Near-optimal small-depth lower bounds
for small distance connectivity

Xi Chen[*]
Columbia University

Igor C. Oliveira[†]
Columbia University

Rocco A. Servedio[‡]
Columbia University

Li-Yang Tan[§]
Toyota Technological Institute

September 25, 2015

Abstract

We show that any depth-d circuit for determining whether an n-node graph has an s-to-t path of length at most k must have size $n^{\Omega(k^{1/d}/d)}$. The previous best circuit size lower bounds for this problem were $n^{k^{\exp(-O(d))}}$ (due to Beame, Impagliazzo, and Pitassi [BIP98]) and $n^{\Omega((\log k)/d)}$ (following from a recent formula size lower bound of Rossman [Ros14]). Our lower bound is quite close to optimal, since a simple construction gives depth-d circuits of size $n^{O(k^{2/d})}$ for this problem (and strengthening our bound even to $n^{k^{\Omega(1/d)}}$ would require proving that undirected connectivity is not in NC^1.)

Our proof is by reduction to a new lower bound on the size of small-depth circuits computing a skewed variant of the "Sipser functions" that have played an important role in classical circuit lower bounds [Sip83, Yao85, Hås86]. A key ingredient in our proof of the required lower bound for these Sipser-like functions is the use of *random projections*, an extension of random restrictions which were recently employed in [RST15]. Random projections allow us to obtain sharper quantitative bounds while employing simpler arguments, both conceptually and technically, than in the previous works [Ajt89, BPU92, BIP98, Ros14].

1 Introduction

Graph connectivity problems are of great interest in theoretical computer science, both from an algorithmic and a computational complexity perspective. The "st-connectivity," or $STCONN$, problem — given an n-node graph G with two distinguished vertices s and t, is there a path of edges from s to t? — plays a particularly central role. One longstanding question is whether any improvement is possible on Savitch's $O((\log n)^2)$-space algorithm [Sav70], based on "repeated squaring," for the *directed STCONN* problem; since this problem is complete for NL, any such improvement would show that $\mathsf{NL} \subseteq \mathsf{SPACE}(o(\log^2 n))$, and hence would have a profound impact on our understanding of non-deterministic space complexity. Wigderson's survey [Wig92] provides a now somewhat old, but still very useful, overview of early results on connectivity problems.

In this paper we consider the "small distance connectivity" problem $STCONN(k(n))$ which is defined as follows. The input is the adjacency matrix of an undirected n-vertex graph G which has two distinguished vertices s and t, and the problem is to determine whether G contains a path of length at most $k(n)$ from s to t. We study this problem from the perspective of small-depth circuit complexity; for a given depth d (which may depend on k), we are interested in the size of unbounded fan-in depth-d circuits of AND, OR, and NOT gates that compute $STCONN(k(n))$. (As several authors [BIP98, Ros14] have observed, the directed and undirected versions of the $STCONN(k(n))$ problems are essentially equivalent via a simple reduction that converts a directed graph into a layered undirected graph; for simplicity we focus on the undirected problem in this paper.)

An impetus for this study comes from the above-mentioned question about Savitch's algorithm. As noted by Wigderson [Wig92], a simple reduction shows that if Savitch's algorithm is optimal, then for all k polynomial-size unbounded fan-in circuits for $STCONN(k(n))$ must have depth $\Omega(\log k)$. By giving lower bounds on the size of small-depth circuits for $STCONN(k(n))$, Beame, Impagliazzo, and Pitassi [BIP98] have shown that depth $\Omega(\log \log k)$ is required for $k(n) \leq \log n$, and more recently Rossman [Ros14] has shown that depth $\Omega(\log k)$ is required for $k(n) \leq \log \log n$. These bounds for restricted ranges of k motivate further study of the circuit complexity of small-depth circuits for $STCONN(k(n))$. Below we give a more thorough discussion of both upper and lower bounds for this problem, before presenting our new results.

1.1 Prior results

Upper bounds (folklore). A natural approach to obtain efficient circuits for $STCONN(k(n))$ is by repeated squaring of the input adjacency matrix. If $x_{i,j}$ is the input variable that takes value 1 if edge $\{i, j\}$ is present in the input graph, then the graph contains a path of length at most 2 from i to j if and only if the depth-2 circuit $\bigvee_{k=1}^{n}(x_{i,k} \wedge x_{k,j})$ is satisfied (assuming that $x_{i,i} = 1$ for every i). Iterating this construction yields a circuit of size $\mathrm{poly}(n)$ and depth $2 \log k$ that computes $STCONN(k(n))$, whenever k is a power of two.

For smaller depths, a natural extension of this approach leads to the following construction. Let G_0 be the input graph. For every pair of nodes u, v in G_0, check by exhaustive search for paths of length at most $t = k^{1/d}$ connecting these nodes. (We assume that $k^{1/d}$ is an integer in order to avoid unnecessary technical details.) Note that this can be done simultaneously for every pair of nodes by a (multi-output) depth-2 OR-of-ANDs circuit of size $n^{t+O(1)}$. Let G_1 be a new graph that has an edge between u and v if and only if a path of length at most t connects these nodes. In general, if we start with G_0 and repeat this procedure d times, we obtain a sequence of graphs $G_0, G_1, \ldots, G_d$ for which the following holds: G_i has an edge between nodes u and v if and only

if they are connected by a path of length at most t^i in the initial graph G_0. In particular, this construction provides a circuit of depth $2d$ and size $n^{k^{1/d}+O(1)}$ that computes $STCONN(k(n))$.

Summarizing this discussion, it follows that for all $k \leq n$ and $d \leq \log k$, $STCONN(k(n))$ can be computed by depth-$2d$ circuits of size $n^{O(k^{1/d})}$, or equivalently by depth-d circuits of size $n^{O(k^{2/d})}$.

Lower bounds. Furst, Saxe, and Sipser [FSS84] were the first to show that $STCONN \overset{\text{def}}{=} STCONN(n) \notin \mathsf{AC}^0$ via a reduction from their lower bound against small-depth circuits computing the parity function. By the same reduction, Håstad's subsequent optimal lower bound against parity [Hås86] implies that depth-d circuits computing $STCONN(k(n))$ must have size $2^{\Omega(k^{1/(d+1)})}$; in particular, for $k(n) = (\log n)^{\omega(1)}$ polynomial-size circuits computing $STCONN(k(n))$ must have depth $d = \Omega(\log k / \log \log n)$. Note, however, that this is not a useful bound for *small* distance connectivity, since when $k(n) = o(\log n)$ the $2^{\Omega(k^{1/(d+1)})}$ lower bound is less than n and hence trivial.

Ajtai [Ajt89] was the first to show that $STCONN(k(n)) \notin \mathsf{AC}^0$ for all $k(n) = \omega_n(1)$; however, his proof did not yield an explicit circuit size lower bound. His approach was further analyzed and simplified by Bellantoni, Pitassi, and Urquhart [BPU92], who showed that this technique gives a (barely super-polynomial) $n^{\Omega(\log^{(d+3)} k)}$ lower bound on the size of depth-d circuits for $STCONN(k(n))$, where $\log^{(i)}$ denotes the i-times iterated logarithm. This implies that polynomial-size circuits computing $STCONN(k(n))$ must have depth $\Omega(\log^* k)$.

Beame, Impagliazzo, and Pitassi [BIP98] gave a significant quantitative strengthening of Ajtai's result in the regime where $k(n)$ is not too large. For $k(n) \leq \log n$, they showed that any depth-d circuit for $STCONN(k(n))$ must have size $n^{\Omega(k^{\phi^{-2d}/3})}$, where $\phi = (\sqrt{5}+1)/2$ is the golden ratio. Their arguments are based on a special-purpose "connectivity switching lemma" that they develop, which combines elements of both the Ajtai [Ajt83] "independent set style" switching lemma and the later approach to switching lemmas given by Yao [Yao85], Hastad [Hås86] and Cai [Cai86].

Observe that the [BIP98] lower bound shows that polynomial-size circuits for $STCONN(k(n))$ require depth $\Omega(\log \log k)$ (and as noted above, the [BIP98] lower bound only holds for $k(n) \leq \log n$). Beame et al. asked whether this $\Omega(\log \log k)$ could be improved to $\Omega(\log k)$, which is optimal by the upper bound sketched above. This was achieved recently by Rossman [Ros14], who showed that for $k(n) \leq \log \log n$, polynomial-size circuits for $STCONN(k(n))$ require depth $\Omega(\log k)$. In more detail, he showed that for $k(n) \leq \log \log n$ and $d(n) \leq \log n / (\log \log n)^{O(1)}$, depth-$d$ *formulas* for $STCONN(k(n))$ require size $n^{\Omega(\log k)}$. By the trivial relation between formulas and circuits (every circuit of size S and depth d is computed by a formula of size S^d and depth d), this implies that for such $k(n)$ and $d(n)$, depth-d *circuits* for $STCONN(k(n))$ require size $n^{\Omega((\log k)/d)}$. While this answers the question of Beame et al., the $n^{\Omega((\log k)/d)}$ circuit size bound that follows from Rossman's formula size bound is significantly smaller than the $n^{\Omega(k^{\phi^{-2d}/3})}$ circuit size bound of [BIP98] when d is small. Furthermore, Rossman's result only holds for $k(n) \leq \log \log n$ whereas [BIP98]'s holds for $k(n) \leq \log n$ (and ideally we would like a lower bound for all distances $k(n) \leq n$).

