2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)

Tree-depth and the Formula Complexity of Subgraph Isomorphism

Deepanshu Kush
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
Email: deepkush@cs.toronto.edu

Benjamin Rossman
Department of Computer Science
Duke University
Durham, North Carolina, USA
Email: benjamin.rossman@duke.edu

***Abstract*—For a fixed "pattern" graph G, the *colored G-subgraph isomorphism problem* (denoted $\mathrm{SUB}(G)$) asks, given an n-vertex graph H and a coloring $V(H) \to V(G)$, whether H contains a properly colored copy of G. The complexity of this problem is tied to parameterized versions of *P* =? *NP* and *L* =? *NL*, among other questions. An overarching goal is to understand the complexity of $\mathrm{SUB}(G)$, under different computational models, in terms of natural invariants of the pattern graph G.**

In this paper, we establish a close relationship between the *formula complexity* of $\mathrm{SUB}(G)$ and an invariant known as *tree-depth* (denoted $\mathtt{td}(G)$). $\mathrm{SUB}(G)$ is known to be solvable by monotone *AC*^{0} formulas of size $O(n^{\mathtt{td}(G)})$. Our main result is an $n^{\tilde{\Omega}(\mathtt{td}(G)^{1/3})}$ lower bound for formulas that are monotone *or* have sub-logarithmic depth. This complements a lower bound of Li, Razborov and Rossman [8] relating tree-width and *AC*^{0} circuit size. As a corollary, it implies a stronger homomorphism preservation theorem for first-order logic on finite structures [14].

The technical core of this result is an $n^{\Omega(k)}$ lower bound in the special case where G is a complete binary tree of height k, which we establish using the *pathset framework* introduced in [15]. (The lower bound for general patterns follows via a recent excluded-minor characterization of tree-depth [4], [6].) Additional results of this paper extend the pathset framework and improve upon both, the best known upper and lower bounds on the average-case formula size of $\mathrm{SUB}(G)$ when G is a path.

***Keywords*-circuit complexity; subgraph isomorphism; tree-width;**

I. Introduction

Let G be a fixed "pattern" graph. In the COLORED G-SUBGRAPH ISOMORPHISM PROBLEM, denoted $\mathrm{SUB}(G)$, we are given an n-vertex "host" graph H and a vertex-coloring $c : V(H) \to V(G)$ as input and required to determine whether or not H contains a properly colored copy of G (i.e., a subgraph $G' \subseteq H$ such that the restriction of c to $V(G')$ constitutes an isomorphism from G' to G). This general problem includes, as special cases, several important problems in parameterized complexity. In particular, $\mathrm{SUB}(G)$ is equivalent (up to *AC*^{0} reductions) to the k-CLIQUE and DISTANCE-k CONNECTIVITY problems when G is a clique or path of order k.

For any fixed pattern graph G, the problem $\mathrm{SUB}(G)$ is solvable by brute-force search in polynomial time $O(n^{|V(G)|})$. Understanding the fine-grained complexity of $\mathrm{SUB}(G)$ — in this context, we mean the exponent of n in the complexity of $\mathrm{SUB}(G)$ under various computational models — for general patterns G is an important challenge that is tied to major open questions including *P* =? *NP*, *L* =? *NL*, *NC*^{1} =? *L*, and their parameterized versions (*FPT* =? *W*[1], etc.) An overarching goal is to bound the fine-grained complexity of $\mathrm{SUB}(G)$ in terms of natural invariants of the graph G.

Two key invariants arising in this connection are tree-width (tw) and tree-depth (td). The *tree-depth* of G is the minimum height of a rooted forest whose ancestor-descendant closure contains G as a subgraph. This invariant has a number of equivalent characterizations and plays a major role in structural graph theory and parameterized complexity [10]. *Tree-width* is even more widely studied in graph theory and parameterized complexity [5]. It is defined in terms of a different notion of tree decomposition and provides a lower bound on tree-depth ($\mathsf{tw} + 1 \leq \mathsf{td}$).

These two invariants provide well-known upper bounds on the *circuit size* and *formula size* of $\mathrm{SUB}(G)$. To state this precisely, we regard $\mathrm{SUB}(G)$ as a sequence of boolean functions $\{0,1\}^{|E(G)| \cdot n^2} \to \{0,1\}$ where the input encodes a host graph H with vertex set $V(G) \times \{1, \dots, n\}$ under the vertex-coloring that maps (v, i) to v. (Restricting attention to this class of inputs is without loss of generality.) Throughout this paper, we consider circuits and formulas in the unbounded fan-in basis $\{\mathrm{AND}_\infty, \mathrm{OR}_\infty, \mathrm{NOT}\}$; we measure *size* of both circuits and formulas by the number of gates. A circuit or formula is *monotone* if it contains no NOT gates. We use *AC*^{0} as an adjective that means "depth $O(1)$"

This is an extended abstract. The full version of this paper can be found at https://eccc.weizmann.ac.il/report/2020/061/.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00012

in reference to upper bounds and "depth $o(\log n)$" in reference to lower bounds on formula size.[1]

Theorem 1 (Folklore upper bounds)**.** *For all pattern graphs G, $\mathrm{SUB}(G)$ is solvable by monotone AC^0 circuits (respectively, formulas) of size $O(n^{\mathsf{tw}(G)+1})$ (respectively, $O(n^{\mathsf{td}(G)})$).*

It is conjectured that $\mathrm{SUB}(G)$ requires circuit size $n^{\Omega(\mathsf{tw}(G))}$ for all graphs G; if true this would imply $FPT \neq W[1]$ and $P \neq NP$ in a very strong way. As evidence for this conjecture, Marx [9] proved a conditional $n^{\Omega(\mathsf{tw}(G)/\log \mathsf{tw}(G))}$ lower bound assuming the Exponential Time Hypothesis. Providing further evidence, Li, Razborov and Rossman [8] established an unconditional $n^{\Omega(\mathsf{tw}(G)/\log \mathsf{tw}(G))}$ lower bound for AC^0 circuits, via a technique that extends to (unbounded depth) monotone circuits. This result is best stated in terms of a certain graph invariant $\kappa(G)$ introduced in [8]:

Theorem 2 (Lower bound on the restricted circuit size of $\mathrm{SUB}(G)$ [8])**.** *For all pattern graphs G, the circuit size of $\mathrm{SUB}(G)$ — in both the AC^0 and monotone settings — is at least $n^{\kappa(G)-o(1)}$ where $\kappa(G)$ is a graph invariant satisfying $\Omega(\mathsf{tw}(G)/\log \mathsf{tw}(G)) \leq \kappa(G) \leq \mathsf{tw}(G)+1$.*[2]

Shifting our focus from circuits to formulas, it is natural to conjecture that $\mathrm{SUB}(G)$ requires formula size $n^{\Omega(\mathsf{td}(G))}$. This statement generalizes the prominent conjecture that DISTANCE-k CONNECTIVITY requires formula size $n^{\Omega(\log k)}$, which as a consequence implies $NC^1 \neq NL$. (There is also an average-case version of this conjecture which implies $NC^1 \neq L$, as we explain shortly.)

In this paper, we carry out the final step in the proof of an analogous result to Theorem 2 that lower bounds the restricted formula size of $\mathrm{SUB}(G)$ in terms of an invariant $\tau(G)$ that is polynomially related to tree-depth:

Theorem 3 (Lower bound on the restricted formula size of $\mathrm{SUB}(G)$)**.** *For all patterns graphs G, the formula size of $\mathrm{SUB}(G)$ — in both the AC^0 and monotone settings — is at least $n^{\tau(G)-o(1)}$ where $\tau(G)$ is a graph invariant satisfying $\widetilde{\Omega}(\mathsf{td}(G)^{1/3}) \leq \tau(G) \leq \mathsf{td}(G)$.*

The invariant $\tau(G)$ was introduced in [16], where it was also shown that $n^{\tau(G)-o(1)}$ is a lower bound on the formula size of $\mathrm{SUB}(G)$ in the AC^0 and monotone settings. The results of [16] generalized lower bounds for $\mathrm{SUB}(P_k)$ from papers [13], [15], which showed that $\tau(P_k) = \Omega(\log k)$ (where P_k is the path graph of length k). As we will explain shortly, this lower bound for $\tau(P_k)$ implies that $\tau(G) = \Omega(\log \mathsf{td}(G))$ for all graphs G. The contribution of the present paper lies in improving this logarithmic lower bound to a polynomial one by showing $\tau(G) = \widetilde{\Omega}(\mathsf{td}(G)^{1/3})$.

