number of leaves of $T$.

**Claim 1.** The height of a full balanced binary tree $T$ is at most $\log(\ell(T)) + 2$.

**Proof.** By definition of pseudoexpander, $|P^{1/2}_{i \rightarrow j}| \leq \log m/4.9 + 3$. The variables of $S_{i,j}$ consist of internal variables of $P^{1/2}_{i \rightarrow j}$, the internal variables of $P^{1/2}_{j \rightarrow i}$, the siblings of these two, and $\ell_{i,j}$. The number of internal vertices of $P^{1/2}_{i \rightarrow j}$ is at most $\log m/4.9 + 1$. Clearly, the same upper bound holds for the number of internal vertices of $P^{1/2}_{j \rightarrow i}$, for the number of siblings of the internal vertices of $P^{1/2}_{i \rightarrow j}$, and the number of siblings of the internal vertices of $P^{1/2}_{j \rightarrow i}$. Therefore, the total size of $S_{i,j}$ is at most $4 \times (\log m/4.9 + 1) + 1 = 4 \times \log m/4.9 + 5$.

**Proof of the first statement of Theorem 8** The lower bound on $Pr(X_{i,j} = 1)$ is obtained by the following line of reasoning.

$$Pr(X_{i,j} = 1) = \sum_{S \in POSX_{i,j}} Pr(EC(S)) = \sum_{S \in POSX_{i,j}} (1/2)^{|S|} = 3 \times (1/2)^{|S_{i,j}|+2} \geq 3 \times (1/2)^{4 \times \log m/4.9 + 7} = 3/128 \times m^{-4/4.9} \quad (20)$$
where the first equality follows from the second statement of Lemma 15, the second equality follows from the combination of the first statement of Lemma 14 and Lemma 18, the third equality follows from definition of POSX_{i,j}, the first inequality follows from Lemma 16.

In order to establish the second statement of Theorem 8, we represent \( Pr(X_{i,j} = 0) \) as the sum of probabilities of EC-events of guarded sets of literals. For this purpose, we define the set NEGX_{i,j} consisting of all the set of literals over the variables of \( S_{i,j} \) plus \( \{t_i, t_j\} \) such that each \( S \in \text{NEGX}_{i,j} \) does not falsify any clause and satisfies one of the following two conditions.

1. At least one non-root variable of \( C_{i,j} \) is assigned positively.

2. At least one sibling of an internal variable of \( C_{i,j} \) is assigned negatively.

**Lemma 17.** 1. Each \( S \in \text{NEGX}_{i,j} \) is guarded.

2. \( Pr(X_{i,j} = 0) = Pr(\bigcup_{S \in \text{NEGX}_{i,j}} \text{EC}(S)). \)

**Proof.** \( S \in \text{NEGX}_{i,j} \) assigns all the variables of \( C_{i,j} \) and does not falsify \( C_{i,j} \). This means that either one of non-leaf variables of \( C_{i,j} \) occur positively in \( S \) (and in this case the second condition of Definition 20 is satisfied) or, otherwise, the \( \ell_{i,j} \) occur positively in \( C_{i,j} \) and the rest of variables occur negatively, thus satisfying the third condition of Definition 20. Consequently, \( S \) is guarded from \( C_{i,j} \). Except \( \ell_{i,j} \), no other leaf variable occurs in \( S \), therefore \( S \) is guarded from the rest of clauses of \( \phi_H \) according to the first condition of Definition 20 confirming the first statement.

For the second statement, it is enough to show that \( \{S' | S' \in \text{SAT}(H) \land X_{i,j}(S') = 0\} = \bigcup_{S \in \text{NEGX}_{i,j}} \text{EC}(S). \)

Let \( S' \in \text{SAT}(H) \) such that \( X_{i,j}(S') = 0 \). By Observation 1, \( S_{i,j} \not\subseteq S' \). Then the projection \( S'' \) of \( S' \) to the variables of \( S_{i,j} \) plus \( \{t_i, t_j\} \) does not contain \( S_{i,j} \) either. Clearly, \( S'' \) satisfies one of the two conditions for \( \text{NEGX}_{i,j} \) and, as \( S'' \) does not falsify any clause, \( S'' \in \text{NEGX}_{i,j} \). As \( S' \in \text{EC}(S'') \), we conclude that \( S' \in \bigcup_{S \in \text{NEGX}_{i,j}} \text{EC}(S). \)

Conversely, let \( S' \in \bigcup_{S \in \text{NEGX}_{i,j}} \text{EC}(S) \). Then there is \( S'' \in \text{NEGX}_{i,j} \) such that \( S' \in \text{EC}(S'') \). By the definition of \( \text{NEGX}_{i,j} \), all the variables of \( S_{i,j} \) occur in \( S'' \) and at least one of them occurs oppositely to its occurrence in \( S_{i,j} \). Clearly, the same is preserved in \( S' \). Hence, \( S_{i,j} \not\subseteq S' \) and therefore \( X_{i,j}(S') = 0. \)

We also need one more statement on guarded sets which will be proved in Subsection 5.3.

**Lemma 18.** Let \( S_1, \ldots, S_q \) be guarded sets of literals over sets \( V_1, \ldots, V_q \) of variables such that \( V_i \cap V_j = \emptyset \) for all \( 1 \leq i, j \leq q \). Then \( Pr(\bigcap_{i=1}^q \bigcup_{S \in S_i} \text{EC}(S)) = \prod_{i=1}^q Pr(\bigcup_{S \in S_i} \text{EC}(S)) \)

**Proof of the second statement of Theorem 8.** We need to show that for any \( M' \subseteq M \) and for any set of numbers \( a_{i,j} \in \{0,1\} \) for each \( \{t_i, t_j\} \in M' \), \( Pr(\bigcap_{\{t_i, t_j\} \in M'} (X_{i,j} = a_{i,j})) = \prod_{\{t_i, t_j\} \in M'} Pr(X_{i,j} = a_{i,j}) \). Partition \( M' \) into
6.3 Proof of Lemmas for guarded sets

Lemma 19. Let $S$ be a guarded set of literals. Then the following statements hold.

1. Let $x$ be a literal of a non-leaf variable that does not occur in $S$. Then $S \cup \{x\}$ is a guarded set if literals with $\text{Fix}(S \cup \{x\}) = \text{Fix}(S)$.

2. Let $x$ be a literal of $\ell_{i,j}$ such that $\ell_{i,j}$ does not occur in $S$ and at least one variable of $C_{i,j}$ occurs positively in $S$. Then $S \cup \{x\}$ is a guarded set if literals with $\text{Fix}(S \cup \{x\}) = \text{Fix}(S)$.

3. Assume that $\ell_{i,j}$ does not occur in $S$ and the rest of variables of $C_{i,j}$ occur negatively in $S$. Then $S \cup \{\ell_{i,j}\}$ is a guarded set if literals with $\text{Fix}(S \cup \{\ell_{i,j}\}) = \text{Fix}(S)$.

Proof. Suppose that $x \notin S$ is a literal of a variable $y$ such that $S \cup \{x\}$ is not guarded while $S$ is guarded. Hence, there is a clause $C_{i,j}$ such that $S$ is guarded from $C_{i,j}$ while $S \cup \{x\}$ is not guarded from $C_{i,j}$. Clearly, this means that $y$ occurs in $C_{i,j}$, and hence not all variables of $C_{i,j}$ occur in $S$. Consequently, $S$ does not satisfy the third condition of Definition 20 regarding $C_{i,j}$ because this condition clearly requires that all the variables of $C_{i,j}$ occur in $S$. Also, note that if $S$ satisfies the second condition of Definition 20 regarding $C_{i,j}$, this condition cannot be violated by introduction of a new literal. Consequently, $S$ can satisfy the first condition only, that is some non-leaf variables of $C_{i,j}$ occur in $S$ and all their occurrences are negative. The first condition clearly cannot be violated if $y$ is a non-leaf variable. Thus $y = \ell_{i,j}$.