1.2 Our results

Our main result is a near-optimal lower bound for the small-depth circuit size of $STCONN(k(n))$ for all distances $k(n) \leq n$. We prove the following:

Theorem 1. *For any $k(n) \leq n^{1/5}$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size $n^{\Omega(k^{1/d}/d)}$. Furthermore, for any $k(n) \leq n$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size $n^{\Omega(k^{1/5d}/d)}$.*

2

	Circuit size	Depth of poly-size circuits	Range of k's
Implicit in [Hås86]	$2^{\Omega(k^{1/(d+1)})}$	$\Omega(\log k/\log\log n)$	All k
[Ajt89, BPU92]	$n^{\Omega(\log^{(d+3)} k)}$	$\Omega(\log^* k)$	All k
[BIP98]	$n^{\Omega(k^{\phi^{-2d}/3})}$	$\Omega(\log\log k)$	$k \le \log n$
[Ros14]	$n^{\Omega((\log k)/d)}$	$\Omega(\log k)$	$k \le \log\log n$
Folklore upper bound	$n^{O(k^{2/d})}$	$2\log k$	All k
This work	$n^{\Omega(k^{1/d}/d)}$ $n^{\Omega(k^{1/5d}/d)}$	$\Omega(\log k/\log\log k)$	$k \le n^{1/5}$ All k

Table 1: Previous work and our results on the size of depth-d circuits for $STCONN(k(n))$. The column "Range of k's" indicates the values of k for which the lower bound is proved to hold.

Our lower bound is very close to the best possible, given the $n^{O(k^{2/d})}$ upper bound. Indeed, strengthening our theorem to $n^{k^{\Omega(1/d)}}$ for all values of k and d would imply a breakthrough in circuit complexity, showing that unbounded fan-in circuits of depth $o(\log n)$ computing $STCONN$ must have super-polynomial size. Since every function in NC^1 can be computed by unbounded fan-in circuits of polynomial size and depth $O(\log n/\log\log n)$ (see e.g. [KPPY84]), such a strengthening would yield an unconditional proof that $STCONN \notin \mathsf{NC}^1$.

Comparing to previous work, our $n^{\Omega(k^{1/d}/d)}$ lower bound subsumes the main $n^{\Omega(k^{\phi^{-2d}/3})}$ lower bound result of Beame et al. [BIP98] for all depths d, and improves the $n^{\Omega((\log k)/d)}$ circuit size lower bound that follows from Rossman's formula size lower bound [Ros14] except when d is quite close to $\log k$ (specifically, except when $\Omega(\log k/\log\log k) \le d \le O(\log k)$). For large distances $k(n)$ for which the results of [BIP98, Ros14] do not apply (i.e. $k(n) = \omega(\log n)$), our lower bound subsumes the $2^{\Omega(k^{1/(d+1)})}$ lower bound that is implied by [Hås86] for all distances $k(n) \le n^{1/5}$ and depths d, and it subsumes the subsequent $n^{\Omega(\log^{(d+3)} k)}$ lower bound of [Ajt89, BPU92] for all distances k and depths d.

Another perspective on Theorem 1 is that it implies that polynomial-size circuits require depth $\Omega(\log k/\log\log k)$ to compute $STCONN(k(n))$ for all distances $k(n) \le n$. While Rossman's results give $\Omega(\log k)$, they hold only for the significantly restricted range $k(n) \le \log\log n$. (And indeed, as noted above a lower bound of $\Omega(\log k)$ for all $k(n)$ would imply that $STCONN \notin \mathsf{NC}^1$.)

1.3   Our approach

Previous state-of-the-art results on this problem employed rather sophisticated arguments and involved machinery. Beame et al. [BIP98] (as well as the earlier works of [Ajt89, BPU92]) obtained their lower bounds by considering the $STCONN(k(n))$ problem on layered graphs of permutations, i.e., graphs with $k + 1$ layers of n vertices per layer in which the induced graph between adjacent layers is a perfect bipartite matching. They developed a special-purpose "connectivity switching lemma" that bounds the depth of specialized decision trees for randomly-restricted layered graphs. Rossman [Ros14] considered random subgraphs of the "complete k-layered graph" (with $k+1$ layers

3

of n vertices and kn^2 edges) where each edge is independently present with probability $1/n$. At the heart of his proof is an intricate notion of "pathset complexity," which roughly speaking measures the minimum cost of constructing a set of paths via the operations of union and relational join, subject to certain "density constraints."

In contrast, we feel that our approach is both conceptually and technically simple. Instead of working with layered permutation graphs or random subgraphs of the complete layered graph, we consider a class of series-parallel graphs that are obtained in a straightforward way (see Section 3) from a skewed variant of the "Sipser functions" that have played an important role in the classical circuit lower bounds of Sipser [Sip83], Yao [Yao85], and Håstad [Hås86]. Briefly, for every $d \in \mathbb{N}$, the d-th Sipser function Sipser_d is a read-once monotone formula with d alternating layers of AND and OR gates of fan-in w, where $w \in \mathbb{N}$ is an asymptotic parameter that tends to ∞ (and so Sipser_d computes an $n = w^d$ variable function). Building on the work of Sipser and Yao, Håstad used the Sipser functions[1] to prove an optimal depth-hierarchy theorem for circuits, showing that for every $d \in \mathbb{N}$, any depth-d circuit computing Sipser_{d+1} must have size $\exp(n^{\Omega(1/d)})$.

The skewed variant of the Sipser functions that we use to prove our near-optimal lower bounds for $STCONN(k(n))$ is as follows. For every $d \in \mathbb{N}$ and $2 \le u \le w$, the d-th u-*skewed Sipser function*, denoted $\mathsf{SkewedSipser}_{u,d}$, is essentially Sipser_{2d+1} but with the AND gates having fan-in u rather than w (see Section 3 for a precise definition; as we will see, the number of levels of AND gates is the key parameter for $\mathsf{SkewedSipser}$, which is why we write $\mathsf{SkewedSipser}_{u,d}$ to denote the n-variable formula that has d levels of AND gates and $d + 1$ levels of OR gates.) Via a simple reduction given in Section 3, we show that to get lower bounds for depth-d circuits computing $STCONN(u^d)$ on n-node graphs, it suffices to prove that depth-d circuits for $\mathsf{SkewedSipser}_{u,d}$ must have large size. Under this reduction the fan-in of the AND gates is directly related to the length of (potential) paths between s and t. This is why we must use a *skewed* variant of the Sipser function in order to obtain lower bounds for small distance connectivity. We remark that even the case $u = 2$ is interesting and can be used to get the $n^{\Omega(k^{1/d}/d)}$ lower bound of Theorem 1 for k up to roughly $2^{\sqrt{\log n}}$. Allowing a range of values for u enables us to get the lower bound for k up to $n^{1/5}$ (as stated in Theorem 1).

Our main technical result of the paper is a lower bound for $\mathsf{SkewedSipser}_{u,d}$, a formula of depth $2d + 1$ over $n = n(u, w, d) = u^d w^{d+33/100}$ variables (for technical reasons we use a smaller fan-in for the first layer of OR gates next to the inputs).

Theorem 2. *Let $d(w) \ge 1$ and $2 \le u(w) \le w^{33/100}$, where $w \to \infty$. Then any depth-d circuit computing $\mathsf{SkewedSipser}_{u,d}$ has size at least $w^{\Omega(u)} = n^{\Omega(u/d)}$.*

Observe that setting $u = k^{1/d}$ this size lower bound is $n^{\Omega(k^{1/d}/d)}$, and therefore we indeed obtain the lower bound for $STCONN(k(n))$ stated in Theorem 1 as a corollary. As we point out in Section 6 (Remark 18), the lower bound given in Theorem 2 for $\mathsf{SkewedSipser}$ is essentially optimal.

Though they are superficially similar, Theorem 2 and Håstad's depth hierarchy theorem differ in two important respects. Both result from our goal of using Theorem 2 to get lower bounds for small distance connectivity, and both pose significant challenges in extending Håstad's proof:

1. Håstad showed that depth-d unbounded fan-in circuits require large size to compute a single highly symmetric "hard function," namely Sipser_{d+1}. In contrast, toward our goal of understanding the depth-d circuit size of $STCONN(k(n))$ for *all values of $k = k(n)$ and $d = d(n)$*,

---

[1]The exact definition of the function used in [Hås86] differs slightly from our description for some technical reasons which are not important here.

4

we seek lower bounds on the size of depth-d unbounded fan-in circuits computing any one of a spectrum of asymmetric hard functions, namely $\mathsf{SkewedSipser}_{u,d}$ for all $u := k^{1/d}$ (with stronger quantitative bounds as k and u get larger).

2. To get the strongest possible result in his depth hierarchy theorem, Håstad (like Yao and Sipser) was primarily focused on lower bounding the size of circuits of depth exactly one less than Sipser_{d+1}. In contrast, since in our framework our goal is to lower bound the size of depth-d circuits computing $\mathsf{SkewedSipser}_{u,d}$ (corresponding to $STCONN(k(n))$ with $k = u^d$) which has depth $2d + 1$, we are interested in the size of circuits of depth (roughly) *half* that of our hard function $\mathsf{SkewedSipser}_{u,d}$.

In Section 2 we recall the high-level structure of Håstad's proof of his depth hierarchy theorem (based on the method of random restrictions), highlight the issues that arise due to each of the two differences above, and describe how our techniques — specifically, the method of *random projections* — allow us to prove Theorem 2 in a clean and simple manner.