Remark 1. It is helpful to keep in mind the related inequalities:

$$\text{circuit size} \leq \text{formula size}, \quad \mathsf{tw}+1 \leq \mathsf{td}, \quad \kappa \leq \tau.$$

It is further known that $\mathsf{td}(G) \leq (\mathsf{tw}(G)+1)\log|V(G)|$ [10]. A nearly maximal separation between invariants td and tw is witness by *bounded-degree trees* T, which have tree-width 1 but tree-depth $\Omega(\log|V(T)|)$. This class includes paths and complete binary trees, the two families of pattern graphs studied in this paper.

For trees T, we point out that $\mathrm{SUB}(T)$ is computable by monotone AC^0 circuits of size $c(T)\cdot n^2$ for a constant $c(T)$ depending on T. (This follows from Theorem 1, since all trees have tree-width 1.) Although formulas are a weaker model than circuits, establishing formula lower bounds for $\mathrm{SUB}(T)$ of the form $n^{\Omega(\log|V(T)|)}$, as we do in this paper, is a subtle task which requires techniques that distinguish formulas from circuits. Accordingly, Theorem 3 involves greater machinery than Theorem 2. The invariant $\tau(G)$ is also significantly harder to define and analyze compared to $\kappa(G)$.

A. Minor-monotonicity

Recall that a graph F is a *minor* of G if F can be obtained from G by a sequence of edge deletions and contractions (i.e., remove an edge and identify its two endpoint). A graph invariant p is said to be *minor-monotone* if $p(F) \leq p(G)$ whenever F is a minor of G. As observed in [8], the complexity of $\mathrm{SUB}(G)$ (under any reasonable class of circuits) is minor-monotone in the following sense:

Lemma 4. *If F is a minor of G, then there is a reduction from $\mathrm{SUB}(F)$ to $\mathrm{SUB}(G)$ via a monotone projection.*[3]

In the quest to characterize the complexity of $\mathrm{SUB}(G)$ in terms of invariants of G, it makes sense to

[1] Here and elsewhere, asymptotic notation hides constants that may depend on G. In other contexts, e.g. $\Omega(\mathsf{td}(G))$, hidden constants are absolute.

[2] It is actually shown that $\kappa(G)$ is at most the *branch-width* of G, an invariant related to tree-width by $\frac{2}{3}(\mathsf{tw}+1) \leq \mathsf{bw} \leq \mathsf{tw}+1$. The relationship between $\kappa(G)$ and $\mathsf{tw}(G)$ was further investigated by Rosenthal [11], who identified the separating example $\kappa(Q) = \Theta(\mathsf{tw}(Q)/\sqrt{\log \mathsf{tw}(Q)})$ for hypercubes Q.

[3] That is, for every n, there is a reduction from $\mathrm{SUB}(F)$ to $\mathrm{SUB}(G)$, viewed as boolean functions $\{0,1\}^{|E(F)|\cdot n} \to \{0,1\}$ and $\{0,1\}^{|E(G)|\cdot n} \to \{0,1\}$, via a monotone projection that maps each variable of $\mathrm{SUB}(G)$ to a variable of $\mathrm{SUB}(F)$ or a constant 0 or 1.

focus on minor-monotone ones. Indeed, invariants tw, td, κ, τ are all minor-monotone. This feature is useful in bounding the complexity of $\mathrm{SUB}(G)$. For example, we can combine the result of [2] that every graph with tree-width at least $k^9 \operatorname{polylog} k$ contains a $(k \times k)$-grid minor, with the lower bound $\kappa((k \times k)\text{-grid graph}) = \Omega(k)$ from [1], in order to conclude that $\kappa(G) = \widetilde{\Omega}(\mathsf{tw}(G)^{1/9})$ for all graphs G. (Notation $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ suppresses poly-logarithmic factors.) The stronger $\kappa(G) = \Omega(\mathsf{tw}(G)/\log \mathsf{tw}(G))$ bound of Theorem 2 is obtained by a more nuanced analysis of the invariant κ.

In a similar manner, we can combine the fact that every graph G contains a path of length $\mathsf{td}(G)$ [10], with the lower bound $\tau(P_k) = \Omega(\log k)$ [15], in order to conclude that $\tau(G) = \Omega(\log \mathsf{td}(G))$ for all graphs G. With the goal of improving this lower bound to $\Omega(\operatorname{poly} \mathsf{td}(G))$ (that is, $\Omega(\mathsf{td}(G)^\varepsilon)$ for some constant $\varepsilon > 0$), Kawarabayashi and Rossman [6] established a polynomial excluded-minor characterization of tree-depth, which was subsequently sharpened by Czerwiński, Nadara and Pilipczuk [4].

Theorem 5 (Excluded-minor characterization of tree-depth [6], [4])**.** *Every graph G with tree-depth $\Omega(k^3)$ satisfies at least one of the following:*

(i) *G has tree-width $\geq k$,*
(ii) *G contains a path of length 2^k,*
(iii) *G contains a T_k-minor, where T_k is the complete binary tree of height k.*

Theorem 5 reduces the task of proving $\tau(G) = \Omega(\operatorname{poly} \mathsf{td}(G))$ to the task of proving $\tau(T_k) = \Omega(\operatorname{poly} k)$. It is this final step that we tackle in this paper.[4]

Theorem 6 (Main result of this paper)**.** $\tau(T_k) = \Omega(k)$.

This lower bound is proved in Section III using a certain potential function (described in Section II), which further reduces our task to a combinatorial problem concerning *join-trees* over T_k, that is, rooted binary trees whose leaves are labeled by edges of T_k. This is the same combinatorial framework as the $\tau(P_k) = \Omega(\log k)$ lower bound of [15]; however, the task of analyzing join-trees over T_k turned out to be significantly harder compared with P_k.

Theorems 5 and 6 combine to prove Theorem 3 (the bound $\tau(G) = \widetilde{\Omega}(\mathsf{td}(G)^{1/3})$) as follows. For a graph G with tree-depth $\Omega(k^3)$, we can see that $\tau(G) = \Omega(k/\log k)$ by considering the three cases given by Theorem 5:

(i) If G has tree-width $\geq k$, then $\tau(G) \geq \kappa(G) = \Omega(k/\log k)$ by Theorem 2.
(ii) If G contains a path of length 2^k, then $\tau(G) \geq \tau(P_{2^k}) = \Omega(k)$ by the lower bound of [15].
(iii) If G contains a T_k-minor, then $\tau(G) \geq \tau(T_k) = \Omega(k)$ by Theorem 6.

B. Corollary in finite model theory

Theorem 3 has a striking consequence in finite model theory, observed in the paper [14].

Corollary 1 (Polynomial-rank homomorphism preservation theorem over finite structures)**.** *Every first-order sentence of quantifier-rank r that is preserved under homomorphisms of finite structures is logically equivalent on finite structures to an existential-positive first-order sentence of quantifier-rank $\widetilde{O}(r^3)$.*

The polynomial upper bound of Corollary 1 improves an earlier *non-elementary* upper bound of [12]. This surprising connection between circuit complexity and finite model theory was in fact the original motivation behind Theorems 3 and 5, as well as the present paper.

C. Improved bounds for average-case $\mathrm{SUB}(P_k)$

Additional results of this paper improve both the average-case upper and lower bounds for $\mathrm{SUB}(P_k)$ [13]. Here *average-case* refers to the p-biased product distribution on $\{0,1\}^{kn^2}$ where $p = n^{-(k+1)/k}$. This input distribution corresponds to a random graph $\boldsymbol{X}$, comprised of $k+1$ layers of n vertices, where every pair of vertices in adjacent layers is connected by an edge independently with probability p. For this choice of p, the probability that $\boldsymbol{X}$ contains a path of length k containing one vertex from each layer is bounded away from 0 and 1.

Theorem 7 ([15])**.** *$\mathrm{SUB}(P_k)$ is solvable on $\boldsymbol{X}$ with probability $1 - o(1)$ by monotone AC^0 formulas of size $n^{\frac{1}{2}\lceil \log_2(k) \rceil + o(1)}$. On the other hand, AC^0 formulas solving $\mathrm{SUB}(P_k)$ on $\boldsymbol{X}$ with probability ≥ 0.9 require size $n^{\tau(P_k)-o(1)}$ where $\tau(P_k) \geq \frac{1}{2}\log_{\sqrt{13}+1}(k)$ $(\geq 0.22\log_2(k))$.*

A similar average-case lower bound for (unbounded depth) monotone formulas was subsequently shown in [13]. Precisely speaking, that paper gives an $n^{\frac{1}{2}\tau(P_k)-o(1)}$ lower bound under $\boldsymbol{X}$, as well as an $n^{\tau(P_k)-o(1)}$ lower bound under the distribution that, half

[4] Theorem 6 delivers on a promise in papers [6], [14], [16], which cite $\tau(T_k) = \Omega(\operatorname{poly} k)$ as an unpublished result of upcoming work. Let us mention that, after finding many devils in the details of an earlier sketch of an $\Omega(\sqrt{k})$ bound by the second author, we worked out an entirely different approach in this paper, which moreover gives a *linear* lower bound (which is tight up to a constant since $\tau(T_k) \leq \mathsf{td}(T_k) = k$).

of the time, is $\boldsymbol{X}$ and, the other half, is a uniform random path of length k with no additional edges.