The reasoning in the previous paragraph immediately implies that $S \cup \{x\}$, as in the first statement, is guarded because $x$ is a literal of an internal variable and that $S \cup \{x\}$, as in the second statement, is guarded because the second condition is satisfied regarding $C_{i,j}$. For the third statement, note since all the non-leaf variables of $C_{i,j}$
occur negatively in $S$, $S \cup \{\ell_{i,j}\}$ satisfies the third condition of Definition 20 regarding $C_{i,j}$. Therefore, $S \cup \{\ell_{i,j}\}$ remains guarded.

Now, let us prove the equations concerning the fixed sets. Observe that in all the considered cases $Fix(S) \subseteq Fix(S \cup \{x\})$. Indeed, suppose $\ell_{i,j} \in Fix(S)$. This means that $\ell_{i,j} \in S$ and the rest of variables of $C_{i,j}$ occur negatively in $S$. Clearly, these occurrences are preserved in any superset of $S$.

Now, suppose that $\ell_{i,j} \in Fix(S \cup \{x\}) \setminus Fix(S)$, and hence $x = \ell_{i,j}$.

Then $\ell_{i,j} \in (S \cup \{x\}) \setminus S$. Indeed, if $\ell_{i,j} \in S$ then to keep $S$ guarded from $C_{i,j}$, the second condition of Definition 20 must be satisfied (the third condition cannot be satisfied because of the assumption that $\ell_{i,j} \notin Fix(S)$). But then one of non-leaf variables of $C_{i,j}$ occurs in $S$ (and hence also in $S \cup \{x\}$) positively blocking the possibility of $\ell_{i,j}$ being fixed. Moreover, all the internal variables of $C_{i,j}$ occur in $S$ negatively. This immediately implies that $Fix(S \cup \{x\}) \subseteq Fix(S)$ in the first case (because $x$ is not a leaf variable) and in the second case (because $S$ contains a positive occurrence of a non-leaf variable of $C_{i,j}$).

For the third case, the above reasoning implies that $Fix(S \cup \{x\}) \subseteq Fix(S \cup \{\ell_{i,j}\})$ and since by definition, $\ell_{i,j} \in Fix(S \cup \{x\})$, the opposite containment follows as well.

Proof of Lemma 14 The proof is by induction on the number of variables that do not occur in $S$. Assume first that all the variables are assigned. Note that $S$ does not falsify any clause. Indeed, if clause $C_{i,j}$ is falsified then none of the conditions of Definition 20 can be met. By definition of the probability space, $Pr(EC(S)) = 2^{-|\mathcal{V}(H) \setminus Fix(S)|} = 2^{-|S \setminus Fix(S)|}$.

Assume now that $S$ does not assign all the variables. Suppose first that some non-leaf variable $y$ does not occur in $S$.

As in any $S' \in EC(S)$ $y$ occurs either positively or negatively, $EC(S) = EC(S \cup \{y\}) \cup EC(S \cup \{-y\})$. In fact, by Lemma 8 the union is disjoint. Also, note that for each $x \in \{y, -y\}$, the following is true.

$$|S \cup \{x\} \setminus Fix(S \cup \{x\})| = |S \cup \{x\}| - |Fix(S \cup \{x\})| = |S + 1| - |Fix(S)| = |S \setminus Fix(S)| + 1 \quad (22)$$

where the first equality is correct because $Fix(S \cup \{x\}) \subseteq S \cup \{x\}$, $|S \cup \{x\}| = |S| + 1$

since $x$ does not occur in $S$, $Fix(S \cup \{x\}) = Fix(S)$ according to the first statement of Lemma 19 and the last equality is correct because $Fix(S) \subseteq S$.

Therefore,

$$Pr(EC(S)) = Pr(EC(S \cup \{y\})) + Pr(EC(S \cup \{-y\})) = 2^{-|S \cup \{y\}| \setminus Fix(S \cup \{y\})} + 2^{-|S \cup \{-y\}| \setminus Fix(S \cup \{-y\})} = 2^{-|S \setminus Fix(S)| + 1} + 2^{-|S \setminus Fix(S)| + 1} = 2^{-|S \setminus Fix(S)|} \quad (23)$$

where the second equality follows because, according to the first statement of Lemma 19 both $S \cup \{y\}$ and $S \cup \{-y\}$ are guarded and they assigned more variables than $S$, therefore the statement of this lemma holds for them according to the induction assumption. The third equality follows from (22).
Suppose now that $S$ assigns all the non-leaf variables and let $\ell_{i,j}$ be a variable not assigned by $S$. If at least one of variables of $C_{i,j}$ occurs positively in $S$, then the reasoning is analogous to the above with the second statement of Lemma 19 used instead of the first one.

Otherwise, all the non-leaf variables of $C_{i,j}$ occur in $S$ negatively. In this case $\ell_{i,j}$ occurs positively in any element of $EC(S)$ (for otherwise $C_{i,j}$ is falsified). That is, $EC(S) = EC(S \cup \{\ell_{i,j}\})$. Therefore

$$Pr(EC(S)) = Pr(EC(S \cup \{\ell_{i,j}\})) = \frac{1}{2}|S \cup \{\ell_{i,j}\}| - |Fix(S \cup \{\ell_{i,j}\})| =$$

$$\frac{1}{2}|S| - |Fix(S)| = \frac{1}{2}|S| \sum_{i=1}^{q} |Fix(S_i)| \tag{24}$$

where the second equality follows from the third statement of Lemma 19 and the induction assumption, the third equality follows because $Fix(S \cup \{\ell_{i,j}\}) \subseteq S \cup \{\ell_{i,j}\}$. For the fourth equality, note that $|S \cup \{\ell_{i,j}\}| = |S| + 1$ as $\ell_{i,j} \notin S$ and that by the third statement of Lemma 19, $Fix(S \cup \{\ell_{i,j}\}) = Fix(S) \cup \{\ell_{i,j}\}$ and hence $|Fix(S \cup \{\ell_{i,j}\})| = |Fix(S)| + 1$. The last equality follows because $Fix(S) \subseteq S$. □

**Lemma 20.** Let $S_1, \ldots, S_q$ be guarded sets of literals over pairwise disjoint sets of variables. Then $Pr(EC(\bigcup_{i=1}^{q} S_i)) = \prod_{i=1}^{q} Pr(EC(S_i))$.

**Proof.** Let us show that $S = \bigcup_{i=1}^{q} S_i$ is guarded. Indeed, consider a clause $C_{j,k}$. If $\ell_{j,k}$ does not occur in $S$, then $S$ satisfies the first condition of Definition 20. Otherwise, there is $S_i$ such that $\ell_{j,k}$ occurs in $S_i$. As $S_i$ is guarded from $C_{j,k}$, either the second or the third condition is satisfied for $S_i$ regarding $C_{j,k}$. It is not hard to see that both these conditions are preserved in every superset of $S_i$.

By Lemma 14 and also taking into account that $Fix(S) \subseteq S$ and that $S_1, \ldots, S_q$ are mutually disjoint, we obtain the following.

$$Pr(EC(S)) = \frac{1}{2}|S| - |Fix(S)| = \frac{1}{2}|S| \sum_{i=1}^{q} |S_i| - |Fix(S_i)| \tag{25}$$

We next show that $Fix(S) = \bigcup_{i=1}^{q} Fix(S_i)$. As each fixed element of $F_i$ remains fixed in a superset of $F_i$, $\bigcup_{i=1}^{q} Fix(S_i) \subseteq Fix(S)$.