2 Håstad's depth hierarchy theorem, random projections, and proof outline of Theorem 2

Håstad's depth hierarchy theorem and its proof. Recall that Håstad's depth hierarchy theorem shows that Sipser_{d+1} cannot be computed by any circuit C of depth d and size $\exp(n^{O(1/d)})$. The main idea is to design a sequence of random restrictions $\{\mathcal{R}_\ell\}_{2 \le \ell \le d}$ satisfying two competing requirements:

- **Circuit C collapses.** The randomly restricted circuit $C \upharpoonright \boldsymbol{\rho}^{(d)} \cdots \boldsymbol{\rho}^{(2)}$, where $\boldsymbol{\rho}^{(\ell)} \leftarrow \mathcal{R}_\ell$ for $2 \le \ell \le d$, collapses to a "simple function" with high probability. This is shown via iterative applications of a switching lemma for the $\mathcal{R}_\ell$'s, where each application shows that with high probability a random restriction $\boldsymbol{\rho}^{(\ell)}$ decreases the depth of the circuit $C \upharpoonright \boldsymbol{\rho}^{(d)} \cdots \boldsymbol{\rho}^{(\ell+1)}$ by at least one. The upshot is that while C is a size-S depth-d circuit, $C \upharpoonright \boldsymbol{\rho}^{(d)} \cdots \boldsymbol{\rho}^{(2)}$ collapses to a small-depth decision tree (i.e. a "simple function") with high probability.

- **Hard function Sipser_{d+1} retains structure.** In contrast with the circuit C, the hard function Sipser_{d+1} is "resilient" against the random restrictions $\boldsymbol{\rho}^{(\ell)} \leftarrow \mathcal{R}_\ell$. In particular, each random restriction $\boldsymbol{\rho}^{(\ell)}$ simplifies Sipser only by one layer, and so $\mathsf{Sipser}_{d+1} \upharpoonright \boldsymbol{\rho}^{(d)} \cdots \boldsymbol{\rho}^{(\ell)}$ contains Sipser_ℓ as a subfunction with high probability. Therefore, with high probability. $\mathsf{Sipser}_{d+1} \upharpoonright \boldsymbol{\rho}^{(d)} \cdots \boldsymbol{\rho}^{(2)}$ still contains Sipser_2 as a subfunction, and hence is a "well-structured function" which cannot be computed by a small-depth decision tree.

We remind the reader that to satisfy these competing demands, the random restrictions $\{\mathcal{R}_\ell\}$ devised by Håstad specifically for his depth hierarchy theorem are not the "usual" random restrictions where each coordinate is independently kept alive with probability $p \in (0, 1)$, and set to a uniform bit otherwise (it is not hard to see that Sipser does not retain structure under these random restrictions). Likewise, the switching lemma for the $\mathcal{R}_\ell$'s is not the "standard" switching lemma (which Håstad used to obtain his optimal lower bounds against the parity function). Instead, at the heart of Håstad's proof are new random restrictions $\{\mathcal{R}_\ell\}_{2 \le \ell \le d}$ designed to satisfy both requirements above: the coordinates of $\mathcal{R}_\ell$ are carefully correlated so that $\mathsf{Sipser}_{\ell+1}$ retains structure, and

Håstad proved a special-purpose switching lemma showing that C collapses under these carefully tailored new random restrictions.

Issues that arise in our setting. At a technical level (related to point (1) described at the end of Section 1), Håstad's special-purpose switching lemma is not useful for analyzing our $\mathsf{SkewedSipser}_{u,d}$ formulas for most values of $u = k^{1/d}$ of interest, since they have a "fine structure" that is destroyed by his too-powerful random restrictions. His switching lemma establishes that any DNF of width $n^{O(1/d)}$ collapses to a small-depth decision tree with high probability when it is hit by a random restriction $\boldsymbol{\rho}^{(\ell)} \leftarrow \mathcal{R}_\ell$. Observe that his hard function Sipser_{d+1} has DNF-width $\Omega(\sqrt{n})$, so his switching lemma does not apply to it (and indeed as discussed above, hitting Sipser_{d+1} with his random restriction results in a well-structured function that still contains Sipser_d as a subfunction with high probability). In contrast, in our setting the hard function $\mathsf{SkewedSipser}_{u,d}$ has d levels of AND gates of fan-in u, and in particular, can be written as a DNF of width $u^d = k$. So for all $k = k(n)$ and $d = d(n)$ such that $k \ll n^{O(1/d)}$ (indeed, this holds for most values of k and d of interest), the relevant hard function $\mathsf{SkewedSipser}_{u,d}$ collapses to a small-depth decision tree after a single application of Håstad's random restriction.

Next (related to point (2)), recall that the formula computing Håstad's hard function Sipser_{d+1} has a highly regular structure where the fan-ins of all gates — both AND's and OR's — are the same. As discussed above, Håstad employs a random restriction which (with high probability) "peels off" a single layer of Sipser_{d+1} and results in a function that contains Sipser_d as a subfunction. Due to their regular structures, Sipser_d is dual to Sipser_{d+1} (more precisely, the bottom-layer depth-2 subcircuits of Sipser_d are dual to those of Sipser_{d+1}), and this allows Håstad to repeat the same procedure $d-1$ times. In contrast, in our setting we are dealing with the highly asymmetric $\mathsf{SkewedSipser}_{u,d}$ formulas where the fan-ins of the AND gates are much less than those of the OR gates. Therefore, in order to reduce to a smaller instance of the same problem, our setup requires that we peel off two layers of $\mathsf{SkewedSipser}_{u,d}$ at a time rather than just one as in Håstad's argument. To put it another way, while Håstad's switching lemma uses a single layer of his hard function Sipser_{d+1} (i.e. disjoint copies of OR's/AND's of fan-in w) to "trade for" one layer of depth reduction in C, our switching lemma will use two layers of our hard function $\mathsf{SkewedSipser}_{u,d}$ (i.e. disjoint copies of read-once CNF's with $u = k^{1/d}$ clauses of width w) to trade for one layer of depth reduction in C.

Our approach: random projections. A key technical ingredient in Håstad's proof of his depth hierarchy theorem — and indeed, in the works of [BIP98, Ros14] on $STCONN(k(n))$ as well — is the *method of random restrictions*. In particular, they all employ *switching lemmas* which show that a randomly-restricted small-width DNF collapses to a small-depth decision tree with high probability: as mentioned above, Håstad proved a special-purpose switching lemma for random restrictions tailored for the Sipser functions, while Beame et al. developed a "connectivity switching lemma" for random restrictions of layered permutation graphs, and Rossman used Håstad's "usual" switching lemma in conjunction with his pathset complexity machinery.

In this paper we work with *random projections*, a generalization of random restrictions. Given a set of formal variables $\mathcal{X} = \{x_1, ..., x_n\}$, a restriction $\boldsymbol{\rho}$ either fixes a variable x_i (i.e. $\boldsymbol{\rho}(x_i) \in \{0, 1\}$) or keeps it alive (i.e. $\boldsymbol{\rho}(x_i) = x_i$, often denoted by $*$). A *projection*, on the other hand, either fixes x_i or maps it to a variable y_j from a possibly different space of formal variables $\mathcal{Y} = \{y_1, ..., y_m\}$. Restrictions are therefore a special case of projections where $\mathcal{Y} \equiv \mathcal{X}$, and each x_i can only be fixed or mapped to itself. (See Section 4 for precise definitions.) Our arguments crucially employ projections in which $\mathcal{Y}$ is smaller than $\mathcal{X}$, and where moreover each x_i is only mapped to a specific

element y_j where j depends on i in a carefully designed way that depends on the structure of the formula computing the SkewedSipser function. Such "collisions", where multiple formal variables in $\mathcal{X}$ are mapped to the same new formal variable $y_j \in \mathcal{Y}$, play an important role in our approach.

Random projections were used in the recent work of Rossman, Servedio, and Tan [RST15], where they are the key ingredient enabling that paper's average-case extension of Håstad's worst-case depth hierarchy theorem. In earlier work, Impagliazzo, Paturi, and Saks [IPS97] used random projections to obtain size-depth tradeoffs for threshold circuits, and Impagliazzo and Segerlind [IS01] used them to establish lower bounds against constant-depth Frege systems in proof complexity. Our work provides further evidence for the usefulness of random projections in obtaining strong lower bounds: random projections allow us to obtain sharper quantitative bounds while employing simpler arguments, both conceptually and technically, than in the previous works [Ajt89, BPU92, BIP98, Ros14] on the small-depth complexity of $STCONN(k(n))$.

We remark that although [RST15] and this work both employ random projections to reason about the Sipser function (and its skewed variants), the main advantage offered by projections over restrictions are different in the two proofs. In [RST15] the overarching challenge was to establish *average-case* hardness, and the identification of variables was key to obtaining uniform-distribution correlation bounds from the composition of highly-correlated random projections. As outlined above, in this work a significant challenge stems from our goal of understanding the depth-d circuit size of $STCONN(k(n))$ for *all values of $k = k(n)$ and $d = d(n)$*. The added expressiveness of random projections over random restrictions is exploited both in the proof of our projection switching lemma (see Section 2.1 below) and in the arguments establishing that our SkewedSipser$_{u,d}$ functions "retain structure" under our random projections.

2.1 Proof outline of Theorem 2

Our approach shares the same high-level structure as Håstad's depth hierarchy theorem, and is based on a sequence $\boldsymbol{\Psi}$ of $d - 1$ random projections satisfying two competing requirements (it will be more natural for us to present them in the opposite order from our discussion of Håstad's theorem in the previous section):

- **Hard function SkewedSipser retains structure.** Our random projections are defined with the hard function SkewedSipser in mind, and are carefully designed so as to ensure that SkewedSipser$_{u,d}$ "retains structure" with high probability under their composition $\boldsymbol{\Psi}$.