1) Upper bound: The *average-case* upper bound of Theorem 7 can be recast, in stronger terms, as a *worst-case randomized* upper bound for the problem of multiplying k $(n \times n)$-permutation matrices $Q_1, \dots, Q_k$. This problem is solvable by deterministic (non-randomized) *AC*^{0} formulas of size $n^{\log_2(k)+O(1)}$ via the classic "recursive doubling" procedure: recursively compute matrix products $L := Q_1 \cdots Q_{\lceil k/2 \rceil}$ and $R := Q_{\lceil k/2 \rceil + 1} \cdots Q_k$ and then obtain $Q_1 \cdots Q_k = LR$ by a single matrix multiplication.

Randomization lets us achieve quadratically smaller formula size $n^{\frac{1}{2}\log_2(k)+O(1)}$. The idea is as follows. Generate $m := \widetilde{O}(\sqrt{n})$ independent random sets $\boldsymbol{I}_1, \dots, \boldsymbol{I}_m \subseteq [n]$, each of size $\sqrt{n}$. Rather than compute all entries of the permutation matrix L using n^2 subformulas, we will encode the information in L more efficiently using $(2\log n + 1)m^2 = \widetilde{O}(n)$ subformulas (note that $\log(n!) = O(n \log n)$ bits are required to encode a permutation matrix). For each $(r,s) \in [m]^2$, we recursively construct

- one subformula that indicates[5] whether or not there exists a *unique* pair $(a,b) \in \boldsymbol{I}_r \times \boldsymbol{I}_s$ such that $L_{a,b} = 1$, and
- $2\log n$ additional subformulas that give the binary representation of a and b whenever such (a,b) uniquely exists.

Similarly, with respect to the permutation matrix R, for each $(s,t) \in [m]^2$, we have $2\log n + 1$ recursively constructed subformulas that indicate whether there exists a *unique* pair $(b,c) \in \boldsymbol{I}_s \times \boldsymbol{I}_t$ such that $R_{b,c} = 1$, and if so, give the binary representation of b and c. Using these subformulas for subproblems L and R, we construct the corresponding formulas for $Q_1 \cdots Q_k$ which, for each $(r,t) \in [m]^2$, indicate whether there exists a *unique* pair $(a,c) \in \boldsymbol{I}_r \times \boldsymbol{I}_t$ such that $R_{a,c} = 1$, and if so, give the binary representation of a and c. These formulas check, for each $s \in [m]$, whether the (r,s)- and (s,t)-subformulas of the L- and R-subproblems output (a,b) and (b',c), respectively such that $b = b'$. These formulas are therefore larger than the subformulas for subproblems L and R by a factor $\widetilde{O}(m)$. This implies an upper bound $\widetilde{O}(m)^{\lceil \log_2(k) \rceil} = n^{\frac{1}{2}\log_2(k)+O(1)}$ on size and $O(\log k)$ on depth of the resulting randomized *AC*^{0} formulas.

A similar construction solves $\mathrm{SUB}(P_k)$ in the average-case. This yields an upper bound on $\frac{1}{2}\log k + O(1)$ on the parameter $\tau(P_k)$, which we initially guessed might be optimal. However, in the course of trying prove a matching lower bound, we were surprised to discover a better upper bound!

Theorem 8. *There exist randomized AC*^{0} *formulas of size* $n^{\frac{1}{3}\log_\varphi(k)+O(1)}$ $(\leq n^{0.49\log_2(k)+O(1)})$*, where* $\varphi = (\sqrt{5}+1)/2$ *is the golden ratio, which compute the product of* k *permutation matrices.*

The algorithm generalizes the randomized "recursively doubling" method outlined above. Here we give a brief sketch (full details are given in the full version). Let $k = \mathrm{Fib}(\ell)$ where $\ell \geq 3$ (i.e., the ℓ^{th} Fibonacci number, which satisfies $\mathrm{Fib}(\ell) = \mathrm{Fib}(\ell-1) + \mathrm{Fib}(\ell-2)$). We will represent information about the product $Q_1 \cdots Q_k$ by constructing formulas that enumerate all triples $(a,c,d) \in [n]^3$ such that

(1)
$$(Q_1 \cdots Q_{\mathrm{Fib}(\ell-1)})_{a,c} = (Q_{\mathrm{Fib}(\ell-1)+1} \cdots Q_k)_{c,d} = 1.$$

This is accomplished by generating $m := \widetilde{O}(n^{1/3})$ independent random sets $\boldsymbol{I}_1, \dots, \boldsymbol{I}_m$, each of size $n^{2/3}$, and recording the *unique* triples $(a,c,d) \in \boldsymbol{I}_r \times \boldsymbol{I}_t \times \boldsymbol{I}_u$ for which (1) holds.

The recursive construction breaks into a "left" subproblem on $(Q_1, \dots, Q_{\mathrm{Fib}(\ell-1)})$ and a "right" subproblem on $(Q_{\mathrm{Fib}(\ell-2)+1}, \dots, Q_k)$.[6] (In contrast to the "recursive doubling" method, here the "left" and "right" subproblems involve overlapping subsequences of permutation matrices.) In the "left" subproblem: for each $(r,s,t) \in [m]^3$, we have

- $3\log n + 1$ subformulas that indicate whether there exists a *unique* triple $(a,b,c) \in \boldsymbol{I}_r \times \boldsymbol{I}_s \times \boldsymbol{I}_t$ such that $(Q_1 \cdots Q_{\mathrm{Fib}(\ell-2)})_{a,b} = (Q_{\mathrm{Fib}(\ell-2)+1} \cdots Q_{\mathrm{Fib}(\ell-1)})_{b,c} = 1$, and if so, give the binary representation of a, b, c.

In the "right" subproblem: for each $(r,s,t) \in [m]^3$, we have

- $3\log n + 1$ subformulas that indicate whether there exists a *unique* triple $(b,c,d) \in \boldsymbol{I}_s \times \boldsymbol{I}_t \times \boldsymbol{I}_u$ such that $(Q_{\mathrm{Fib}(\ell-2)+1} \cdots Q_{\mathrm{Fib}(\ell-1)})_{b,c} = (Q_{\mathrm{Fib}(\ell-1)+1} \cdots Q_k)_{c,d} = 1$, and if so, give the binary representation of b, c, d.

The subformulas in the "left" and "right" subproblems may be combined to produce the analogous (left-handed) formulas for the original input $(Q_1, \dots, Q_k)$: for each $(r,t,u) \in [m]^3$, we construct

[5] When describing the behavior of randomized formulas in this subsection (using verbs like "indicate", "output", etc.), we leave implicit that the description holds *correctly with high probability* for any input.

[6] The "right" subproblem on $(Q_{\mathrm{Fib}(\ell-2)+1}, \dots, Q_k)$ can also be viewed as a "left" subproblem on $(P_1, \dots, P_{\mathrm{Fib}(\ell-1)})$ where P_i is the transpose of Q_{k-i+1}.

- $3\log n + 1$ subformulas that indicate whether there exists a *unique* triple $(a, c, d) \in \boldsymbol{I}_r \times \boldsymbol{I}_t \times \boldsymbol{I}_u$ such that (1) holds, and if so, give the binary representation of a, c, d.

These formulas check, for each $s \in [m]$, whether the (r, s, t)- and (s, t, u)-subformulas in the "left" and "right" subproblems output triples (a, b, c) and (b', c', d), respectively, such that $b = b'$ and $c = c'$. These formulas are therefore larger than the subformulas in the "left" and "right" subproblems by a factor $\widetilde{O}(m)$. Taking $k = \mathrm{Fib}(3) = 2$ as our base case with formula size $n^{O(1)}$, this gives an upper bound $\widetilde{O}(m)^{\ell-3} \cdot n^{O(1)} = n^{\frac{1}{3}\log_\varphi(k)+O(1)}$ for all $k = \mathrm{Fib}(\ell)$ (which extends as well to non-Fibonacci numbers k).

We introduce a broad class of randomized algorithms (based on a simplification of the pathset complexity measure; see full version for details) that generalize both the "recursive doubling" and "Fibonacci overlapping" algorithms outlined above. We also discuss reasons, including experimental data, which suggest that $n^{\frac{1}{3}\log_\varphi(k)+O(1)}$ might in fact be the asymptotically *tight* bound on the randomized formula size of multiplying k permutations.

2) Lower bound: The final result of this paper improves the $\tau(P_k) \geq \frac{1}{2}\log_{\sqrt{13}+1}(k)$ $(\geq 0.22\log_2(k))$ lower bound of Theorem 7.