Conversely, suppose that there is $\ell_{j,k} \in Fix(S) \setminus \bigcup_{i=1}^{q} Fix(S_i)$. As $\ell_{j,k} \in S$, there is $i$ such that $\ell_{j,k} \in S_i$. Since $\ell_{j,k} \notin Fix(S_i)$, $S_i$ does not satisfy the third condition of Definition 20 regarding $C_{j,k}$. To remain guarded from $C_{j,k}$, $S_i$ must satisfy the second condition of Definition 20 regarding $C_{j,k}$. That is, a non-leaf variable of $C_{j,k}$ occurs positively in $S_i$ and hence also in $S$, in contradiction to $\ell_{j,k} \in Fix(S)$, thus confirming that $Fix(S) \subseteq \bigcup_{i=1}^{q} Fix(S_i)$.

Due to the pairwise disjointness of $S_1, \ldots, S_q$, $Fix(S_1), \ldots, Fix(S_q)$ are pairwise disjoint (as being subsets of $S_1, \ldots, S_q$ respectively.) Therefore, $|Fix(S)| = \sum_{i=1}^{q} |Fix(S_i)|$. Substituting this equality into the right-most item of (25), we obtain
\[
Pr(\text{EC}(S)) = (1/2)^{(|S| - \sum_{i=1}^{q} |\text{Fix}(S_i)|)} = (1/2)^{\sum_{i=1}^{q}(|S_i| - |\text{Fix}(S_i)|)} = \\
(1/2)^{\sum_{i=1}^{q} |S_i \setminus \text{Fix}(S_i)|} = \prod_{i=1}^{q} \frac{1}{2} |S_i \setminus \text{Fix}(S_i)| = \prod_{i=1}^{q} Pr(\text{EC}(S_i))
\]

where the second equality follows because \(\text{Fix}(S_i) \subseteq S_i\) and the last equality follows from Lemma 14. \[\blacksquare\]

**Proof of Lemma 18** The lemma is proved through the following line of reasoning.

\[
Pr\left(\bigcap_{i=1}^{q} \bigcup_{S_i \in S_i} \text{EC}(S_i)\right) = Pr\left(\bigcup_{(S_1,\ldots,S_q) \in S_1 \times \cdots \times S_q} \bigcap_{i=1}^{q} \text{EC}(S_i)\right) = \\
Pr\left(\bigcup_{(S_1,\ldots,S_q) \in S_1 \times \cdots \times S_q} \text{EC}\left(\bigcup_{i=1}^{q} S_i\right)\right) = \sum_{(S_1,\ldots,S_q) \in S_1 \times \cdots \times S_q} Pr\left(\text{EC}\left(\bigcup_{i=1}^{q} S_i\right)\right) = \\
\prod_{i=1}^{q} Pr\left(\bigcup_{S_i \in S_i} \text{EC}(S_i)\right)
\]

For the second equality, we demonstrate that for any set \(S_1,\ldots,S_q\) of literals, \(\text{EC}\left(\bigcup_{i=1}^{q} S_i\right) = \bigcap_{i=1}^{q} \text{EC}(S_i)\). If \(S \in \text{EC}\left(\bigcup_{i=1}^{q} S_i\right)\) then \(S_i \subseteq S\) for \(1 \leq i \leq q\), thus \(S \in \text{EC}(S_i)\) for \(1 \leq i \leq q\). Conversely, if \(S \in \bigcap_{i=1}^{q} \text{EC}(S_i)\), then \(S_i \subseteq S\) for all \(1 \leq i \leq q\). That is \(\bigcup_{i=1}^{q} S_i \subseteq S\) and hence \(S \in \text{EC}\left(\bigcup_{i=1}^{q} S_i\right)\). Note that since the elements of both sets are subsets of \(\text{SAT}(H)\), there is no need to explicitly mention that the \(S\) being considered is a satisfying assignment of \(\phi_H\).

The third equality follows from Lemma 8 because for two distinct tuples \((S_1,\ldots,S_q)\), \(\bigcup_{i=1}^{q} S_i\) are distinct sets of literals over the same set of variables. The fourth equality follows from Lemma 20. The last equality follows from Lemma 8 applied individually to each \(S_i\). \[\blacksquare\]

**References**

[1] Noga Alon and Wolfgang Maass. Meanders and their applications in lower bounds arguments. *J. Comput. Syst. Sci.*, 37(2):118–129, 1988.

[2] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudo-random generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[3] Allan Borodin, Alexander A. Razborov, and Roman Smolensky. On lower bounds for read-k-times branching programs. *Computational Complexity*, 3:1–18, 1993.
[4] Anna Gáll. A simple function that requires exponential size read-once branching programs. *Inf. Process. Lett.*, 62(1):13–16, 1997.

[5] Stasys Jukna. A note on read-k times branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(27), 1994.

[6] Stasys Jukna. A nondeterministic space-time tradeoff for linear codes. *Inf. Process. Lett.*, 109(5):286–289, 2009.

[7] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012.

[8] Stasys Jukna and Georg Schnitger. Triangle-freeness is hard to detect. *Combinatorics, Probability & Computing*, 11(6):549–569, 2002.

[9] Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[10] Igor Razgon. On the read-once property of branching programs and cnfs of bounded treewidth. *Algorithmica*, To appear, 2015.

[11] Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM Monographs on Discrete Mathematics and applications, 2000.

A Simulation of a $d$-NBP by a uniform $d$-NBP

Let $Z$ be a $d$-NBP and let $v_1,\ldots,v_n$ be its vertices listed in the topological order. For a variable $x$ of $F(Z)$ and a node $v$ of $Z$, denote by $n_Z(v,x)$ the largest number of occurrences of variable $x$ on a path of $Z$ from the source to $v$. We introduce a sequence of NBPs $Z_1,\ldots,Z_n$ defined as follows. $Z_1 = Z$.

For $i > 1$, $Z_i = Z_{i-1}$ if $v$ has only one in-coming edge. Otherwise, $Z_{i-1}$ is transformed into $Z_i$ by the following process. First, whenever there is an edge $(u,v)$ of $Z_{i-1}$ that is labelled subdivide this edge by introducing a new vertex $w$ and replacing $(u,v)$ with edges $(u,w)$ and $(w,v)$ with $(u,w)$ being labelled by the label of $(u,v)$ and $(w,v)$ being unlabelled. This way we obtain an NBP $Z'_i$ where $v$ has the same number of in-coming edges as in $Z_{i-1}$ however all in-coming edges of $v$ are unlabelled.

The second stage of transformation ensures that for each incoming edge $(u,v)$ of $Z'_i$ and for each variable $x$, the largest number of occurrences of $x$ on a path from the sink to $v$ that goes through $(u,v)$ is $n_{Z_{i-1}}(v,x)$. To this end, for each edge $(u,v)$ of $Z'_i$, let $x_1,\ldots,x_q$ be the set of variables such that for each $1 \leq i \leq q$, $n_{Z'_i}(u,x_i) < n_{Z_{i-1}}(v,x_i)$. Let $m_1,\ldots,m_q$ be such that $m_i = n_{Z_{i-1}}(v,x_i) - n_{Z'_i}(u,x_i)$ for $1 \leq i \leq q$. Introduce new vertices $v_{1,1},\ldots,v_{1,m_1},\ldots,v_{q,1},\ldots,v_{q,m_q}$. Replace the edge $(u,v)$ by the following edges.

- Two edges from $v$ to $v_{1,1}$ one labelled with $x_1$, the other labelled with $\neg x_1$.
- For $1 < j \leq m_1$, two edges from $v_{1,j-1}$ to $v_{1,j}$, one labelled with $x_1$, the other labelled with $\neg x_1$.
• For each $1 < i \leq q$, two edges from $v_{i-1,m_{i-1}}$ to $v_{i,1}$ one labelled with $x_i$, the other labelled with $\neg x_i$.

• For each $1 < i \leq q$ ad each $1 < j \leq m_i$, two edges from $v_{i,j-1}$ to $v_{i,j}$ one labelled with $x_i$, the other labelled with $\neg x_i$.