  In more detail, each of the $d - 1$ individual random projections comprising $\boldsymbol{\Psi}$ peels off two layers of SkewedSipser, and a randomly projected SkewedSipser$_{u,\ell}$ contains SkewedSipser$_{u,\ell-1}$ as a subfunction with high probability. These individual random projections are simple to describe: each bottom-layer depth-2 subcircuit of SkewedSipser$_{u,\ell}$ (a read-once CNF with $u = k^{1/d}$ clauses of width w) independently "survives" with probability $q \in (0, 1)$ and is "killed" with probability $1 - q$ (where q is a parameter of the restrictions), and

  - if it survives, all uw variables in the CNF are **projected** to the same fresh formal variable (with different CNFs mapped to different formal variables);
  - if it is killed, all its variables are **fixed** according to a random 0-assignment of the CNF chosen uniformly from a particular set of $2u$ many 0-assignments.

  In other words, each bottom-layer depth-2 subcircuit independently simplifies to a fresh formal variable (with probability q) or the constant 0 (with probability $1 - q$). With the appropriate

7

definition of SkewedSipser and choice of q, it is easy to verify that indeed a randomly projected SkewedSipser$_{u,\ell}$ contains SkewedSipser$_{u,\ell-1}$ as a subfunction with high probability. (For this to happen, the fanin of the bottom OR gates of SkewedSipser is chosen to be moderately smaller than w, the fanin of all other OR gates in SkewedSipser; see Definition 3 for details.)

- **Circuit C collapses.** In contrast with SkewedSipser$_{u,d}$, any depth-d circuit C of size $n^{O(u/d)}$ collapses to a small-depth decision tree under $\boldsymbol{\Psi}$ with high probability. Following the standard "bottom-up" approach to proving lower bounds against small-depth circuits, we establish this by arguing that each of the individual random projections comprising $\boldsymbol{\Psi}$ "contributes to the simplification" of C by reducing its depth by (at least) one.

  More precisely, in Section 5 we prove a *projection switching lemma*, showing that a small-width DNF or CNF "switches" to a small-depth decision tree with high probability under our random projections. (The depth reduction of C follows by applying this lemma to every one of its bottom-level depth-2 subcircuits.) Recall that the random projection of a depth-2 circuit over a set of formal variables $\mathcal{X}$ yields a function over a new set of formal variables $\mathcal{Y}$, and in our case $\mathcal{Y}$ is significantly smaller than $\mathcal{X}$. In addition to the structural simplification that results from setting variables to constants (as in the switching lemmas of [Hås86, BIP98, Ros14] for random *restrictions*), the proof of our projection switching lemma also exploits the additional structural simplification that results from distinct variables in $\mathcal{X}$ being mapped to the same variable in $\mathcal{Y}$.

2.2 Preliminaries

A *restriction* over a finite set of variables A is an element of $\{0,1,*\}^A$. We define the *composition* $\rho\rho'$ of two restrictions $\rho, \rho' \in \{0,1,*\}^A$ over a set of variables A to be the restriction

$$(\rho\rho')_\alpha \stackrel{\text{def}}{=} \begin{cases} \rho_\alpha & \text{if } \rho_\alpha \neq * \\ \rho'_\alpha & \text{otherwise} \end{cases}, \quad \text{for all } \alpha \in A.$$

A *DNF* is an OR of ANDs (terms) and a *CNF* is an AND of ORs (clauses). The *width* of a DNF (respectively, CNF) is the maximum number of variables that occur in any one of its terms (respectively, clauses).

The *size* of a circuit is its number of gates, and the *depth* of a circuit is the length of its longest root-to-leaf path. We count input variables as gates of a circuit (so any circuit for a function that depends on all n input variables trivially has size at least n). We will assume throughout the paper that circuits are *alternating*, meaning that every root-to-leaf path alternates between AND gates and OR gates. We also assume that circuits are *layered*, meaning that for every gate G, every root-to-G path has the same length. These assumptions are without loss of generality as by a standard conversion (see e.g. the discussion at [Sta]), every depth-d size-S circuit is equivalent to a depth-d alternating layered circuit of size at most poly(S) (this polynomial increase is offset by the "$\Omega(\cdot)$" notation in the exponent of all of our theorem statements.)

3   Lower bounds against SkewedSipser yield lower bounds for small distance connectivity

In this section we define $\mathsf{SkewedSipser}_{u,d}$ and show that computing this formula on a particular input z is equivalent to solving small-distance connectivity on a certain undirected (multi)graph $G(z)$. In a bit more detail, every input z corresponds to a subgraph $G(z)$ of a fixed ground graph G that depends only on $\mathsf{SkewedSipser}_{u,d}$. (Jumping ahead, we associate each input bit of $\mathsf{SkewedSipser}_{u,d}$ with an edge of its corresponding ground graph G.) Roughly speaking, AND gates translate into sequential paths, while OR gates correspond to parallel paths. After defining $\mathsf{SkewedSipser}_{u,d}$ and describing this reduction, we give the proof of Theorem 1, assuming Theorem 2.

The $\mathsf{SkewedSipser}$ formula is defined in terms of an integer parameter w; in all our results this is an asymptotic parameter that approaches $+\infty$, and so w should be thought of as "sufficiently large" throughout the paper.

Definition 3. For $2 \leq u \leq w$ and $d \geq 0$, $\mathsf{SkewedSipser}_{u,d}$ is the Boolean function computed by the following monotone read-once formula:

- There are $2d+1$ alternating layers of OR and AND gates, where the top and bottom-layer gates are OR gates. (So there are $d+1$ layers of OR gates and d layers of AND gates.)

- AND gates all have fan-in u.

- OR gates all have fan-in w, except bottom-layer OR gates which have fan-in $w^{33/100}$; we assume that $w^{1/100}$ is an integer throughout the paper. (The most important thing about the constant $33/100$ in the above definition is that it is less than 1; the particular value $33/100$ was chosen for technical reasons so that we could get the constant 5 in Theorem 1.)

Consequently, $\mathsf{SkewedSipser}_{u,d}$ is a Boolean function over $n = (uw)^d w^{33/100}$ variables in total.

From $\mathsf{SkewedSipser}_{u,d}$ to small-distance connectivity. There is a natural correspondence between read-once monotone Boolean formulas and series-parallel multigraphs in which each graph has a special designated "start" node s and a special designated "end" node t. We now describe this correspondence via the inductive structure of read-once monotone Boolean formulas. As we shall see, under this correspondence there is a bijection between the *variables* of a formula f and the *edges* of the graph $G(f)$.

- If $f(x) = x$ is a single variable, then the graph $G(f)$ has vertex set $V(f) = \{s, t\}$ and edge set $E(f)$ consisting of a single edge $\{s, t\}$.

- Let $f_1, \ldots, f_m$ be read-once monotone Boolean formulas over disjoint sets of variables, where $G(f_i)$ is the (multi)graph associated with f_i and s_i, t_i are the start and end nodes of $G(f_i)$.

  - If $f = \mathrm{AND}(f_1, \ldots, f_m)$: The graph $G(f)$ is obtained by identifying t_1 with s_2, t_2 with s_3, $\ldots$, and t_{m-1} with s_m. The start node of $G(f)$ is s_1 and the end node is t_m. Thus the vertex set $V(f)$ is $V(f_1) \cup \cdots \cup V(f_m) \setminus \{t_1, \ldots, t_{m-1}\}$ and the edge set $E(f)$ is the multiset $E'(f_1) \cup \cdots \cup E'(f_m)$, where each $E'(f_i)$ is obtained from $E(f_i)$ by renaming the appropriate vertices.
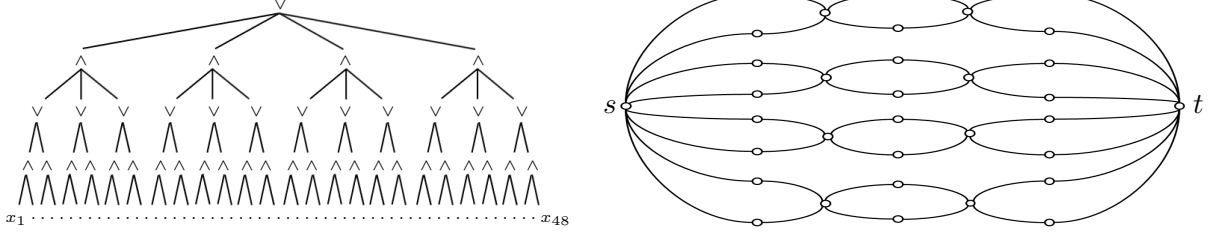
Figure 1: A read-once formula f (on the left), which is a fan-in 4 OR of fan-in 3 ANDs of fan-in 2 ORs of fan-in 2 ANDs, and the corresponding graph $G(f)$ (on the right).

- If $f = \mathrm{OR}(f_1, \ldots, f_m)$: The graph $G(f)$ is obtained by identifying $s_1, \ldots, s_m$ all to a new start vertex s and $t_1, \ldots, t_m$ all to a new end vertex t. Thus the vertex set $V(f)$ is $V(f_1) \cup \cdots \cup V(f_m) \cup \{s, t\} \setminus \{s_1, \ldots, s_m, t_1, \ldots, t_m\}$ and the edge set $E(f)$ is the multiset $E'(f_1) \cup \cdots \cup E'(f_m)$, where again each $E'(f_i)$ is obtained from the corresponding edge set $E(f_i)$ by renaming vertices accordingly.