Theorem 9. $\tau(P_k) \geq \log_{\sqrt{5}+5}(k) - 1$ $(\geq 0.35\log_2(k) - 1)$

More significant than the quantitative improvement we obtain in Theorem 9 is the fact that our proof further develops *pathset framework* by introducing a new potential function that gives stronger lower bounds on $\tau(G)$. This development and the proof of Theorem 9 are presented in detail in latter sections of the full version.

Since $\frac{1}{3}\log_\varphi(k) = \log_{\sqrt{5}+2}(k)$, our upper and lower bounds are off by exactly 3 in the base of the logarithm. It would be very interesting to completely close this gap.

D. Related work

There have been several papers, including [3], [7], [9], which give conditional lower bounds (under ETH and other assumptions) on the *circuit size* of $\mathrm{SUB}(G)$ and its uncolored variant. We are not aware of any conditional hardness results for the *formula size* of $\mathrm{SUB}(G)$. It would be interesting to show that $\mathrm{SUB}(G)$ requires (unrestricted) formula size $n^{\Omega(\mathsf{td}(G))}$ under a natural assumption.

II. Preliminaries

For a natural number n, $[n]$ denotes the set $\{1, \dots, n\}$. For simplicity of presentation, we occasionally omit floors and ceilings, e.g., treating quantities like $\sqrt{n}$ as natural numbers). This is always without loss of parameters in our results. When no base is indicated, $\log(\cdot)$ denotes the base-2 logarithm.

A. Graphs

In this paper, *graphs* are simple graphs, i.e., pairs $G = (V(G), E(G))$ where $V(G)$ is a set and $E(G)$ is a subset of $\binom{V(G)}{2}$ (the set of unordered pairs $\{v, w\}$ where v, w are distinct elements of $V(G)$). Unless explicitly stated otherwise, graphs are assumed to be locally finite (i.e., every vertex has finite degree) and without isolated vertices (i.e., $V(G) = \bigcup_{e \in E(G)} e$). For a vertex $v \in V(G)$, $\deg_G(v)$ or simply $\deg(v)$ denotes the degree of v in G.

We regard G as a fixed (possibly infinite) "pattern" graph. F shall consistently denote a finite subgraph of G. We write $\subseteq$ for the subgraph relation and $\subset$ (or sometimes $\subsetneq$) for the proper subgraph relation. If F is a subgraph of G, then $G \setminus F$ denotes the graph with edge set $E(G) \setminus E(F)$ (and no isolated vertices).

Two important graphs in this paper are paths and complete binary trees. P_k denotes the path graph of length k (with $k+1$ vertices and k edges). T_k denotes the complete binary tree of height k (with $2^{k+1} - 1$ vertices and $2^{k+1} - 2$ edges). We also consider infinite versions of these graphs. P_∞ is the path graph with vertex set $\mathbb{Z}$ and edge set $\{(i, i+1) : i \in \mathbb{Z}\}$. T_∞ is the union $\bigcup_{k=1}^{\infty} T_k$ under the nesting $T_1 \subset T_2 \subset T_3 \subset \cdots$ where $\mathrm{Leaves}(T_1) \subset \mathrm{Leaves}(T_2) \subset \mathrm{Leaves}(T_3) \subset \cdots$. Thus, T_∞ is an infinite, rootless, layered binary tree, with leaves in layer 0, their parents in level 1, etc.

We use terms *graph invariant* and *graph parameter* interchangeably in reference to real-valued functions on graphs that are invariant under isomorphism.

B. Threshold weightings

We describe a family of edge-weightings on graphs G, which in the case of finite graphs correspond to product distributions that are balanced with respect to the problem $\mathrm{SUB}(G)$. (Definitions in this section are adapted from [8].)

Definition 1. For any graph G and function $\theta : E(G) \to \mathbb{R}$, we denote by Δ_θ : {finite subgraphs of G} $\to \mathbb{R}$ the function

$$\Delta_\theta(F) := |V(F)| - \sum_{e \in E(F)} \theta(e).$$

Definition 2. A *threshold weighting* for a graph G is a function $\theta : E(G) \to [0,2]$ such that $\Delta_\theta(F) \geq 0$ for all finite subgraphs $F \subseteq G$; if G is finite, we additionally require that $\Delta_\theta(G) = 0$.

We refer to the pair (G, θ) as a *threshold-weighted graph*. When θ is fixed, we will at times simply write $\Delta(\cdot)$ instead of $\Delta_\theta(\cdot)$.

Definition 3. A *Markov chain* on a graph G is a matrix $[0,1]^{V(G)\times V(G)}$ that satisfies

- $\sum_{w\in V(G)} M_{v,w} = 1$ for all $v \in V(G)$ and
- $M_{v,w} > 0 \implies \{v,w\} \in E(G)$ for all $v, w \in V(G)$.

Lemma 10. *Every Markov chain M on G induces a threshold weighting θ on G defined by*

$$\theta(\{v,w\}) := M_{v,w} + M_{w,v}.$$

This threshold weighting satisfies

$$\Delta_\theta(F) = \sum_{v\in V(F)} \sum_{w\in V(G)\,:\,\{v,w\}\notin E(F)} M_{v,w}.$$

We remark that this lemma has a converse (shown in [11]): *Every threshold weighting on G is induced by a (not necessarily unique) Markov chain on G.* Lemma 10 also gives us a way to define *threshold weightings* when G is an infinite graph; this will be useful later on.

Example 1. Let M be the transition matrix of the uniform random walk on T_k where $k \geq 2$. That is,

$$M_{v,w} := \begin{cases} 1/\deg(v) & \text{if } \{v,w\} \in E(T_k), \\ 0 & \text{otherwise.} \end{cases}$$

For the associated threshold weighting $\theta : E(T_k) \to [0,2]$, we have

$$\theta(e) = \begin{cases} 4/3 & \text{if } e \text{ contains a leaf,} \\ 5/6 & \text{if } e \text{ contains the root,} \\ 2/3 & \text{otherwise.} \end{cases}$$

A key property of this θ that we will use later on (Lemma 17) is that

$$\Delta_\theta(F) \geq \frac{|V(F) \cap V(T_k \setminus F)|}{3}$$

(that is, $\Delta_\theta(F)$ is at least one-third the size of the boundary of F) for all graphs $F \subseteq T_k$. This is a straightforward consequence of Lemma 10, which is also true in the infinite tree T_∞.

Example 2. Let P_k be the path of length k (with $k+1$ vertices and k edges). The constant function $\theta \equiv 1 + \frac{1}{k}$ is a threshold weighting for P_k. (This is different from the threshold function induced by the uniform random walk on P_k, which has value $1/2$ on the two outer edges of P_k and value 1 on the inner edges.)

This example again makes sense for $k = \infty$. The constant function $E(P_\infty) \mapsto \{1\}$ is a threshold weighting for P_∞. This threshold function has the nice property that

$$\begin{aligned} \Delta(F) &= |V(F)| - |E(F)| \\ &= \#\{\text{connected components of } F\} \end{aligned}$$

for all finite subgraphs $F \subset P_\infty$.

Definition 4. Let G be a finite graph, let θ be a threshold weigting on G, and let n be a positive integer. We denote by $\boldsymbol{X}_{\theta,n}$ be the random $V(G)$-colored graph (i.e., input distribution to $\mathrm{SUB}(G)$) with vertex set $V(G) \times [n]$, vertex-coloring $(v,i) \mapsto v$, and random edge relation given by

$$\begin{aligned} &\mathbb{P}[\,\{(v,i),(w,j)\} \text{ is an edge of } \boldsymbol{X}_{\theta,n}\,] \\ &= \begin{cases} 1/n^{\theta(\{v,w\})} & \text{if } \{v,w\} \in E(G), \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

independently for all $\{(v,i),(w,j)\} \in \binom{V(G)\times[n]}{2}$.

Lemma 11 ([8]). *The probability that $\boldsymbol{X}_{\theta,n}$ is a YES-instance of* $\mathrm{SUB}(G)$ *is bounded away from* 0 *and* 1.

The lower bounds of Theorem 2 and 3 are in fact average-case lower bounds for $\mathrm{SUB}(G)$ under $\boldsymbol{X}_{\theta,n}$ for arbitrary threshold weightings θ. Parameters $\kappa(G)$ and $\tau(G)$ are obtained by taking the optimal choice of threshold weighting θ, as we describe in the next subsection.

C. Join-trees and parameters $\kappa(G)$ and $\tau(G)$

Parameters $\kappa(G)$ and $\tau(G)$ are defined in terms of a notion called *join-trees* for subgraphs of G. A join-tree is simply a "formula" computing a subgraph of G, starting from individual edges, where union ($\cup$) is the only operation.