• Two edges from $v_{q,m_q}$ to $v$ one labelled with $x_q$, the other labelled with $\neg x_q$.

Arguing by induction on $v_1, \ldots, v_n$, the following statements can be made.

1. Each $Z_i$ is a $d$-NBP.

2. For each $Z_i$ and each variable $x$, all the paths from the source to $v_i$ have the same number of occurrences of $x$ which is $n_{Z_i}(v_i, x)$.

3. All the $Z_i$ compute the same function.

For $i = 1$, all three statements follow from definition of $Z$ so we assume $i > 1$.

Next, we are going to show that for each in-neighbour $u$ of $v$ in $Z_i'$ and for each variable $x$, the number of occurrences of $x$ on each path of $Z_i'$ from the source to $u$ is exactly $n_{Z_i'}(u, x)$.

Indeed, assume first that $u$ is a vertex of $Z_{i-1}$ (that is, it has not been introduced in $Z_i'$ as a result of subdivision). Then $u = v_j$ for some $j < i$. Indeed, otherwise, $u$ is a new vertex that has been introduced during construction of some $Z_j$ for $j < i$. Then only vertex of $Z_j$ such a new vertex can be an in-neighbour at the stage of its introducing is $v_j$. Moreover, such a vertex cannot have an out-going edge to $v_i$ introduced at a latter stage because one end of such edge must be a new vertex introduced at a latter stage (which is not $u$ that has been introduced during construction of $Z_j$ and not $v_i$ that exists in $Z_1$). Thus, in this case, the desired statement is correct by the induction assumption. The last statement requires further explanation. The induction assumption, as such, states that the number of occurrences of $x$ on each path of $Z_j$ from the source to $v_j$ is $n_{Z_j}(v_j, x)$. In particular, the induction assumption says nothing about the paths from the source to $v_j$ in $Z_{i'}$ for $i' > j$. However, the paths from the source to $v_i$ are not affected by the subsequence transformations of $Z_j$ into $Z_{j+1}$ and so on, hence the induction assumption remains invariant and, moreover, $n_{Z_j}(v_j, x) = n_{Z_{i'}}(v_j, x)$ for all $i' > j$. As paths from the source to $u$ are not affected by a transformation from $Z_{i-1}$ to $Z_i'$, $n_{Z_i}(v_j, x) = n_{Z_{i'}}(v_j, x)$ as required.

Otherwise, if $u$ is not a vertex of $Z_{i-1}$, there is an in-neighbour $u'$ of $v_i$ such that $u$ is introduced as a result of subdivision of an edge $e$ from $u'$ to $v_i$. By the previous paragraph $u' = v_j$ for some $j < i$. Then, by the induction assumption, the number of occurrences of $x$ on any path of $Z_j$ from the source to $u'$ is $n_{Z_{i-1}}(v_j, x)$ (the reasoning is as in the previous paragraph). Any path of $Z_i'$ from the source to $u$ is a path from the source to $u'$ plus the edge $(u', u)$. If the variable labelling $(u', u)$ is not $x$ then the number of occurrences on any such path is $n_{Z_{i-1}}(v_j, x)$, otherwise it is $n_{Z_{i-1}}(v_j, x) + 1$. In both cases, the number of occurrences on any path of $Z_i'$ from the source to $u$ is invariant and hence equals $n_{Z_i'}(u, x)$.

Note that $n_{Z_i'}(u, x) \leq n_{Z_i}(v_i, x)$ because it is just the largest number of occurrences of $x$ on a path of $Z_{i-1}$ from the source to $v_i$ going through a particular edge.
((u, v) or (u', v) depending on whether this edge is subdivided by transformation from $Z_{i-1}$ to $Z_i$). If $n_{Z_i}(u, x) = n_{Z_{i-1}}(v_i)$ then $x$ does not occur on a path between $u$ and $v_i$ in $Z_i$. Therefore the number of occurrences of $x$ on a path of $Z_i$ from the source to $v_i$ is as in $Z_i$ which is $n_{Z_i}(u, x) = n_{Z_{i-1}}(v_i, x)$. Otherwise, there is $1 \leq j \leq q$ such that $x = x_j$ and on each path of $Z_i$ from $u$ to $v$ there are $m_j = n_{Z_{i-1}}(v_i, x) - n_{Z_i}(u, x)$ occurrences of $x$. That is the total number of occurrences on each path of $Z_i$ from the source to $v_i$ passing through $u$ is $n_{Z_i}(u, x) + m_j = n_{Z_{i-1}}(v_i, x)$. Since $u$ is selected as an arbitrary in-neighbour of $v_i$ in $Z_i$, there are $n_{Z_{i-1}}(v_i, x)$ occurrences of $x$ on any path from the source to $v_i$. This confirms the second statement.

For the first statement, it is sufficient to verify that on any source-sink path of $Z_i$ going through $v_i$ each variable $x$ occurs at most $d$ times. Indeed, on each $v_i$-sink path $Q$ of $Z_{i-1}$ each variable $x$ occurs at most $d - n_{Z_{i-1}}(v_i, x)$ times. Otherwise, take a path $P$ of $Z_{i-1}$ from the source to $v_i$ on which there are $n_{Z_{i-1}}(v_i, x)$ occurrences of $Z_i$ (such a path exists by definition of $n_{Z_{i-1}}(v_i, x)$) and let $Q$ be a path of $Z_{i-1}$ from $v_i$ to the sink having more that $d - n_{Z_{i-1}}(v_i, x)$ occurrences of $x$. Then $P + Q$ is source-sink path of $Z_{i-1}$ with more than $d$ occurrences of $x$ in contradiction to the first statement holding regarding $Z_{i-1}$ by the induction assumption. Since the paths from $v_i$ to sink are not affected by transformation from $Z_{i-1}$ to $Z_i$, we conclude that on each $v_i$-sink path $Q$ of $Z_i$, each variable $x$ occurs at most $d - n_{Z_{i-1}}(v_i, x)$ times. Together with the second statement, this implies that on each source sink path of $Z_i$ going through $v_i$ each variable occurs at most $d$ times thus confirming the first statement.

Let us verify the third statement, in particular, let us show that $Z_{i-1}$ and $Z_i$ compute the same function. Let $S$ be a satisfying assignment of $F(Z_{i-1})$. This means that $Z_{i-1}$ has a computational path $P$ with $A(P) \subseteq S$. If $P$ does not contain $v_i$ then $P$ is a computational path of $Z_i$ and hence $S$ is a satisfying assignment of $Z_i$. Otherwise, let $u$ be the vertex preceding $v_i$ in $P$. If $u$ is an in-neighbour of $v_i$ in $Z_i$ then, again, $P$ is a computational path of $Z_i$. Otherwise, let $P_1$ be the prefix of $P$ ending at $u$ and let $P_2$ be the suffix of $P$ beginning at $v_i$. By construction, both $P_1$ and $P_2$ are paths of $Z_i$. Then choose a path $P_0$ between $u$ and $v$ in $Z_i$, so that each variable labelling the path has the same occurrence as in $S$ (this possible to do due to the 'parallel edges' construction as described above). Then $P_1 + P_0 + P_2$ is a path of $Z_i$ such that $A(P_1 + P_0 + P_2) \subseteq S$.

Conversely, let $S$ be a satisfying assignment of $Z_i$ and let $P$ be a computational path of $Z_i$ with $A(P) \subseteq S$. It is sufficient to consider the case where $P$ is not a computational path of $Z_{i-1}$. Then $P$ contains new vertices of $Z_i$ and hence contains $v_i$ (because any path from a new vertex to the sink goes through $v_i$). Moreover, the immediate predecessor of $v_i$ in $P$ is a new vertex of $Z_i$. Let $u$ be the last in-neighbour of $v_i$ in $Z_{i-1}$ preceding $v_i$ in $P$. Let $P'$ be obtained from $P$ by replacing the $u \rightarrow v_i$ subpath of $P$ with the $(u, v_i)$ edge of $Z_{i-1}$ carrying the same label as in $Z_{i-1}$. Then $P'$ is a computational path by construction $A(P') \subseteq A(P)$ (in particular, the process of transformation of edge $(u, v_i)$ into a path retains the label of $(u, v_i)$ on any resulting $u \rightarrow v_i$ path and hence $A(P') \subseteq S$).