Since f is read-once, the number of edges of $G(f)$ is precisely the number of variables of f, and there is a natural correspondence between edges and variables. Figure 1 provides a concrete example of this construction.

Remark 4. We note that if f is a read-once monotone Boolean formula in which the bottom-level gates are AND gates and have fan-in at least two, then $G(f)$ is a *simple* graph and not a multigraph.

A simple inductive argument gives the following:

Observation 5. *If f is a read-once monotone alternating formula with r layers of* AND *gates of fan-ins $\alpha_1, \ldots, \alpha_r$, respectively, then every shortest path from s to t in the graph $G(f)$ has length exactly $\alpha_1 \cdots \alpha_r$. Furthermore, if H is a subgraph of $G(f)$ that contains some s-to-t path, then it contains a path of length $\alpha_1 \cdots \alpha_r$.*

As a corollary, we have:

Observation 6. *Every shortest path from s to t in $G(\mathsf{SkewedSipser}_{u,d})$ has length exactly u^d.*

Given a read-once monotone formula f over variables $x_1, \ldots, x_n$ and an assignment $z \in \{0, 1\}^n$ to the variables $x_1, \ldots, x_n$, we define the graph $G(f, z)$ to be the (spanning) subgraph of $G(f)$ which has vertex set $V(f, z) = V(f)$ and edge set $E(f, z)$ defined as follows: each edge in $E(f)$ is present in $E(f, z)$ if and only if the corresponding coordinate of z is set to 1. A simple inductive argument gives the following:

Observation 7. *Given a read-once monotone alternating formula with r layers of* AND *gates of fan-ins $\alpha_1, \ldots, \alpha_r$, respectively, and an assignment $z \in \{0, 1\}^n$, the graph $G(f, z)$ contains a path from s to t of length $\alpha_1 \cdots \alpha_r$ if and only if $f(z) = 1$.*

From these observations we obtain the following connection between $\mathsf{SkewedSipser}_{u,d}$ and small-distance connectivity, which is key to our lower bound:

Corollary 3.1. *The multigraph $G(\mathsf{SkewedSipser}_{u,d}, z)$ contains an s-to-t path of length at most u^d if and only if $\mathsf{SkewedSipser}_{u,d}(z) = 1$.*

10

Note that Corollary 3.1 and Theorem 2 together can be used to prove lower bounds for small-distance connectivity on *multigraphs*. One way to obtain lower bounds for *simple* graphs instead of multigraphs is by extending $\mathsf{SkewedSipser}_{u,d}$ with an extra layer of fan-in two AND gates next to the input variables, then relying on Remark 4. We use this simple observation and Theorem 2 to establish Theorem 1.

Theorem 1. *For any $k(n) \leq n^{1/5}$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size $n^{\Omega(k^{1/d}/d)}$. Furthermore, for any $k(n) \leq n$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size $n^{\Omega(k^{1/5d}/d)}$.*

Proof. We assume that $d < 2\log k / \log\log k$ and $(k/2)^{1/d} \geq 2$ (observe that the claimed bound is trivial if $d \geq 2\log k / \log\log k$ or $(k/2)^{1/d} < 2$). Let

$$u_0 = \left\lfloor (k/2)^{1/d} \right\rfloor .$$

Then we have $u_0 \geq 2$ and $u_0 = \Omega(k^{1/d})$. For convenience, let

$$k_0 \overset{\text{def}}{=} u_0^d \leq k/2 \quad \text{and} \quad n' \overset{\text{def}}{=} \left\lfloor \frac{n}{2} \right\rfloor . \tag{1}$$

Further, let w_0 be the largest positive integer such that

$$(u_0 w_0)^d w_0^{33/100} \leq n'. \tag{2}$$

Observe that, since $k \leq n^{1/5}$ and $d < 2\log k / \log\log k$, as $n \to +\infty$ we have similarly $w_0 \to +\infty$. Our choice of w_0 also implies that w_0 satisfies

$$u_0^d (w_0 + 1)^{d+33/100} > n'.$$

Let $n_0 \overset{\text{def}}{=} (u_0 w_0)^d w_0^{33/100}$. Then from $d < 2\log k / \log\log k$ and $w_0 \to +\infty$ we have

$$n_0 \geq u_0^d \left(\frac{w_0 + 1}{2} \right)^{d + \frac{33}{100}} = \Omega(n/2^d) = \omega(n^{0.99}).$$

Combining this with $k \leq n^{1/5}$ and $k_0 \leq k/2$ we have that $k_0 = o(n_0^{20/99})$ and $n_0 \geq k_0^{4.9}$ when n is sufficiently large.

We define a variant of our $\mathsf{SkewedSipser}_{u,d}$ formula so we can rely on Remark 4 and work directly with simple graphs instead of multigraphs. More precisely, let $\mathsf{SkewedSipser}^\dagger_{u_0,w_0,d}$ be analogous to $\mathsf{SkewedSipser}_{u,d}$ with parameters u_0 (AND gate fan-in), w_0 (OR gate fan-in), and d but containing an extra layer of fan-in 2 AND gates at the bottom connected to a new set of input variables. In other words, this is a depth $2d + 2$ read-once alternating formula with twice the number of input variables of our original $\mathsf{SkewedSipser}$ formula (each input variable of $\mathsf{SkewedSipser}$ becomes an AND gate connected to two new fresh variables). Since $\mathsf{SkewedSipser}_{u_0,d}$ can be obtained by restricting $\mathsf{SkewedSipser}^\dagger_{u_0,w_0,d}$ appropriately (i.e. by setting to 1 a single variable in every new pair of variables) a lower bound on the circuit complexity of $\mathsf{SkewedSipser}$ immediately implies the same lower bound for $\mathsf{SkewedSipser}^\dagger$.

In order to obtain a lower bound via Theorem 2, we need that $w_0^{33/100} \geq u_0$. This is equivalent to having $n_0 \geq u_0^{133d/33+1}$, which follows from $d \geq 2$ (we may assume $d \geq 2$ since no depth-1 circuit, i.e. single AND or OR gate, can compute $STCONN(k(n))$) and $n_0 \geq k_0^{4.9}$ since

$$n_0 \geq k_0^{4.9} > k_0^{133/33+1/2.}$$

Consequently, we can apply Theorem 2 to $\mathsf{SkewedSipser}_{u_0,d}$, and it follows from our discussion above that any depth-d circuit computing $\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d}$ must have size at least

$$n_0^{\Omega(u_0/d)} = n^{\Omega(k^{1/d}/d)}. \tag{3}$$

In the rest of the proof we translate (3) into a lower bound for $STCONN(k(n))$. Following the explanation given above, we consider the simple graph $G(\mathsf{SkewedSipser}^{\dagger})$ with appropriate parameters. Since $u_0^d \leq k/2$, it follows from the same argument used to establish Corollary 3.1 that the graph $G(\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d}, z)$ contains an s-to-t path of length at most $2u_0^d \leq k$ if and only if we have $\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d}(z) = 1$. Because $G(\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d})$ has no isolated vertices and has n_0 edges, it contains at most $2n_0 \leq 2n' \leq n$ vertices by (1) and (2). Thus, a circuit $\mathcal{C}$ that computes $STCONN(k(n))$ on undirected simple graphs on n vertices can also be used to compute the formula $\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d}$, and (3) yields that $\mathcal{C}$ must have size $n^{\Omega(k^{1/d}/d)}$. This completes the first part of Theorem 1.

It remains to prove the lower bound for $STCONN(k'(n))$ with $n^{1/5} < k'(n) \leq n$. For this, let $k(n) \overset{\text{def}}{=} n^{1/5}$. We have established above that computing $STCONN(k(n))$ on subgraphs of $G(\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d})$ using depth-d circuits requires size $n^{\Omega(k^{1/d}/d)}$. However, a subgraph of $G(\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d})$ contains an s-to-t path of length at most $k(n)$ if and only if it contains a path from s to t of length at most $k'(n)$ (Observation 5). Consequently, any circuit $\mathcal{C}$ that computes $STCONN(k'(n))$ on general n-vertex graphs can be used to compute $STCONN(k(n))$ on subgraphs of $G(\mathsf{SkewedSipser}^{\dagger}_{u_0,w_0,d})$ (by setting some input edges to 0). In particular, $\mathcal{C}$ must have size

$$n^{\Omega(k^{1/d}/d)} = n^{\Omega(n^{1/5d}/d)} = n^{\Omega(k'^{1/5d}/d)}.$$

This completes the second part of Theorem 1. $\qquad\square$

Remark 8. It is not hard to see that our reduction in fact also captures other natural graph problems such as directed k-path ("Is there a directed path of length $k(n)$ in G?") and directed k-cycle ("Is there a directed cycle of length $k(n)$ in G?"), and hence the lower bounds of Theorem 1 apply to these problems as well. This suggests the possibility of similarly obtaining other lower bounds from (variants of) depth hierarchy theorems for Boolean circuits, and we leave this as an avenue for further investigation.

4 The Random Projection

In this section we define our random projections, which will be crucial in the proof of Theorem 2. First, we introduce notation to manipulate the first two layers of $\mathsf{SkewedSipser}_{u,d}$.

Definition 9. For $2 \leq u \leq w$, we define $\mathsf{CNFSipser}_u$ to be the Boolean function computed by the following monotone read-once formula:

- The top gate is an AND gate and the bottom-layer gates are OR gates.