Definition 5. A *join-tree over G* is a finite rooted binary tree A together with a labeling $\mathrm{label}_A : \mathrm{Leaves}(A) \to E(G) \cup \{\bot\}$ (which may also be viewed as a partial function $\mathrm{Leaves} \rightharpoonup E(G)$). We reserve symbols A, B, C, D, E for join-trees. (F will always denote a subgraph of G.)

The *graph* of A, denoted G_A, is the subgraph of G with edge set $E(G) \cap \mathrm{Range}(\mathrm{label}_A)$. (Note that G_A is always finite.) As a matter of notation, we write $E(A)$ for $E(G_A)$ and $V(A)$ for $V(G_A)$. We also write $\Delta_\theta(A)$ for $\Delta_\theta(G_A)$ where θ is a threshold weighting on G.

We write $\langle\rangle$ for the single-node join-tree labeled by $\bot$. For $e \in E(G)$, we write $\langle e\rangle$ for the single-node join-tree labeled by e. For join-trees B and C, we write $\langle B, C\rangle$ for the join-tree consisting a root with B and C as children (with the inherited labels, i.e., $\text{label}_{\langle B,C\rangle} = \text{label}_B \cup \text{label}_C$). Note that $G_{\langle B,C\rangle} = G_B \cup G_C$.

Every join-tree A is clearly either $\langle\rangle$, or $\langle e\rangle$ where $e \in E(G)$, or $\langle B, C\rangle$ where B, C are join-trees. In the first two cases, we say that A is *atomic*; in the third case, we say that A is *non-atomic*.

We say that B is a *child* of A if $A \in \{\langle B, C\rangle, \langle C, B\rangle\}$ for some C. We say that D is a *sub-join-tree* of A (denoted $D \preceq A$) if $D = \langle\rangle$ or $D = A$ or D is a sub-join-tree of a child of A. We say that D is a *proper sub-join-tree* (denoted $D \prec A$) if $D \preceq A$ and $D \neq A$.

We are now able to define the invariant $\kappa(G)$ in Theorem 2, which lower bounds the restricted circuit size of $\text{SUB}(G)$. (In fact, $\kappa(G)$ also provides a nearly tight upper bound on the average-case AC^0 circuit size of $\text{SUB}(G)$ [11].)

Definition 6 (The invariant $\kappa(G)$ of Theorem 2)**.** For finite graphs G, let

$$\kappa(G) := \max_{\text{threshold wts. } \theta \text{ for } G} \ \min_{\text{join-trees } A:\ G_A = G} \ \max_{B \preceq A} \Delta_\theta(B).$$

The invariant of $\tau(G)$ of Theorem 3 is significantly more complicated to define. We defer its precise definition to the full version. Here, we focus on a simpler "potential function" on join-trees, denoted $\Phi_\theta(A)$, which we use to lower bound $\tau(G)$. In order to state the definition of $\Phi_\theta(A)$, we require the following operation $\ominus$ ("restriction away from") on graphs and join-trees.

Definition 7 (The operation $\ominus$ on graphs and join-trees)**.** For $F \subseteq G$ and a subset $S \subseteq V(G)$, we denote by $F \ominus S$ the graph consisting of the connected components of F that are vertex-disjoint from S.

For a join-tree A, we denote $A \ominus S$ the join-tree with the same rooted tree structure as A and leaf labeling function

$$\text{label}_{A\ominus S}(l) := \begin{cases} \text{label}_A(l) & \text{if } \text{label}_A(l) \in E(G_A \ominus S), \\ \bot & \text{otherwise.} \end{cases}$$

That is, $A \ominus S$ deletes all labels except for edges in $G_A \ominus S$. Note that $G_{A\ominus S} = G_A \ominus S$.

As a matter of notation, if B is another join-tree, we write $A \ominus B$ for $A \ominus V(B)$ and $A \ominus (S \cup B)$ for $A \ominus (S \cup V(B))$.

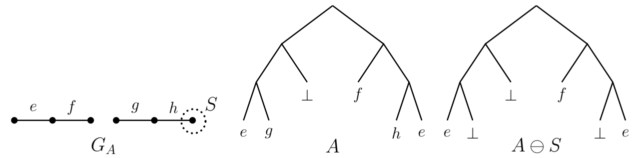

Figure 1: An example where A is a join-tree whose graph G_A consists of two paths of length 2 with edges e, f, g, h. S is the set containing just the external endpoint of h. The join-tree $A \ominus S$ is depicted to the right.

Definition 8 (The potential function Φ_θ on join-trees)**.** Fix a threshold weighting θ on a graph G. The potential function $\Phi_\theta : \{\text{join-trees over } G\} \to \mathbb{R}_{\geq 0}$ is the unique pointwise minimum function satisfying the following two inequalities for all join-trees A, B, C, D:

(†) $\Phi_\theta(A) \geq \Phi_\theta(D) + \Delta_\theta(C \ominus D) + \Delta_\theta(A \ominus (C \cup D))$

if $A \in \{\langle B, C\rangle, \langle C, B\rangle\}$ and $D \preceq B$; and

(‡) $$\Phi_\theta(A) \geq \frac{1}{2}\Big(\Phi_\theta(D) + \Phi_\theta(E \ominus D) + \Delta_\theta(A) + \Delta_\theta(A \ominus (D \cup E))\Big)$$

Alternatively, $\Phi_\theta(A)$ has the following recursive characterization: If A is an atomic join-tree, then

$$\Phi_\theta(A) := \Delta_\theta(A) = \begin{cases} 0 & \text{if } A = \langle\rangle, \\ 2 - \theta(e) & \text{if } A = \langle e\rangle \text{ where } e \in E(G). \end{cases}$$

(Obs: In the case $A = \langle e\rangle$, the constraint $\Phi_\theta(A) \geq \Delta_\theta(A)$ is forced by (‡) where $D = E = \langle\rangle$.)

If $A = \langle B, C\rangle$, then $\Phi_\theta(A)$ is defined to be the maximum over the following three quantities:

$$\max_{D \preceq B} \ \Phi_\theta(D) + \Delta_\theta(C \ominus D) + \Delta_\theta(A \ominus (C \cup D)),$$
$$\max_{D \preceq C} \ \Phi_\theta(D) + \Delta_\theta(B \ominus D) + \Delta_\theta(A \ominus (B \cup D)),$$
$$\max_{D,E \prec A} \ \frac{1}{2}\Big(\Phi_\theta(D) + \Phi_\theta(E \ominus D) + \Delta_\theta(A) + \Delta_\theta(A \ominus (D \cup E))\Big).$$

That is, at least one of inequalities (†) or (‡) is tight for each join-tree A.

This definition, although opaque at first, is clarified in the full version. The key property of $\Phi_\theta(A)$ is that it provides a lower bound the invariant $\tau(G)$, which in turn provides a lower bound on the restricted formula complexity of $\text{SUB}(G)$.

Theorem 12 ([16])**.** *The invariant $\tau(G)$ of Theorem 3 satisfies*

$$\tau(G) \geq \max_{\text{threshold wts. } \theta \text{ for } G} \ \min_{\text{join-trees } A:\ G_A = G} \Phi_\theta(A).$$

The definition of $\tau(G)$ and proof of Theorem 12 are deferred for the full version. First, in Section III, we will present our combinatorial main lemma, which gives a lower bound on $\Phi_\theta(A)$ for all join-trees with graph T_k under the threshold weighting θ of Example 1.

D. Observations about Φ_θ

Note that inequality (†) implies $\Phi_\theta(A) \geq \Phi_\theta(D)$ for all $D \preceq A$ (since $\Delta_\theta(\cdot)$ is nonnegative). Also note that inequality (‡) implies $\Phi_\theta(A) \geq \Delta_\theta(A)$ in the special case $B = C = \langle\rangle$ (since $\Phi_\theta(\langle\rangle) = 0$ and $A \ominus$ (the empty graph) $= A$). Combining these observations, we see that $\Phi_\theta(A) \geq \Delta_\theta(D)$ for all $D \preceq A$. It follows that $\tau(G) \geq \kappa(G)$ for all graphs G, which makes sense in light of the fact that $\kappa(G)$ bounds circuit size and $\tau(G)$ bounds formula size.

Next, observe that $\Phi_\theta(A)$ always equals either $\Phi_\theta(D)$ + (some $\Delta_\theta(\cdot)$-terms) or $\frac{1}{2}(\Phi_\theta(D) + \Phi_\theta(E \ominus D))$ + (*some* $\Delta_\theta(\cdot)$*-terms*) where D and E are proper sub-join-trees of A. This can be expanded out until we get a nonnegative linear combination of $\Delta_\theta(\cdot)$-terms. Looking closely, we see that

$$\Phi_\theta(A) = \sum_{F \subseteq G} c_F \cdot \Delta_\theta(F)$$

where coefficients c_F (which depend on both θ and A) are nonnegative dyadic rational numbers coming from the tight instances of inequalities (†) and (‡). We may further observe, for any $v \in V(G)$, that

$$\sum_{F \subseteq G \,:\, v \in V(F)} c_F \leq 1.$$

This is easily shown by induction using the fact that graphs F_1 and $F_2 \ominus F_1$ and $F_3 \ominus (F_1 \cup F_2)$ are pairwise disjoint for any $F_1, F_2, F_3 \subseteq G$.