It follows that $Z_n$ computes the same function as $Z$ and each variable $x$ occurs the same number of times $n_x \leq k$ on each computational path of $Z_n$. For those variables $x$ where $n_x < k$ add $k - n_x$ entries on each in-coming edge of the sink using the same subdivision technique as was used for transformation from $Z_{i-1}$ to $Z_i$. Then, by the analogous reasoning it can be verified that the resulting NBP $Z^*$ becomes uniform.
$k$-NBP computing the same function as $Z$.

Let us calculate the size of $Z^*$ in terms of the size of $Z$. Note that for the purpose of this proof $|Z|$ is denoted by $n$. The number of variables cannot be more than the number of edges of $Z$ therefore is bounded by $O(n^2)$. On each of the $n$ iterations, at most $n$ edges are transformed (an in-degree of a node cannot be larger than $n$) and on each node the number of copies added is bounded by the number of variables multiplied by $k$ that is $O(n^2) + k$. Therefore, the size of $Z^*$ is $O(n^4 + k)$. (Note that the size increase of the final transformation from $Z_n$ to $Z^*$ is dominated by this complexity and so can be safely ignored).

\section{Proof of Theorem \ref{thm:main}}

\subsection{Constructively created CNFs}

We will next define a set of graphs $G_m$. Then we will define a set of CNFs $\varphi(G_m)$ with graphs $G_m$. The CNFs $\varphi(G_m)$ is the class for which we will prove the lower bound stated in the theorem.

Let $T_1^*, \ldots, T_m^*$ be a set binary rooted trees with $\lfloor m^{1/4.9} \rfloor$ leaves each and height at most $\log m/4.9 + 2$. Such trees can be created as follows. Let $a \geq m^{1/4.9}$ be the nearest to $m^{1/4.9}$ integer which is a power of two. Take a complete binary tree $T'$ with $a$ leaves. Then choose an arbitrary set $L$ of $\lfloor m^{1/4.9} \rfloor$ leaves and let each $T^*_i$ be obtained from $T'$ by taking the union of all root-leaf trees ending at $L$. The height of $T'$ is $\log a + 1$ and, as $a \leq 2 \cdot m^{1/4.9}$, the height does not exceed $\log m/4.9 + 2$.

Then for each $1 \leq i \neq j \leq m$ and each pair of leaves $\ell_1$ of $T_i$ and $\ell_2$ of $T_j$, introduce a new vertex $v$ and connect it by edges $\ell_1$ and $\ell_2$. We call these new vertices $v$ subdivision vertices. The graph $G_m$ consists of the union $T_1^*, \ldots, T_m^*$ plus the subdivision vertices together with the edges incident to them.

The variables of $\varphi(G_m)$ are the vertices of $G_m$. The clauses correspond to the subdivision vertices. Let $v$ be a subdivision vertex. Then the clause $C_v$ corresponding to it is created as follows. Let $\ell_1$ and $\ell_2$ be two leaves of trees $T_i^*$ and $T_j^*$ incident to $v$. Let $P_1$ and $P_2$ be respective root-leaf paths of $T_i^*$ and $T_j^*$ ending with $\ell_1$ and $\ell_2$. The literals of $C_v$ are $V(P_1) \cup \{v\} \cup V(P_2)$.

It is not hard to see that $\varphi(G_m)$ can be created by a deterministic procedure taking polynomial time in the number of variables.

In the next subsection we formally prove that for a sufficiently large $m$, it is possible to assign a subset of subdivision vertices with true so that the remaining CNF is $\phi_H$ where $H$ is a pseudoexpander with $m$ roots. Then, in Section B.3, we will show that the lower bound for $\phi_H$ implies that lower bound for $\varphi(G_m)$.

\subsection{Extraction of CNFs based on pseudoexpanders}

Let $F$ be a Boolean function and let $S$ be a partial assignment to a subset of its variables. Then $F|_S$ is a function on variables not assigned by $S$ and $S'$ is a satisfying assignment of $F|_S$ if and only if $S \cup S'$ is a satisfying assignment of $F$. 

\pagebreak
If $\varphi$ is a monotone CNF and $S$ consists of positive literals of a subset of variables of $\varphi$ then $\varphi|_{S}$ is obtained by removal of clauses where the variables of $S$ occur. In this section we are going to prove the following lemma.

**Lemma 21.** Let $m$ be an integer and suppose that there exist a pseudoexpander $H$ with $m$ roots and max pseudodegree $m^{1/4.9}$. Then there is a set $S$ of positive literals of subdivision vertices of $\varphi(G_m)$ such that $\varphi(G_m)|_S$ is $\phi_H$, where $H'$ is a subgraph of $G_m$ which is a pseudoexpander with $m$ roots and with $U(H)$ isomorphic to $U(H')$.

**Proof.** Let $\text{Roots}(H) = \{t_1, \ldots, t_m\}$ and let $t'_1, \ldots, t'_m$ be the respective roots of the trees $T'_1, \ldots, T'_m$ of $G_m$ (let us call them the roots of $G_m$). Recall that by $U(H)$ we denote the underlying graph of $H$. For $1 \leq i \leq m$, denote by $I_i$ the set of $j$ such that $t_i$ is adjacent to $t_j$ in $U(H)$. Note that since the max-degree of $U(H)$ is at most $m^{1/4.9}$, the number of leaves of $T_i'$ is at least $|I_i|$. For each $i$, fix an arbitrary subset of $|I_i|$ leaves of $T_i'$ and mark them with the indices of $I_i$, each index marking one leaf.

Now consider a subdivision vertex $v$ of $G_m$ connecting a leaf $\ell_1$ of $T_i$ and a leaf $\ell_2$ of $T_j$. Let us call $v$ meaningful if $\ell_1$ is marked with $j$ and $\ell_2$ is marked with $i$. Let $S$ be the set consisting of all non-meaningful subdivision vertices and let $\varphi' = \varphi(G_m)|_S$.

For $1 \leq i \leq m$, let $T'_i$ be the rooted tree obtained from $T_i'$ by taking the union of all root-leaf paths of $T_i'$ that end with marked leaves. We prove the following statements about trees $T'_i$.

1. Each leaf of each $T'_i$ is incident to exactly one meaningful vertex.
2. For each edge $\{t_i, t_j\}$ of $U(H)$, there is exactly one meaningful vertex adjacent to both $T'_i$ and $T'_j$.
3. If $t_i$ and $t_j$ are not adjacent in $U_H$ then there is no meaningful vertex adjacent to both $T'_i$ and $T'_j$.

Indeed, let $\ell_1$ be a leaf of $T'_i$. By definition of $I_i$, $\ell_1$ is marked with a $j$ such that $\{t_i, t_j\}$ is an edge of $U(H)$. Consequently, by definition of $I_j$, $T'_j$ has a leaf $\ell_2$ marked with $i$. By definition of $G_m$, there exists a subdivision vertex $v$ incident to both $\ell_1$ and $\ell_2$. By definition of meaningful vertices, $v$ is meaningful. Assume that $\ell_1$ is incident to another meaningful vertex $u \neq v$. Then, by definition of $G_m$, $u$ connects $\ell_1$ to a leaf $\ell_3 \neq \ell_2$ of $T_j$. By definition of a meaningful vertex, $\ell_3$ is marked with $i$ in contradiction to the procedure of marking of $T'_j$ that does not assign two different leaves with the same element of $I_j$. Thus we have proved the first statement.