- The top AND gate has fanin u.

- The bottom-layer OR gates all have fan-in $w^{33/100}$.

For $\mathsf{SkewedSipser}_{u,d}$ and each $\ell \in [d+1]$, we write $\mathrm{OR}^{(\ell)}$ to denote an OR gate that is in the ℓ-th level of OR gates away from the input variables and similarly write $\mathrm{AND}^{(\ell)}$ to denote an AND gate that is in the ℓ-th level of AND gates away from the input variables. So the root of $\mathsf{SkewedSipser}_{u,d}$ is the only $\mathrm{OR}^{(d+1)}$ gate; each $\mathrm{AND}^{(\ell)}$ gate has u many $\mathrm{OR}^{(\ell)}$ gates as its inputs; each $\mathrm{AND}^{(1)}$ gate of $\mathsf{SkewedSipser}_{u,d}$ computes a disjoint copy of $\mathsf{CNFSipser}_u$.

Next we introduce an addressing scheme for gates and variables of $\mathsf{SkewedSipser}_{u,d}$.

Addressing scheme. Viewing $\mathsf{SkewedSipser}_{u,d}$ as a tree (with its leaves being variables and the rest being AND, OR gates), we index its nodes (gates or variables) by addresses as follows. The root (gate) is indexed by ε, the empty string. The j-th child of a node is indexed by the address of its parent concatenated with j. Thus, the variables of $\mathsf{SkewedSipser}_{u,d}$ are indexed by addresses

$$A(d) := \left\{ (b_0, a_1, b_1, \ldots, a_d, b_d) \colon a_i \in [u], b_0, \ldots, b_{d-1} \in [w], b_d \in [w^{33/100}] \right\}.$$

Block and section decompositions. We will refer to the set of $uw^{33/100}$ addresses of variables below an $\mathrm{AND}^{(1)}$ gate as a *block*, and the set of $w^{33/100}$ addresses of variables below an $\mathrm{OR}^{(1)}$ gate as a *section*.

It will be convenient for us to view the set of all variable addresses $A(d)$ as

$$A(d) = B(d) \times A', \text{ where}$$
$$B(d) = \left\{ (b_0, a_1, b_1 \ldots, a_{d-1}, b_{d-1}) \colon a_i \in [u], b_i \in [w] \right\} \quad \text{and} \quad A' = [u] \times [w^{33/100}].$$

Here $B(d)$ can be viewed as the set of addresses of the $\mathrm{AND}^{(1)}$ gates of $\mathsf{SkewedSipser}_{u,d}$, and A' can be viewed as the set of variable addresses of $\mathsf{CNFSipser}_u$ computed by each such gate (following the same addressing scheme).

More formally, for a fixed $\beta \in B(d)$ we call the set of addresses

$$A(d, \beta) \stackrel{\text{def}}{=} \left\{ (\beta, \tau) \colon \tau \in A' \right\}$$

a *block* of $A(d)$; these are the addresses of variables below the $\mathrm{AND}^{(1)}$ gate specified by β. Thus, $A(d)$ is the disjoint union of $w(uw)^{d-1}$ many blocks, each of cardinality $|A'| = uw^{33/100}$.

For a fixed $\beta \in B(d)$ and $a \in [u]$, we call the set of addresses

$$A(d, \beta, a) \stackrel{\text{def}}{=} \left\{ (\beta, a, b) \colon b \in [w^{33/100}] \right\}$$

a *section* of $A(d)$; these are the addresses of variables below the $\mathrm{OR}^{(1)}$ gate specified by (β, a). Each block $A(d, \beta)$ is the disjoint union of u many sections, each of cardinality $w^{33/100}$.

To summarize, the set of addresses of variables $A(d)$ can be decomposed into $w(uw)^{d-1}$ many blocks $A(d, \beta)$ (corresponding to the $\mathrm{AND}^{(1)}$ gates), $\beta \in B(d)$, and each such block can be further decomposed into u many sections $A(d, \beta, a)$ (corresponding to its u input $\mathrm{OR}^{(1)}$ gates), $a \in [u]$.

Accordingly we also decompose A', the set of variable addresses of $\mathsf{CNFSipser}_u$, into sections

$$A'(a) \overset{\text{def}}{=} \{(a,b)\colon b \in [w^{33/100}]\}, \quad \text{for } a \in [u].$$

The following fact is trivial given the definition of $\mathsf{CNFSipser}_u$. (Below and subsequently, we use "ϱ" to denote a restriction to the variables of $\mathsf{CNFSipser}$ and "ρ" to denote a restriction to the variables of $\mathsf{SkewedSipser}$.)

Fact 4.1. *For any $a \in [u]$ and restriction $\varrho \in \{0,1,*\}^{A'}$ that sets all variables in the a-th section $A'(a)$ to 0, i.e., $\varrho_\tau = 0$ for all $\tau \in A'(a)$, we have that $\mathsf{CNFSipser}_u \upharpoonright \varrho \equiv 0$.*

Now we define our random projection operator $\text{proj}_{\boldsymbol{\rho}}(\cdot)$.

Definition 10 (Projection operators). Given a restriction $\rho \in \{0,1,*\}^{A(d)}$, the projection operator proj_ρ maps a function $f\colon \{0,1\}^{A(d)} \to \{0,1\}$ to a function $\text{proj}_\rho(f)\colon \{0,1\}^{B(d)} \to \{0,1\}$, where

$$\big(\text{proj}_\rho(f)\big)(y) = f(x), \quad \text{where } x_{\beta,\tau} \overset{\text{def}}{=} \begin{cases} y_\beta & \text{if } \rho_{\beta,\tau} = * \\ \rho_{\beta,\tau} & \text{if } \rho_{\beta,\tau} \in \{0,1\}. \end{cases}$$

For convenience, we sometimes write $\text{proj}(f \upharpoonright \rho)$ instead of $\text{proj}_\rho(f)$.

Remark 11. The following interpretation of the projection operator will come in handy. Given a restriction $\rho \in \{0,1,*\}^{A(d)}$, if f is computed by a circuit C, then $\text{proj}_\rho(f)$ is computed by a circuit C' obtained from C by replacing every occurrence of $x_{\beta,\tau}$ by y_β if $\rho_{\beta,\tau} = *$, or by $\rho_{\beta,\tau}$ if $\rho_{\beta,\tau} \in \{0,1\}$.

The crux of our *random* projection operator $\text{proj}_{\boldsymbol{\rho}}(\cdot)$ is then a distribution $\mathcal{D}_u^{(d)}$ over restrictions $\{0,1,*\}^{A(d)}$ to the variables $\{x_{\beta,\tau} : (\beta,\tau) \in A(d)\}$, from which $\boldsymbol{\rho}$ is drawn. To this end, we consider the block decomposition $B(d) \times A'$ of $A(d)$, and $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d)}$ is obtained by drawing independently, for each block $\beta \in B(d)$, a restriction $\boldsymbol{\rho}_\beta$ from a distribution $\mathcal{D}_u$ over $\{0,1,*\}^{A'}$ to be defined below.

Definition 12 (Distributions $\mathcal{D}_u$ and $\mathcal{D}_u^{(d)}$). The distribution $\mathcal{D}_u = \mathcal{D}_u(q)$ over $\{0,1,*\}^{A'}$ is parameterized by a probability $q \in (0,1)$. A draw of a restriction $\boldsymbol{\varrho}$ from $\mathcal{D}_u$ is generated as follows:

- With probability q, output $\boldsymbol{\varrho} = \{*\}^{A'}$ (i.e. the restriction fixes no variables).

- Otherwise (with probability $1-q$), we draw $\boldsymbol{a} \leftarrow [u]$ (a random section) and $\boldsymbol{z} \leftarrow \{0,1\}$ (a random bit) independently and uniformly at random, and output $\boldsymbol{\varrho}$ where for each $\tau \in A'$,

$$\boldsymbol{\varrho}_\tau = \begin{cases} \boldsymbol{z} & \text{if } \tau \in A'(\boldsymbol{a}) \\ 1 - \boldsymbol{z} & \text{otherwise.} \end{cases}$$

  Note that in this case $\boldsymbol{\varrho}$ is distributed uniformly among $2u$ many binary strings in $\{0,1\}^{A'}$. These strings are "section-monochromatic", with $u-1$ of the sections taking on entirely the same value $1 - \boldsymbol{z}$ and the one remaining section $\boldsymbol{a}$ taking entirely the other "rare" value $\boldsymbol{z}$.

As described above, a draw of $\boldsymbol{\rho} \in \{0,1,*\}^{B(d) \times A'}$ from $\mathcal{D}_u^{(d)} = \mathcal{D}_u^{(d)}(q)$ is obtained by independently drawing $\boldsymbol{\rho}_\beta \leftarrow \mathcal{D}_u = \mathcal{D}_u(q)$ for each block $\beta \in B(d)$.

The following observation about $\text{supp}(\mathcal{D}_u^{(d)})$ will be useful for us:

14

Remark 13. A restriction $\rho \in \{0, 1, *\}^{B(d) \times A'}$ is in the support of $\mathcal{D}_u^{(d)}$ iff for every block $\beta \in B(d)$, ρ_β is either $\{*\}^{A'}$, or there exists exactly one section $a \in [u]$ such that $\rho_{\beta,\tau} = 0$ if $\tau \in A'(a)$ and 1 otherwise, or there exists exactly one section $a \in [u]$ such that $\rho_{\beta,\tau} = 1$ if $\tau \in A'(a)$ and 0 otherwise.