A consequence of this observation is the following lemma, which is used in proofs of Theorems 6 and 9.

Lemma 13. *Suppose (G, θ) and (G^*, θ^*) are threshold-weighted graphs such that $G \subseteq G^*$ and $\theta^*(e) \leq \theta(e)$ for all $e \in E(G)$. Then for any join-tree A with graph G, we have*

$$\Phi_\theta(A) \geq \Phi_{\theta^*}(A) - \sum_{e \in E(G)} \Big(\theta(e) - \theta^*(e)\Big).$$

E. Lower bounds on Φ_θ

Having introduced the potential function Φ_θ and described its connection to τ in Theorem 12, we conclude this section by briefly explaining how it is used derive lower bounds $\tau(P_k)$ and $\tau(T_k)$. The main combinatorial lemma behind the lower bound of Theorem 7 is the following:

Lemma 14 ([15]). *Let θ be the constant $1+\frac{1}{k}$ threshold weighting on P_k. For every join-tree A with graph P_k, we have $\Phi_\theta(A) \geq \frac{1}{2}\log_{\sqrt{13}+1}(k)$. (Therefore, $\tau(G) \geq \frac{1}{2}\log_{\sqrt{13}+1}(k)$.)*

The proof is included in Appendix of the full version, for the sake of comparison with our two lower bounds below. We remark that this proof makes crucial use of both (†) and (‡); it was shown in [15] that no lower bound better than $\Phi_\theta(A) = \Omega(1)$ is provable using (†) alone or (‡) alone.

Our lower bound $\tau(T_k) = \Omega(k)$ (Theorem 6) is an immediate consequence of the following:

Lemma 15. *Let θ be the threshold weighting arising from the uniform random walk on T_k (Example 1). For every join-tree A with graph T_k, we have $\Phi_\theta(A) \geq k/30 - 1/5$.*

Our proof, given in the next section, is purely graph-theoretic. Interestingly, the argument essentially uses only inequality (‡); we do not require (†), other than in the weak form $\Phi_\theta(A) \geq \Phi_\theta(D)$ for all $D \prec A$.

It is worth mentioning that the choice of threshold weighting is important in Lemma 15. A different, perhaps more obvious, threshold weighting is the constant function with value $\frac{|V(T_k)|}{|E(T_k)|}$ $(= \frac{2^{k+1}-1}{2^{k+1}-2})$. With respect to this threshold weighting, no lower bound better than $\Omega(1)$ is possible.

Finally, our improved lower bound $\tau(P_k) \geq \log_{\sqrt{5}+5}(k)$ (Theorem 9) is obtained via the following lemma. This result involves a 2-parameter extension of $\Phi_\theta(A)$ denoted $\Phi_\theta(A|S)$ (where $S \subseteq V(G)$), which we introduce in the full version.

Lemma 16. *Let θ be the constant $1 + \frac{1}{k}$ threshold weighting on P_k. For every join-tree A with graph P_k, we have $\Phi_\theta(A|\emptyset) \geq \log_{\sqrt{5}+5}(k) - 1$.*

III. Lower bound $\tau(T_k) = \Omega(k)$

We fix the infinite pattern graph T_∞ with the threshold weighting θ induced by the uniform random walk. Recall that $T_\infty = \bigcup_{k=1}^\infty T_k$ under a nesting $T_1 \subset T_2 \subset T_3 \subset \cdots$ with $\text{Leaves}(T_1) \subset \text{Leaves}(T_2) \subset \text{Leaves}(T_3) \subset \cdots$. F, G, H will represent finite subgraphs of T_∞, and A, B, C, D, E will be join-trees over T_∞. (In particular, note that G no longer denotes the ambient pattern graph.)

We next recall the definition of θ from Example 1. Let $M \in [0,1]^{V(T_\infty) \times V(T_\infty)}$ be the transition matrix of the uniform random walk on T_∞, that is,

$$M_{v,w} = \begin{cases} 1 & \text{if } \{v,w\} \in E(T_\infty) \text{ and } v \text{ is a leaf,} \\ 1/3 & \text{if } \{v,w\} \in E(T_\infty) \text{ and } v \text{ is a non-leaf,} \\ 0 & \text{if } \{v,w\} \notin E(T_\infty). \end{cases}$$

This induces the threshold weighting $\theta : E(T_\infty) \to [0,2]$ given by

$$(2) \quad \theta(\{v,w\}) := M_{v,w} + M_{w,v} = \begin{cases} 4/3 & \text{if } v \text{ or } w \text{ is a leaf of } T_\infty, \\ 2/3 & \text{otherwise.} \end{cases}$$

Since θ is fixed, we will suppress it when writing $\Delta(F)$ and $\Phi(A)$.

Definition 9. For all $k \geq 0$, let

$$V_k := \{v \in V(T_k) : v \text{ has distance } k \text{ from a leaf}\}.$$

Thus, V_0 is the set of leaves in T_∞, V_1 is the set of parents of leaves, etc. Note that $V(T_\infty) = \bigcup_{k=0}^{\infty} V_k$. We shall refer to the V_k as the various *levels* of T_∞.

For $k \geq 1$ and $x \in V_k$, let $T_x \subset T_\infty$ be the complete binary tree of height k rooted at x (in the case $k = 0$, we regard T_x as a single isolated vertex). We denote by T_x^+ the graph obtained from T_x by including an extra edge between x and its parent. Note that

$$|V(T_x)| = 2^{k+1} - 1, \ |V(T_x^+)| = 2^{k+1},$$
$$|E(T_x)| = 2^{k+1} - 2, \ |E(T_x^+)| = 2^{k+1} - 1.$$

For $j \in \{0, \ldots, k\}$, let $V_j(T_x) := V_j \cap V(T_x)$.

Observation 1. If $x \in V_k$, then for $j \in \{0, \ldots, k\}$, $|V_j(T_x)| = 2^{k-j}$.

We next define two useful parameters of finite subgraphs of T_∞.

Definition 10 (Max-complete height)**.** For a finite subgraph F of T_∞, define the *max-complete height* $\lambda(F)$ to be the maximum $k \in \mathbb{N}$ for which there exists $x \in V_k$ with $T_x \subseteq G$; $\lambda(F)$ is defined to be zero when no such x exists (in particular, this happens when $V(F) \cap V_0 = \emptyset$).

Observation 2. For any $x \in V_k$, $\lambda(T_x) = \lambda(T_x^+) = k$.

Definition 11 (Boundary size)**.** Let $\partial(F)$ denote the size of *boundary* of F in T_∞:

$$\partial(F) := |V(F) \cap V(T_\infty \setminus F)|.$$

Observation 3. For any $x \in V_k$, we have $\partial(T_x) = \partial(T_x^+) = 1$, as the boundaries in the respective graphs are simply the singletons $\{x\}$ and $\{\text{parent}(x)\}$. Another example is as follows: if $x \in V_k$ for some $k \geq 2$ and F is the subgraph of T_x induced by the set of vertices $V(T_x) \setminus V_0$, then $\partial(F) = 2^{k-1} + 1$ as all vertices in $V_1(T_x)$ (along with x) lie in the boundary of F.

Definition 12 (Grounded and ungrounded subgraphs of T_∞)**.** Let F, H be finite subgraphs of T_∞. We say that F is *grounded* if it is connected and $V(F) \cap V_0 \neq \emptyset$ (that is, F is a tree, at least one of whose leaves is also a leaf of T_∞). We say that H is *ungrounded* if it is non-empty and connected and $V(H) \cap V_0 = \emptyset$ (that is, H is a non-empty tree, none of whose leaves is a leaf of T_∞).

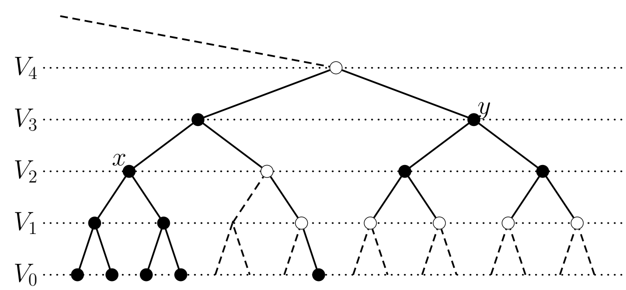

Figure 2: Example of a grounded graph $F \subset T_\infty$. The dotted lines indicate the various levels that $V(F)$ intersects and dashed lines indicate (some of the) edges in $T_\infty \setminus F$. The nodes colored white lie in the boundary of F, and therefore $\partial(F) = 7$ and $\Delta(F) = 11/3$. The max-complete height $\lambda(F) = 2$, since $T_x \subset F$ for $x \in V_2$ and x is the highest such node. The subgraph $H = F \cap T_y$ is ungrounded with $\lambda(H) = 0$.