Let $\{t_i, t_j\}$ be an edge of $U(H)$. Then, by definition of $I_i$ and $I_j$, $T'_i$ has a leaf $\ell_1$ marked with $j$ and $T'_j$ has a leaf $\ell_2$ marked with $i$. The subdivision vertex $v$ connecting $\ell_1$ and $\ell_2$ is a meaningful vertex adjacent to $T'_i$ and $T'_j$. Suppose there is another meaningful vertex $u$ adjacent to both $T'_i$ and $T'_j$. Then $u$ is a subdivision vertex of $G_m$ adjacent to a leaf $\ell_3$ of $T'_i$ and a leaf $\ell_4$ of $T'_j$. Since $u \neq v$, by construction of $G_m$, either $\ell_3 \neq \ell_1$ or $\ell_4 \neq \ell_2$. Assume the former w.l.o.g. Then $\ell_3$ cannot be marked with $j$ and hence $u$ is not meaningful, a contradiction. Thus we have proved the second statement.
For the third statement, assume by contradiction that there are $T'_i$ and $T'_j$ such that, on the one hand $\{t_i, t_j\}$ is not an edge of $U(H)$ and, on the other hand, there is a meaningful vertex $v$ adjacent to both $T'_i$ and $T'_j$. Let $\ell_1$ be the leaf of $T'_i$ adjacent to $v$. Then $\ell_1$ is marked with $j$ implying that $j \in I_v$, a contradiction proving the third statement.

For $1 \leq i \leq m$, let $T''_i$ be the rooted tree having the same root $t'_i$ as $T'_i$ and obtained from $T'_i$ by adding to each leaf $\ell$ of $T'_i$ the edge $\{\ell, v\}$ where $v$ is the meaningful vertex adjacent to $\ell$ according to the first statement above. Let $H'$ be the union of $T''_1, \ldots, T''_m$. We are going to prove that $H' \in \mathsf{BTB}$ with $T(H') = \{T''_1, \ldots, T''_m\}$ with $U(H')$ isomorphic to $U(H)$.

That $T''_1, \ldots, T''_m$ are all extended follows by construction. Let $v$ be a leaf of $T''_i$ and let us show that $v$ is a leaf of exactly one other tree $T''_j$. Let $\ell_1$ be the only neighbour of $v$ in $T''_i$. By definition, $\ell_1$ is a leaf of $T''_i$. Let $j$ be the mark of $\ell_1$. By definition of $v$ as a meaningful vertex, it is connected to a leaf $\ell_2$ of $T'_j$ marked with $i$. We claim that $v$ is a leaf of $T_j$. Indeed, let $u$ be the leaf of $T''_j$ adjacent to $\ell_2$ (such a leaf exists by construction). Then $u$ is a meaningful vertex adjacent to $\ell_2$. By the first statement above, there is only one meaningful vertex adjacent to $\ell_2$ and hence $u = v$. Note that $T''_i$ and $T''_j$ can only have joint leaves as $T'_i$ and $T'_j$ are vertex disjoint. Moreover, by construction, a joint leaf of $T''_i$ and $T''_j$ is a meaningful vertex adjacent to both $T'_i$ and $T'_j$. By the third statement, there can be at most one such a meaningful vertex, confirming that $H' \in \mathsf{BTB}$. The third statement in fact claims that such a vertex exists if and only if $\{t_i, t_j\}$ is an edge of $U(H)$. That is $T''_i$ and $T''_j$ have a joint leaf if and only if $\{t_i, t_j\}$ is an edge of $U(H)$ establishing a natural isomorphism between $U(H')$ and $U(H)$ with $t'_i$ corresponding to $t_i$.

Note that the height of each $T''_i$ does not exceed the height of $T'_i$ and hence is at most $\log m/4.9 + 2$. Hence, the height of each $T''_i$ is at most $\log m/4.9 + 3$. Consequently, $H'$ is a pseudoexpander.

It remains to show that $\phi(G_m)|_S = \phi_i(H')$. By definition, the clauses of $\phi(H')$ correspond to the edges $U(H')$. Consider the clause $C_{i,j}$ corresponding to an edge $\{t'_i, t'_j\}$ of $H'$. Let $v$ be the joint leaf of $T''_i$ and $T''_j$. By definition, $C_{i,j}$ contains the vertices of the root-leaf path of $T''_i$ ending with $v$ and the vertices of the root-leaf path of $T''_j$ ending with $v$. Recall that $v$ is a subdivision vertex of $G_m$ and note that $C_{i,j}$ is exactly the clause $C_v$ of $\phi(G_m)$. As $v$ is a meaningful vertex, it does not belong to $S$ and hence $C_v$ is a clause of $\phi(G_m)|_S$.

Conversely, let $C_v$ be a clause of $\phi(G_m)|_S$. Then $v$ is a meaningful vertex. That is, $v$ is adjacent to respective leaves $\ell_1$ and $\ell_2$ of some trees $T'_i$ and $T'_j$. By construction and the second statement above, $v$ is the joint leaf of $T''_i$ and $T''_j$ and, by definition of $C_v$, it is exactly $C_{i,j}$. ■

### B.3 Proof of Theorem 5

Let $S$ be an assignment to the variables of $\phi(G_m)$. Then for any fixed $d$, the size of smallest $d$-NBP computing $\phi(G_m)$ is greater than or equal to the size of the smallest $d$-NBP computing $\phi(G_m)|_S$. The reason for this is that an $d$-NBP $Z$ computing $\phi(G_m)$ can be transformed into a $d$-NBP computing $\phi(G_m)|_S$ without increase of size. Indeed, for each edge $e$ of $Z$ labelled with a literal $x$ of a variable of $S$, remove the label if
\( x \in S \) or remove the edge \( e \) if \( \neg x \in S \). Let \( Z' \) be an \( \text{NBP} \) obtained by the union of all source-sink paths of \( S \) where no edge has been removed and the removal of labels as above preserved. Then \( Z' \) computes \( \varphi(G_m)|_S \). Indeed, let \( S' \) be such that \( S \cup S' \) is a satisfying assignment of \( \varphi(G_m) \). Let \( P \) be a computational path of \( Z \) with \( A(P) \subseteq S \cup S' \). No edges of \( P \) are removed and all the labels of \( S \) are removed in \( Z' \) so \( P \) remains a computational path of \( Z' \) and the assignment on \( P \) in \( Z' \) is a subset of \( S' \), hence \( S' \) is a satisfying assignment of the function computed by \( Z' \).

Conversely, let \( S' \) be a satisfying assignment of the function computed by \( Z' \). That is \( Z' \) has a computational path \( P' \) with \( A(P') \subseteq S' \). Then \( P' \) is a path of \( Z \) and additional labels on \( P' \) are all elements of \( S \) (they do not contain occurrences opposite to \( S' \)) and do not contain two opposite occurrences. That is \( P' \) is a computational path of \( Z \) with an assignment being a subset of \( S \cup S' \) and hence \( S' \) is a satisfying assignment of \( \varphi(G_m) \).

By Theorem 1 for each sufficiently large \( m \) there is a pseudoexpander \( H_m \) with \( m \) roots and max-degree of the underlying graph at most \( m^{1/4.9} \). Then it follows from Lemma 1 that for each sufficiently large \( m \) there is an assignment \( S_m \) to the variables of \( \varphi(G_m) \) such that \( \varphi(G_m)|_{S_m} = \phi(H_m') \), where \( H_m' \) is a pseudoexpander with \( m \) roots. By Theorem 4 computing \( \varphi(G_m)|_{S_m} \) requires \( d\text{-NBP} \) of exponential size in \( m \). Since, as shown above, the size of \( d\text{-NBP} \) needed to compute \( \varphi(G_m) \) is at least as large and the number of variables of \( \varphi(G_m) \) is polynomial in \( m \), a \( d\text{-NBP} \) of exponential size is required to compute \( \varphi(G_m) \). \( \blacksquare \)

\section*{C Proof of auxiliary lemmas for Theorem 2}

\textbf{Proof of Lemma 5} We are going to prove the following statement. Let \( P \) be a computational path of \( Z \) on which each variable occurs exactly \( d \) times where \( d \leq \log m/40000 \). Then \( P \) contains a set \( X \) of vertices of size at most \( \log m/4000 \) that separates two subsets of \( \text{Roots}(H) \) of size at least \( m^{0.999}/3 \).