Therefore, if T is a term of width at most $u - 1$ such that for all blocks $\beta \in B(d)$, the variables from block β that occur in T all occur with the same sign, then T can be satisfied by a restriction in the support of $\mathcal{D}_u^{(d)}$ (i.e., $T \restriction \rho \equiv 1$ for some $\rho \in \mathrm{supp}(\mathcal{D}_u^{(d)})$). (Note that this crucially uses the fact that T has width at most $u - 1$, and in particular does not contain variables from all u sections of any block β. Also note that the inverse of this is not true, e.g., consider $T = x_{\beta,\tau} \wedge \neg x_{\beta,\tau'}$ with τ and τ' from two different sections.)

5   Projection Switching Lemma

Our goal now is to prove the following projection switching lemma for (very) small width DNFs:

Theorem 14 (Projection Switching Lemma). *For $2 \le u \le w$, let F be an r-DNF over the variables $\{x_{\beta,\tau}\}$, $(\beta, \tau) \in A(d)$, where $r \le u - 1$. Then for all $s \ge 1$ and $q \in (0, 1)$, we have*

$$\Pr_{\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d)}(q)} \left[\mathrm{proj}_{\boldsymbol{\rho}}(F) \text{ has decision tree depth} \ge s \right] \le \left(\frac{8qru}{1 - q} \right)^s.$$

Notice that while F is an r-DNF over formal variables $\{x_{\beta,\tau} : (\beta, \tau) \in A(d)\}$, we will bound the decision tree depth of $\mathrm{proj}_{\boldsymbol{\rho}}(F)$, a function over the new formal variables $\{y_\beta : \beta \in B(d)\}$.

Remark 15. Projections will play a key role in the proof. Consider a term of the form $T = x_{\beta,\tau} \wedge \neg x_{\beta,\tau'}$ for some $\tau \neq \tau'$, and suppose our ρ from $\mathcal{D}_u^{(d)}$ is such that $\rho_{\beta,\tau} = \rho_{\beta,\tau'} = *$. In this case we have $T \restriction \rho = x_{\beta,\tau} \wedge \neg x_{\beta,\tau'}$, i.e., the term survives the restriction ρ, but $\mathrm{proj}_\rho(T) = y_\beta \wedge \neg y_\beta \equiv 0$, i.e., the term is killed by proj_ρ. Our proof will crucially leverage simplifications of this sort.

Remark 16. The parameters of Theorem 14 are quite delicate in the sense that the statement fails to hold for DNFs of width u. To see this, consider $\mathsf{SkewedSipser}_{u,d}$ with $d = 1$, a depth-3 formula that can also be written as a u-DNF. Then by Corollary 6.2 (to be introduced in Section 6), we have that for $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(1)}(q)$ with $q = w^{-669/1000}$, the function $\mathrm{proj}_{\boldsymbol{\rho}}(\mathsf{SkewedSipser}_{u,1})$ contains a $w^{33/100}$-way OR as a subfunction — and hence has decision tree depth at least $w^{33/100}$ — with probability $1 - o(1)$. So while the statement of Theorem 14 holds for $(u - 1)$-DNFs, it does not hold for u-DNFs when $u = o(w^{669/2000})$ and $w \to \infty$.

Remark 17. We observe that the conclusion of Theorem 14 still holds if the condition "F is an r-CNF" replaces "F is an r-DNF." This can be shown either by a straightforward adaptation of our proof, or via a reduction to the DNF case using duality, the invariance of our distribution of random projections under the operation of flipping each bit, and the fact that decision tree depth does not change when input variables and output value are negated.

5.1   Canonical decision tree

Given an r-DNF F over variables $\{x_{\beta,\tau} : (\beta, \tau) \in A(d)\}$ and a restriction $\rho \in \{0, 1, *\}^{A(d)}$, $\mathrm{proj}_\rho(F)$ is a function over the new variables $\{y_\beta : \beta \in B(d)\}$. We assume a fixed but arbitrary ordering on the terms in F, and the variables within terms. The canonical decision tree $\mathsf{CanonicalDT}(F, \rho)$ that computes $\mathrm{proj}_\rho(F)$ is defined inductively as follows.

$\mathsf{CanonicalDT}(F, \rho)$:

0. If $\mathrm{proj}_\rho(F) \equiv 0$ or 1, output 0 or 1, respectively.

1. Otherwise, let T be the first term in F such that $T \upharpoonright \rho$ is non-constant and $T \upharpoonright \rho\rho' \equiv 1$ for some $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$. We observe that such a term must exist, or the procedure would have halted at step 0 above and not reached the current step 1.

   To see this, first note that certainly there must exist a term T' such that $T' \upharpoonright \rho$ is non-constant since otherwise $F \upharpoonright \rho$ is constant (and likewise $\mathrm{proj}_\rho(F)$). We furthermore claim that among these terms T', there must exist one such that $T' \upharpoonright \rho$ is satisfiable by some $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$, i.e. $T' \upharpoonright \rho\rho' \equiv 1$. To prove this, suppose that each of these terms T' satisfies that $T' \upharpoonright \rho$ is non-constant and there exists no restriction $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$ such that $T' \upharpoonright \rho\rho' \equiv 1$. By Remark 13 (and our assumption that $r \leq u - 1$), $T' \upharpoonright \rho$ must contain two literals from the same block occurring with opposite signs, i.e., $x_{\beta,\tau}$ and $\neg x_{\beta,\tau'}$, for some $\beta \in B(d)$. In this case, we have that $\mathrm{proj}_\rho(T')$ contains both y_β and $\neg y_\beta$ and hence $\mathrm{proj}_\rho(T') \equiv 0$. But if each such term T' has $\mathrm{proj}_\rho(T') \equiv 0$, then $\mathrm{proj}_\rho(F) \equiv 0$ and the procedure would have halted at step (0).

2. Define
$$\eta = \big\{ \beta \in B(d) : x_{\beta,\tau} \text{ or } \neg x_{\beta,\tau} \text{ occurs in } T \upharpoonright \rho \text{ for some } \tau \big\}$$
   Our canonical decision tree will then query variables y_β, $\beta \in \eta$ exhaustively, i.e., we grow a complete binary tree of depth $|\eta|$; we will refer to T as *the term* of this tree.

3. For every assignment $\pi \in \{0,1\}^\eta$ to variables y_β, $\beta \in \eta$ (equivalently, every path through the complete binary tree of depth $|\eta|$), we recurse on $\mathsf{CanonicalDT}(F, \rho(\eta \mapsto \pi))$, where we use $(\eta \mapsto \pi) \in \{0,1,*\}^{A(d)}$ to denote the following restriction:
$$(\eta \mapsto \pi)_{\beta,\tau} = \begin{cases} \pi_\beta & \text{if } \beta \in \eta \\ * & \text{otherwise,} \end{cases} \quad \text{for all } \beta \in B(d) \text{ and } \tau \in A'. \qquad (4)$$

Proposition 5.1. *For every $\rho \in \{0,1,*\}^{A(d)}$, we have that $\mathsf{CanonicalDT}(F, \rho)$ computes $\mathrm{proj}_\rho(F)$.*

While $\mathsf{CanonicalDT}$ is well defined for all ρ, we shall mostly be interested in $\rho \in \mathrm{supp}(\mathcal{D}_u^{(d)})$.

5.2 Proof of Theorem 14

Let
$$\mathcal{B} \overset{\mathrm{def}}{=} \big\{ \rho \in \mathrm{supp}(\mathcal{D}_u^{(d)}) : \text{decision tree depth of } \mathsf{CanonicalDT}(F, \rho) \geq s \big\}$$
be the set of bad restrictions, To prove Theorem 14, it suffices to bound $\mathbf{Pr}_{\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d)}(q)}[\boldsymbol{\rho} \in \mathcal{B}]$, the total weight of $\mathcal{B}$ under $\mathcal{D}_u^{(d)}(q)$. Following Razborov's strategy (see [Bea95] for more details), we will construct a map
$$\theta : \mathcal{B} \to \{0,1,*\}^{A(d)} \times \{0,1\}^s \times \{0,1\}^{s(\log r + 1)},$$
with the following two key properties:

1. **(injection)** $\theta(\rho) \neq \theta(\rho')$ for any two distinct restrictions $\rho, \rho' \in \mathcal{B}$;

16

2. **(weight increase)** Let $\theta_1(\rho) \in \{0, 1, *\}^{A(d)}$ denote the first component of $\theta(\rho)$. Then

$$\frac{\mathbf{Pr}[\boldsymbol{\rho} = \theta_1(\rho)]}{\mathbf{Pr}[\boldsymbol{\rho} = \rho]} \geq \Gamma, \quad \text{for all } \rho \in \mathcal{B}, \tag{5}$$

where $\Gamma = ((1-q)/2qu)^s$ is "large".

Assuming such a map θ exists (below we describe its construction and prove the two properties stated above), Theorem 14 follows from a simple combinatorial argument.

Proof of Theorem 14. Fix a pair $O \in \{0, 1\}^s \times \{0, 1\}^{s(1+\log r)}$ and let

$$\mathcal{B}_O = \{\rho \in \mathcal{B} : (\theta_2(\rho), \theta_3(\rho)) = O\} \subseteq \mathcal{B},$$

where we use $\theta_2(\rho)$ and $\theta_3(\rho)$ to denote the second and third components of $\theta(\rho)$, respectively.