We shall think of the function $\partial(F)$ as essentially a proxy for $\Delta(F)$, as it has the advantage of having a simple combinatorial definition. This is justified by the following:

Lemma 17. *For every $F \subset T_\infty$, we have $\Delta(F) \geq \partial(F)/3$.*

It also holds that $\Delta(F) \leq 2\partial(F)/3$ for F without isolated vertices (or $\Delta(F) \leq \partial(F)$ if we allow isolated vertices), but we will not need this upper bound.

Proof: Since θ is the threshold weighting induced by M, Lemma 10 tells us

$$\Delta(F) = \sum_{v \in V(F)} \sum_{w \in V(T_\infty) : \{v,w\} \in E(T_\infty \setminus F)} M_{v,w}.$$

Note that $v \in V(F)$ contributes to this sum if, and only if, it belongs to the boundary of F (i.e., $v \in V(F) \cap V(T_\infty \setminus F)$). Since $M_{v,w} \geq 1/3$ whenever $\{v,w\} \in E(F)$, the claim follows. ∎

We are now ready to state the main theorem of the section.

Theorem 18. *Let $\varepsilon = 1/30$ and $\delta = 2/5$. Then for every join-tree A,*

$$\Phi(A) \geq \varepsilon\lambda(A) + \delta\Delta(A).$$

Theorem 18 directly implies Lemma 15, which in turn yields the lower bound $\tau(T_k) = \Omega(k)$ of Theorem 6. To see why, let θ' be the threshold weighting on T_k coming from the uniform random walk (Example 1). Viewing T_k a subgraph of T_∞, note that $\sum_{e \in E(T_k)}(\theta'(e) - \theta(e)) = 2(\frac{5}{6} - \frac{2}{3}) = \frac{1}{3}$. For any join-tree A with graph T_k, Lemma 13 and Theorem 18 now imply

$$(3) \quad \Phi_{\theta'}(A) \geq \Phi(A) - \frac{1}{3} \geq \frac{1}{30}\lambda(T_k) + \frac{2}{5}\Delta(T_k) - \frac{1}{3} = \frac{1}{30}k + \frac{2}{5}\cdot\frac{1}{3} - \frac{1}{3} = \frac{1}{30}k - \frac{1}{5}.$$

We proceed with a few definitions and lemmas needed for the proof of Theorem 18. We present the main arguments here, and leave the proofs of three auxiliary lemmas (19, 20, 21) for the full version.

Lemma 19. *Let H be a non-empty finite subgraph of T_∞, all of whose components are ungrounded. Then $\partial(H) \geq \frac{1}{2}(|E(H)|+3)$.*

We make a note of its following corollary here.

Corollary 2. *Suppose H be a finite connected subgraph of T_∞ and $y \in V(H)$ such that $E(H) \cap E(T_y) \neq \emptyset$ and H does not contain any path from y to a leaf of T_y. Then $\partial(H) \geq \frac{1}{2}(|E(H) \cap E(T_y)|+1)$.*

Proof: Let F be the graph with edge set $E(F) := E(H) \cap E(T_y)$. Note that F is non-empty, connected and ungrounded. Observe that $\partial(H) \geq \partial(F) - 1$ because all vertices in the boundary of H, with the only possible exception of y, also lie in the boundary of F. Hence by Lemma 19, $\partial(H) \geq \frac{1}{2}(|E(H) \cap E(T_y)|+1)$. ■

The second auxiliary lemma gives a useful inequality relating $\partial(G)$, $\lambda(G)$ and $|E(G)|$.

Lemma 20. *For every finite subgraph G of T_∞, we have $\lambda(G) + \partial(G) \geq \log(|E(G)|+1)$.*

(This is tight when $G = T_x^+$ for some $x \in V_k$, in which case $\lambda(G) = k$ and $\partial(G) = 1$ and $|E(G)| = 2^{k+1} - 1$.) The third auxiliary lemma shows that subgraphs of T_k that contain at most half the edges of T_k and have boundary size j ($\leq k/2$) have empty intersection with a large complete subtree of T_k of height $k - j$.

Lemma 21. *Let $x \in V_k$ and suppose $G \subseteq T_x$ such that $|E(G \cap T_x)| \leq 2^k - 1$ and $\partial(G) \leq k/6$. Then there exists a vertex $z \in V_{k-\partial(G)}(T_x)$ such that $E(G) \cap E(T_z^+) = \emptyset$.*

We now state and prove the main lemma used in the proof of Theorem 18.

Lemma 22. *For any integers $1 \leq t \leq \ell$, let $z \in V_\ell$ and suppose A is a join-tree such that $T_z \subseteq G_A$. Then one of the following conditions holds:*

(i) *There exists $D \preceq A$ such that $\partial(D) \geq t$ and $\lambda(D) + \partial(D) \geq \ell$.*
(ii) *There exists $C \prec A$ with $\lambda((C \cap T_z) \ominus \{z\}) + \partial((C \cap T_z) \ominus \{z\}) \geq \ell - t$.*
(iii) *There exists $E \prec A$ such that $\partial(E) \geq \ell - t$.*

Proof: Descend in the join-tree A until reaching a $B \preceq A$ such that B contains a path P from z to a leaf of T_z, but no $B' \prec B$ contains a path from z to a leaf of T_z.

Let $j \in \{1, \ldots, \ell\}$ be maximal such that there exists $y \in V_j(T_z) \cap P$ such that $T_y \subseteq G_B$. We claim that $\partial(B) \geq \ell - j$. To see this, note that for every vertex $v \neq y$ on the path from z to y (a subpath of P), if $c(v)$ denotes the child of v that is not on the path P, then G does not contain $T_{c(v)}^+$ (because if it did then $\lambda(G) > j$). As a result, it must be the case that for every vertex $v \neq y$ on the path from z to y, some vertex in $V(G_B) \cap V(T_{c(v)}^+)$ lies in the boundary of G and so, $\partial(G) \geq \ell - j$.

Consider the case that $j \leq \ell - t$ (again see Figure **??**). Letting $D := B$, we have $\partial(D) \geq \ell - j \geq t$ and $\lambda(D) \geq j$, so condition (i) is satisfied. We shall therefore proceed under the assumption that

$$j \geq \ell - t + 1.$$

Since $T_y \subseteq G_B$, at least one child C of B satisfies

$$|E(C) \cap E(T_y)| \geq \frac{1}{2}|E(T_y)| = 2^j - 1.$$

Fix one such C.

Consider the case that C does not contain the path between z and y. Then $C \cap T_y \subseteq C \ominus \{z\}$, so by Lemma 20,

$$(4) \quad \lambda((C \cap T_z) \ominus \{z\}) + \partial((C \cap T_z) \ominus \{z\}) \geq \log(|E(C \cap T_y)|+1) \geq \log(2^j) \geq \ell - t + 1.$$

In this case, we satisfy condition (ii). We shall therefore proceed under the assumption that C contains the path between z and y.

Note that C does not contain a path from y to any leaf of T_y (since otherwise C would contain a path from z to a leaf of T_z, contradicting the way we choose $B \preceq A$). Let H be the connected component of G_C that contains y (and hence also contains the path between z and y).

We now consider two final cases, depending on the size of $|E(H) \cap E(T_y)|$. First, assume $|E(H) \cap E(T_y)| \geq 2(\ell - t)$. In this case, Corollary 2 implies

$$\partial(C) \geq \partial(H) \geq \frac{1}{2}\Big(|E(H) \cap E(T_y)|+1\Big) \geq \ell - t.$$

We satisfy condition (iii) setting $E := C$.

Finally, assume $|E(H) \cap E(T_y)| \leq 2(\ell - t) - 1$. We have

$$\begin{aligned} &|E((C \cap T_z) \ominus \{z\})| \\ &\geq |E(C) \cap E(T_y)| - |E(H) \cap E(T_y)| \\ &\geq (2^j - 1) - (2(\ell - t) - 1) \\ &\geq 2^{\ell - t + 1} - 2(\ell - t). \end{aligned}$$

Lemma 20 now implies

$$\begin{aligned} &\lambda((C \cap T_z) \ominus \{z\}) + \partial((C \cap T_z) \ominus \{z\}) \\ &\geq \log(|E((C \cap T_z) \ominus \{z\})|+1) \\ &\geq \log(2^{\ell - t + 1} - 2(\ell - t) + 1) \\ &> \ell - t \end{aligned}$$

since $2^{x+1} - 2x + 1 > 2^x$ for all $x \geq 0$. Therefore, condition (ii) is again satisfied in this final case. ■

We now prove Theorem 18: the lower bound $\Phi(A) \geq \varepsilon\lambda(A) + \delta\Delta(A)$ where $\varepsilon = 1/30$ and $\delta = 2/5$.