Let us verify that the lemma follows from the above statement. By definition of \( H \), \( n \leq 3m^2 \). Therefore, \( \log n/10^5 \leq \log 3m^2/10^5 = (\log m/50000) + 3 \leq \log m/40000 \) for sufficiently large \( m \). Therefore, the above mentioned set \( X \) of vertices can be found for \( d \leq \log n/10^5 \) and as \( m \leq n \), the size of this set is at most \( \log n/4000 \), as required by the lemma.

\textbf{Definition of an interval.} For an arbitrary sequence \( s_1, \ldots, s_b \), and \( 1 \leq i \leq q \leq b \), let us call the subsequence \( s_i, s_{i+1}, \ldots s_q \) of consecutive elements of \( s_1, \ldots, s_b \) an \textit{interval} of \( s_1, \ldots, s_b \).

The key observation for this proof is the following claim.

\textbf{Claim 2.} Let \( \text{SEQ} \) be a sequence of elements of \( \{1, \ldots, m\} \) where each element appears at most \( m/40000 \) times. Then, for a sufficiently large \( m \), there is a partition of \( \text{SEQ} \) into intervals \( \text{SEQ}_1, \ldots, \text{SEQ}_{c+1} \), where \( c \leq \log m/4000 \) and two disjoint subsets \( U_1 \) and \( U_2 \) of size at most \( m^{0.999} \) each such that one of the following two statements is true.

\begin{enumerate}
\item Elements of \( U_1 \) occur only in \( \text{SEQ}_i \) with an odd \( i \) and elements of \( U_2 \) occur only in \( \text{SEQ}_i \) with an even \( i \).
\end{enumerate}
2. Elements of $U_1$ occur only in $SEQ_i$ with an even $i$ and elements of $U_2$ occur only in $SEQ_i$, with an odd $i$.

Let us show how the lemma follows from Claim [2]. Let $P$ be a computational path of $Z$. Let $u_1, \ldots, u_a$ be the sequence of literals of variables of $\text{Roots}(H)$ appearing in the order as they occur along $P$. Let $SEQ^* = v_1, \ldots, v_b$ be the sequence where each $v_i$ is $u_i$ if $u_i$ is the positive literal and $\neg u_i$ otherwise. That is, $SEQ^*$ is nothing else than a sequence of elements of $\text{Roots}(H)$. As $|\text{Roots}(H)| = m$ and each element occurs at most $\log m/40000$ times (see the second paragraph of the proof for justification), Claim [2] applies to $SEQ^*$. In particular, there is a partition $SEQ^*_1, \ldots, SEQ^*_c+1$ of $SEQ^*$ into intervals, where $c \leq \log m/4000$ and two disjoint subsets $U_1$ and $U_2$ of $\text{Roots}(H)$ such that at least one of the two statements of Claim [2] happens with $SEQ^*$ replacing $SEQ$.

Let $i_1, \ldots, i_c$ be such that for $1 \leq j \leq c$, $v_{i_j}$ is the last element of $SEQ^*_j$. It is not hard to see that $P$ can be partitioned into subpaths $P_1, \ldots, P_{c+1}$ such that the sequence of occurrences of $\text{Roots}(H)$ on $P_1$ is $u_1, \ldots, u_{i_1}$ and for each $1 < j \leq c + 1$, the sequence of occurrences of $\text{Roots}(H)$ on $P_j$ is $u_{i_{j-1}+1}, \ldots, u_{i_j}$. Consequently one of the following two statements is true.

1. Elements of $U_1$ occur only in $P_i$ with an odd $i$ and elements of $U_2$ occur only in $P_i$ with an even $i$.

2. Elements of $U_1$ occur only in $P_i$ with an even $i$ and elements of $U_2$ occur only in $P_i$ with an odd $i$.

Therefore, the set $x_1, \ldots, x_c$ of respective ends of $P_1, \ldots, P_{c+1}$ separates $U_1$ and $U_2$ as required.

**Proof of Claim** [2] An interval $S$ of $SEQ$ is a link between two disjoint subsets $U_1$ and $U_2$ of $\{1, \ldots, m\}$ if no element of $U_1 \cup U_2$ occurs as an intermediate element of $S$ and one of the following two statements is true.

- An element of $U_1$ occurs as the first element of $S$ and an element of $U_2$ occurs as the last element of $S$.

- An element of $U_1$ occurs as the first element of $S$ and an element of $U_2$ occurs as the last element of $S$.

Let $s = \log m/4000 + 1$. We claim that there are two disjoint subsets $U_1$ and $U_2$ of size $\ell = \lfloor m/(2^{s+1}) \rfloor$ that $SEQ$ has at most $s - 1$ links between them. Indeed, assume the opposite. Then, in particular, $SEQ$ has at least $s$ links between every $U_1 \subseteq \{1, \ldots, m/2\}$ and $U_2 \subseteq \{m/2 + 1, \ldots, m\}$ of size $\ell$ each. By Theorem 1.1. of Alon and Maass [1], $|SEQ| \geq 1/8 \times m(s - 9)$. On the other hand, as each element of $\{1, \ldots, m\}$ occurs at most $\log m/40000$ times, $|SEQ| \leq m \times \log m/40000$. That is, $\log m/40000 \geq 1/8 \times (s - 9)$ or, after a transformation, $\log m/5000 + 9 \geq s = \log m/4000 + 1$, which is incorrect for a sufficiently large $m$. This contradiction proves the correctness of the initial claim.

Note that $\ell \geq m/2^{\log m/4000+2} = m^{3999/4000}/4 \geq m^{0.999}$ for a sufficiently large $m$. Therefore, it remains to show that there is a partition $SEQ_1, \ldots, SEQ_{c+1}$ of $SEQ$
into intervals with $c \leq s - 1$ for which one of the statements of Claim 2 is true. Let $I_1, \ldots, I_c$ be a largest set of links between $U_1$ and $U_2$, $c \leq s - 1$ by assumption.

As the first element of the link, completely determines the link itself, the first elements of $I_1, \ldots, I_c$ are all different. We assume that they occur on $SEQ$ in the order that the respective links are listed. Note that for two consecutive links $I_i, I_{i+1}$, the first element of $I_{i+1}$ is either the last element of $I_i$ or occurs on $SEQ$ after the last element of $I_i$. Indeed, since the first element of $I_{i+1}$ occurs after the first element of $I_i$, any ‘deeper’ overlap would imply that the first element of $I_{i+1}$ does not belong to $U_1 \cup U_2$, a contradiction. With this in mind, we can define the following intervals $SEQ_1, \ldots, SEQ_c$. $SEQ_1$ is the prefix of $SEQ$ whose last element is the first element of $SEQ$. For each $1 < i \leq c$, $SEQ_i$ is the interval of $SEQ$ whose first element is the one immediately following the last element of $SEQ_{i-1}$ and whose last element is the first element of $I_i$. Finally, $SEQ_c$ is the suffix of $SEQ$ whose first element is the one immediately following the last element of $SEQ_c$. Note that for each $1 \leq i \leq c$, the last element of $I_i$ belongs to $SEQ_{i+1}$.