We argue by a structural induction on join-trees A. First, consider the case that G_A is empty, then $\Phi(A) = 0$ and $\varepsilon\lambda(A) + \delta\Delta(A) = 0$. We shall assume that G_A is non-empty.

Next consider the base case where A is the atomic join-tree $\langle e \rangle$ for an edge $e \in E(T_\infty)$. In this case, we have $\lambda(A) \leq 1$ and

$$\Phi(A) = \Delta(A) = \begin{cases} 2/3 & \text{if } e \text{ contains a leaf,} \\ 4/3 & \text{otherwise.} \end{cases}$$

Therefore, $\varepsilon\lambda(A) + \delta\Delta(A) \leq (1/30) + (2/5)(4/3) = 17/30$. We are done, since $\Phi(A) \geq 2/3 > \varepsilon\lambda(A) + \delta\Delta(A)$.

From now on, let A be a non-atomic join-tree with whose graph is non-empty. Let

$$k := \lambda(A).$$

Our goal is thus to prove $\Phi(A) \geq \varepsilon k + \delta\Delta(A)$, which we do by analyzing numerous cases.

Consider first the case that $k = 0$. In this case, we clearly have $\Phi(A) \geq \varepsilon k + \delta\Delta(A)$ (since $\Phi \geq \Delta \geq 0$ and $\delta < 1$). So shall proceed on the assumption that $k \geq 1$.

Since $\Phi \geq \Delta$, we are done if $\Delta(A) \geq \varepsilon k + \delta\Delta(A)$. So we shall proceed on the additional assumption that

$$\text{(5)} \qquad \Delta(A) \leq \frac{\varepsilon}{1-\delta}k = \frac{1}{18}k.$$

Fixing $x \in V_k$ with $T_x \subseteq G_A$: By definition of $\lambda(A)$, there exists a vertex $x \in T_k$ such that $T_k \subseteq G_A$. Let us fix any such x.

Fixing $B \preceq A$ with $2^{k-1} \leq |E(B) \cap E(T_x)| \leq 2^k - 1$: We next fix a sub-join-tree $B \preceq A$ satisfying $2^{k-1} \leq |E(B) \cap E(T_x)| \leq 2^k - 1$. To see that such B exists, first note that $|E(A) \cap E(T_x)| = |E(T_x)| = 2^{k+1} - 2$. Consider a walk down A which at each step descends to a child C which maximizes $|E(C) \cap E(T_x)|$. This quantity shrinks by a factor $\geq 1/2$ at each step, eventually reaching size 1. Therefore, at some stage, we reach a sub-join-tree B such that the intersection size $|E(B) \cap E(T_x)|$ is between 2^{k-1} and $2^k - 1$.

Observe that $\Phi(A) \geq \Phi(B) \geq \Delta(B)$ (by (†) and the fact that $\Phi \geq \Delta$ for all join-trees). Therefore, we are done if $\Delta(B) \geq \varepsilon k + \delta\Delta(A)$. So we shall proceed under the additional assumption that

$$\text{(6)} \quad \begin{aligned} \Delta(B) &\leq \varepsilon k + \delta\Delta(A) = \frac{1}{30}k + \frac{2}{5}\Delta(A) \\ &\leq \Big(\frac{1}{30} + \frac{2}{5}\cdot\frac{1}{18}\Big)k = \frac{1}{18}k. \end{aligned}$$

Since $|E(B)| \geq |E(B) \cap E(T_x)| \geq 2^{k-1}$, Lemma 20 tells us that $\lambda(B) + \partial(B) \geq k - 1$. We make note of the fact that this implies

(7)

$$\begin{aligned} \Phi(B) &\geq \varepsilon\lambda(B) + \delta\Delta(B) && \text{(induction hypothesis)} \\ &\geq \varepsilon\Big(k - \partial(B) - 1\Big) + \delta\Delta(B) \\ &\geq \varepsilon k + (\delta - 3\varepsilon)\Delta(B) - \varepsilon && \text{(using } -\partial \geq -3\Delta\text{).} \end{aligned}$$

Fixing $z \in V_{k-\partial(B)}(T_x)$ with $E(B) \cap E(T_z^+) = \emptyset$: Note that $\partial(B) \leq 3\Delta(B) \leq k/6$ by (6). Since $|E(B) \cap E(T_x)| \leq 2^k - 1$, the hypotheses of Lemma 21 are satisfied with respect to the vertex x and the graph $G_B \cap T_x$. Therefore, we may fix a vertex $z \in V_{k-\partial(B)}(T_x)$ such that $E(B) \cap E(T_z^+) = \emptyset$.

We next introduce a parameter

$$t := 6\Delta(A) + 3\Delta(B).$$

Note that t is an integer, since 3Δ is integral. Our choice of parameters moreover ensure that $1 \leq t \leq k/2$, since $\Delta(A), \Delta(B) \leq k/18$ by (5),(6) and $\Delta(A), \Delta(B) \geq 1/3$ for all nonempty graphs. Since $k - \partial(B) \geq k - 3\Delta(B) \geq 5k/6$, it follows that $t < k - \partial(B)$.

Since $z \in V_{k-\partial(B)}$ and $T_z \subseteq T_x \subseteq G_A$, Lemma 22 (with respect to z and $1 \leq t < \ell := k - \partial(B)$) tells us that one of the following conditions holds:

(i) There exists $D \preceq A$ such that $\partial(D) \geq t$ and $\lambda(D) + \partial(D) \geq k - \partial(B)$.
(ii) There exists $C \prec A$ with $\lambda((C \cap T_z) \ominus \{z\}) + \partial((C \cap T_z) \ominus \{z\}) \geq k - \partial(B) - t$.
(iii) There exists $E \prec A$ with $\partial(E) \geq k - \partial(B) - t$.

We show that $\Phi(A) \geq \varepsilon k + \delta\Delta(A)$ in each of these three cases. This follows in a rather straightforward manner by plugging in the bounds on various parameters obtained thus far, through the repeated use of (†) and (‡). We refer the reader to the full version of the paper for the details.

Acknowledgment

We thank the anonymous FOCS reviewers for their numerous helpful comments and suggestions.

References

[1] Kazuyuki Amano. k-Subgraph isomorphism on AC^0 circuits. *Computational Complexity*, 19(2):183–210, 2010.

[2] Julia Chuzhoy and Zihan Tan. Towards tight(er) bounds for the excluded grid theorem. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1445–1464. SIAM, 2019.

[3] Marek Cygan, Fedor V Fomin, Alexander Golovnev, Alexander S Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socała. Tight bounds for graph homomorphism and subgraph isomorphism. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1643–1649. SIAM, 2016.

[4] Wojciech Czerwiński, Wojciech Nadara, and Marcin Pilipczuk. Improved bounds for the excluded-minor approximation of treedepth. *arXiv preprint arXiv:1904.13077*, 2019.

[5] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173, 2005.

[6] Ken-ichi Kawarabayashi and Benjamin Rossman. A polynomial excluded-minor characterization of treedepth. In *29th ACM-SIAM Symposium on Discrete Algorithms*, pages 234–246, 2018.

[7] Robert Krauthgamer and Ohad Trabelsi. Conditional lower bound for subgraph isomorphism with a tree pattern. *arXiv preprint arXiv:1708.07591*, 2017.

[8] Yuan Li, Alexander Razborov, and Benjamin Rossman. On the AC^0 complexity of subgraph isomorphism. *SIAM Journal on Computing*, 46(3):936–971, 2017.

[9] Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6:85–112, 2010.

[10] Jaroslav Nešetřil and Patrice Ossona De Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, 2006.

[11] Gregory Rosenthal. Beating treewidth for average-case subgraph isomorphism. In *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:14, 2019.

[12] Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3):15, 2008.

[13] Benjamin Rossman. Correlation bounds against monotone NC^1. In *30th Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 392–411, 2015.

[14] Benjamin Rossman. An improved homomorphism preservation theorem from lower bounds in circuit complexity. In *8th Innovations in Theoretical Computer Science*, volume 67 of *LIPIcs*, pages 27:1–17, 2017.

[15] Benjamin Rossman. Formulas versus circuits for small distance connectivity. *SIAM Journal on Computing*, 47(5):1986–2028, 2018.

[16] Benjamin Rossman. Lower bounds for subgraph isomorphism. In *Proc. International Congress of Mathematicians (ICM 2018, Rio de Janeiro)*, volume 3, pages 3409–3430, 2018.