Assume w.l.o.g. that the first element of $I_1$ belongs to $U_1$. Then we prove by induction on $1 \leq i \leq c + 1$ that elements of $U_1$ occur in only in intervals $SEQ_i$ with $i$ being odd and the elements of $U_2$ occur only in intervals $SEQ_i$ with $i$ being even.

Consider first $SEQ_1$. By assumption, its last element belongs to $U_1$. Assume that $SEQ_1$ has another element of $U_2$. As all the elements of $U_2$ occur in $SEQ_1$ before the last one, we can identify the last element $u$ in $SEQ_1$ that belongs to $U_2$ and the first element $v$ of $U_1$ following $u$. Then the interval between $u$ and $v$ is a link located before $I_1$, that is $SEQ$ has at least $c + 1$ links between $U_1$ and $U_2$ in contradiction to the maximality of $c$.

Assume now that $i > 1$. Assume first that $i \leq c$. We can assume w.l.o.g. that $i$ is even for the proof for the case where $i$ is odd is symmetric. By the induction assumption, all the elements of $U_1 \cup U_2$ that belong to $SEQ_{i-1}$ in fact belong to $U_1$. In particular, the first element of $I_{i-1}$ belongs to $U_1$. Therefore, the last element $u'$ of $I_{i-1}$ belongs to $U_2$. By construction, any element of $SEQ_i$ preceding $u'$ is an intermediate element of $I_{i-1}$ and hence does not belong to $U_1 \cup U_2$. Therefore, if $SEQ_i$ contains an element of $U_1$ then it occurs after $u'$. Consequently, we can identify on $SEQ$, the first element $v$ of $U_1$ and the last element $u$ of $U_2$ preceding it. The interval between $u$ and $v$ is a link that lies between $I_{i-1}$ and $I_i$, again implying that $SEQ$ has at least $c + 1$ links in contradiction to the maximality of $c$.

Assume now that $i = c + 1$. Again, we can assume w.l.o.g. that $i$ is even. Arguing as in the previous case, we observe that if $SEQ_{c+1}$ contains elements of $U_1$ then $SEQ_{c+1}$ contains a link occurring after $I_c$ in contradiction to the maximality of $c$.

Proof of Lemma 3 Let $c$ be such that $|X| = c - 1$ and let $Q$ be a path containing $X$ and on which $X$ generates a partition $Q_1, \ldots, Q_c$ such that one of the following two statements is true.

1. Elements of $Y_1$ occur only on $Q_i$ with an odd $i$ and elements of $Y_2$ occur on only $Q_i$ with an even $i$.

2. Elements of $Y_1$ occur only on $Q_i$ with an even $i$ and elements of $Y_2$ occur on only $Q_i$ with an odd $i$. 38
We assume w.l.o.g. that the first statement is true. In fact, due to the uniformity of $Z$, a stronger statement is true.

**Claim 3.** For each path $P$ containing $X$, let $P_1, \ldots, P_c$ be the partition of $P$ generated by $X$. Then elements of $Y_1$ occur only on $P_i$ with an odd $i$ and elements of $Y_2$ occur on only $P_i$ with an even $i$.

**Proof.** Due to $Z$ being a DAG, the vertices of $X$ occur in the same order on each path containing $X$. In particular, this means that for each $1 \leq i \leq q$, $P_i$ and $Q_i$ have the same initial vertex and the same final vertex. Assume that the claim does not hold for some $1 \leq i \leq q$. If $i$ is odd this means that an element of $Y_2$ occurs in $P_i$ and does not occur in $Q_i$, in contradiction to Lemma 2. Similarly, if we assume that the claim does not hold for some even $i$ then this means that an element of $Y_1$ occurs on $P_i$ while not occurring on $Q_i$, again in contradiction to Lemma 2. 

Let $Q'_1, \ldots, Q'_c$ and $Q''_1, \ldots, Q''_c$ be respective partitions of $Q'$ and $Q''$ generated by $X$. Let $Q_k^i = Q_k'$ whenever $k$ is odd and $Q_k^i = Q_k''$ whenever $k$ is even. Let $Q^* = Q_1^i + \cdots + Q_c^i$. Clearly, $Q^*$ is a source-sink path of $Z$. We prove that $Q^*$ is a computational path assigning all the vertices of $Y_1 \cup Y_2$ negatively.

Assume the first statement does not hold, that is $Q^*$ is not a computational path. Then there is a variable $x$ that occurs on $Q^*$ both positively and negatively. As all the labels of $Q^*$ are either labels of $Q'$ or labels of $Q''$ which do not have opposite variable occurrences due to being computational paths, variable $x$ occurs positively in one of $A(Q')$, $A(Q'')$ and negatively in the other. As all the variables of $\text{Vars}(Z) \setminus (Y_1 \cup Y_2)$ have the same occurrence in both $A(Q')$ and $A(Q'')$, $x$ must belong to $Y_1 \cup Y_2$.

Assume w.l.o.g. that $x \in Y_1$. The case with $x \in Y_2$ is symmetric. By Claim 3, $x$ occurs only on paths $Q^*_i$ with odd $i$. However, all these paths are subpaths of $Q'$. That is, all the occurrences of $x$ on $Q^*$ are occurrences of $x$ on $Q'$. Since the latter is a computational path and cannot contain opposite occurrences of the same variable, $Q^*$ does not contain opposite occurrences of $x$. Thus we have arrived at a contradiction that there are no variables having opposite occurrences on $Q^*$, thus confirming that $Q^*$ is a computational path.

In order to show that all the variables of $Y_1 \cup Y_2$ are falsified by $Q^*$, consider a variable $x \in Y_1$. Then all the occurrences of $x$ are on $Q_i^*$ with an odd $i$. That is, all the occurrences of $x$ on $Q^*$ are occurrences of $Q'$. As $Q'$ assigns $x$ negatively, so is $Q^*$. For $x \in Y_2$, the reasoning is symmetric with the even fragments of $Q^*$ used instead the odd ones and $Q''$ instead $Q'$.

**D Proof of Proposition 1 (Sketch).**

Let us pick a random assignment to the variables of $H$ using the following procedure.

1. Arbitrarily order the variables of $\phi_H$ so that all the non-leaf variables are ordered before the leaf variables.

2. Pick a literal of each non-leaf variable with probability 0.5 for both positive and negative literals.
3. For each leaf variable $\ell_{i,j}$, do as follows.

- If a positive literal has been chosen for at least one non-leaf variable of $C_{i,j}$ choose either the positive or the negative literal of $\ell_{i,j}$ with probability $0.5$.
- Otherwise, choose the positive literal of $\ell_{i,j}$ with probability $1$.

Let the probability of the resulting assignment be the product of the individual probabilities of the chosen literals.

Let $S \in \text{SAT}$. By construction, the probabilities of literals of non-leaf variables are $0.5$ and the probabilities of literals of leaf variables are $0.5$ if and only if they are not fixed. It follows that the probability assigned to $S$ by the above procedure is $(1/2)^{|S\setminus \text{Fix}(S)|}$ exactly as in the defined probability space.

If $S \notin \text{SAT}$ then there is a clause $C_{i,j}$ falsified by $S$. In particular, this means that $\ell_{i,j}$ occurs negatively in $S$ as well as the rest of variables of $C_{i,j}$. Clearly, according to the above procedure, the probability of a negative literal of $\ell_{i,j}$ is $0$ and hence the probability assigned to $S$ is $0$ as well.

It follows that the sum of probabilities of assignments obtained by the above procedure is in fact the sum of probabilities of satisfying assignments (which are the same as in our probability space). To show that the sum of probabilities is $1$, represent the process of choosing a random assignment as a rooted decision tree with probabilities of literals being the weights of the edges of the tree and the weight of a path being the product of weights of its edges and the weight of a collection of paths being the sum of weights of paths in this collection. Then, starting from the leaves and moving towards the root, argue by induction that the weight of each subtree is $1$. ■