Then we have that

$$\mathbf{Pr}[\boldsymbol{\rho} \in \mathcal{B}_O] = \sum_{\rho \in \mathcal{B}_O} \mathbf{Pr}[\boldsymbol{\rho} = \rho] \leq (1/\Gamma) \cdot \sum_{\rho \in \mathcal{B}_O} \mathbf{Pr}[\boldsymbol{\rho} = \theta_1(\rho)] \leq 1/\Gamma.$$

Here the first inequality uses (5) and the second inequality uses the property of θ being an injection: we have that $\theta_1(\rho) \neq \theta_1(\rho')$ for any two distinct $\rho, \rho' \in \mathcal{B}_O$ (recall that $\theta_2(\rho) = \theta_2(\rho')$ and $\theta_3(\rho) = \theta_3(\rho')$), and therefore $\sum_{\rho \in \mathcal{B}_O} \mathbf{Pr}[\boldsymbol{\rho} = \theta(\rho)_1] \leq 1$. Summing up over all possible O's, we have

$$\mathbf{Pr}[\boldsymbol{\rho} \in \mathcal{B}] = \sum_O \mathbf{Pr}[\boldsymbol{\rho} \in \mathcal{B}_O] \leq 2^s \cdot (2r)^s \cdot (2qu/(1-q))^s = (8qru/(1-q))^s,$$

and this concludes the proof of Theorem 14. $\qquad\square$

The rest of the section is organized as follows. We construct the map θ in Section 5.3. Then we show that it is an injection in Section 5.4, by showing that one can decode ρ from $\theta(\rho)$ uniquely for any $\rho \in \mathcal{B}$. Finally we prove the weight increase, i.e., (5) in Section 5.5.

5.3   Encoding

Let $\rho \in \mathcal{B}$ be a bad restriction. Let π^* be the lexicographically first path of length at least s in the decision tree $\mathsf{CanonicalDT}(F, \rho)$ (witnessing the badness of ρ), and π be its truncation at length s. Then $\theta_2(\rho)$ is defined to be $\mathrm{binary}(\pi) \in \{0, 1\}^s$, the binary representation of π, i.e., $\pi_i \in \{0, 1\}$ is the evaluation of the ith y-variable along π.

Recall that $\mathsf{CanonicalDT}(F, \rho)$ is composed of a collection of complete binary trees, one for each recursive call of $\mathsf{CanonicalDT}$. Let $R_1, \ldots, R_{s'}$ for some $1 \leq s' \leq s$ denote the sequence of complete binary trees that π visits, with R_1 sharing the same root as $\mathsf{CanonicalDT}(F, \rho)$ and π ending in $R_{s'}$. (Here $s' \geq 1$ because $s \geq 1$.) We also use T_i to denote the term of tree R_i, for each $i \in [s']$.

For each $i \in [s' - 1]$, we let

$$\eta_i = \{\beta \in B(d) : y_\beta \text{ is queried in tree } R_i\},$$

and for the special case of $i = s'$, we let

$$\eta_{s'} = \{\beta \in B(d) : y_\beta \text{ is queried in tree } R_{s'} \text{ before the end of } \pi\}. \tag{6}$$

For each $i \in [s']$, π induces a binary string $\pi^{(i)} \in \{0,1\}^{\eta_i}$, where $\pi_\beta^{(i)}$ for each $\beta \in \eta_i$ is set to be the evaluation of y_β along π (in tree R_i). Note that T_i is the i-th term processed by $\mathsf{CanonicalDT}(F, \rho)$ along the bad path π and equivalently, T_i is the first term processed by

$$\mathsf{CanonicalDT}\big(F, \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)})\big),$$

where $(\eta_j \mapsto \pi^{(j)})$ is a restriction defined as in (4). So T_i is the first term in F such that

$$T_i \upharpoonright \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)})$$

is non-constant and

$$T_i \upharpoonright \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)}) \rho' \equiv 1, \quad \text{for some } \rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)}).$$

At a high level, $\theta_1(\rho)$ and $\theta_3(\rho)$ are defined as follows. The third component

$$\theta_3(\rho) = \mathrm{encode}(\eta_1) \circ \cdots \circ \mathrm{encode}(\eta_{s'}) \in \{0,1\}^{s(1+\log r)}$$

is the concatenation of s' binary strings, where each $\mathrm{encode}(\eta_i)$ is a *concise* representation of η_i. In particular, we are able to recover η_i given both $\mathrm{encode}(\eta_i)$ and T_i. We describe the encoding of η_i in Section 5.3.1. For the first component we have

$$\theta_1(\rho) = \rho\sigma^{(1)} \cdots \sigma^{(s')} \in \{0,1,*\}^{A(d)},$$

where each $\sigma^{(i)} \in \{0,1,*\}^{A(d)}$ is a restriction and $\rho\sigma^{(1)} \cdots \sigma^{(s')}$ is their composition (note that each of these $s'+1$ restrictions, like the overall composition, belongs to $\{0,1,*\}^{A(d)}$). We define the $\sigma^{(i)}$'s in Section 5.3.2.

5.3.1 Encoding η_i

Fix an $i \in [s']$. Let $\eta_i = \{\beta_1, \ldots, \beta_t\}$ for some $t \geq 1$, with β_j's ordered lexicographically. It follows from the definition of η_i that every β_j appears in T_i, meaning that either $x_{\beta,\tau}$ or $\neg x_{\beta,\tau}$ appears in T_i for some $\tau \in A'$.

Instead of encoding each β_j directly using its binary representation, we use $\log r$ bits to encode the index of the first $x_{\beta,\cdot}$ or $\neg x_{\beta,\cdot}$ variable that occurs in T_i. Here $\log r$ bits suffice because T_i has at most r variables. Also recall that we fixed an ordering on the variables of each term, so indices of variables in T_i are well defined. We let $\mathrm{location}(\beta_j)$ denote the $\log r$ bits for β_j. We also append it with one additional bit to indicate whether β_j is the last element in η_i. More formally, we write

$$\mathrm{encode}(\eta_i) = \mathrm{location}(\beta_1) \circ 0 \circ \mathrm{location}(\beta_2) \circ 0 \circ \cdots \circ \mathrm{location}(\beta_t) \circ 1 \in \{0,1\}^{|\eta_i|(1+\log r)}.$$

We summarize properties of $\theta_3(\rho)$ below:

Proposition 5.2. *Given $\theta_3(\rho)$, one can recover uniquely s' and $\mathrm{encode}(\eta_1), \ldots, \mathrm{encode}(\eta_{s'})$. Furthermore, given $\mathrm{encode}(\eta_i)$ and T_i for some $i \in [s']$, one can recover uniquely η_i.*

5.3.2 The $\sigma^{(i)}$ restriction

We now define $\sigma^{(i)}$ for a general $i \in [s']$. For ease of notation we define the restriction

$$\rho^{(i-1)} \stackrel{\text{def}}{=} \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)}) \in \{0, 1, *\}^{A(d)}.$$

Note that $\rho^{(0)} = \rho$. Recalling our CanonicalDT algorithm and the definition of T_i as the i-th term processed by CanonicalDT(F, ρ), we have that T_i is the first term in F such that $T_i \upharpoonright \rho^{(i-1)}$ is non-constant and $T_i \upharpoonright \rho^{(i-1)} \rho' \equiv 1$ for some $\rho' \in \text{supp}(\mathcal{D}_u^{(d)})$. Therefore, we have

$$\eta_i = \{\beta \in B(d): x_{\beta,\tau} \text{ or } \neg x_{\beta,\tau} \text{ occurs in } T_i \upharpoonright \rho^{(i-1)} \text{ for some } \tau \in A'\}.$$

We define $\sigma^{(i)} \in \{0, 1, *\}^{B(d) \times A'}$ to be an arbitrary restriction (say the lexicographic first under the ordering $0 \prec 1 \prec *$) satisfying the following three properties:

1. $T_i \upharpoonright \rho^{(i-1)} \sigma^{(i)} \not\equiv 0$, and

2. $\sigma^{(i)} \in \text{supp}(\mathcal{D}_u^{(d)})$, and

3. $\sigma_\beta^{(i)} \in \{0, 1\}^{A'}$ for all $\beta \in \eta_i$, and $\sigma_\beta^{(i)} = \{*\}^{A'}$ for all $\beta \notin \eta_i$.

In words, $\sigma^{(i)}$ is the lexicographic first restriction in $\text{supp}(\mathcal{D}_u^{(d)})$ that completely fixes blocks $\beta \in \eta_i$, leaves all other blocks $\beta \notin \eta_i$ free, and fixes the blocks in η_i in a way that does not falsify $T_i \upharpoonright \rho^{(i-1)}$. For $1 \le i < s'$, we recall that η_i contains all blocks with variables occurring in $T_i \upharpoonright \rho^{(i-1)}$, and so property (1) above can in fact be stated as $T_i \upharpoonright \rho^{(i-1)} \sigma^{(i)} \equiv 1$. (This is not necessarily true for the special case of $i = s'$ since $\eta_{s'}$ may only contain a subset of the blocks with variables occurring in $T_{s'} \upharpoonright \rho^{(s'-1)}$; c.f. (6).)

We observe that such a restriction $\sigma^{(i)}$ (one satisfying all three properties above) must exist. As remarked at the start of this subsection, by the definition of T_i there exists a restriction $\rho' \in \text{supp}(\mathcal{D}_u^{(d)})$ such that $T_i \upharpoonright \rho^{(i-1)} \rho' \equiv 1$. This along with the fact that $\mathcal{D}_u^{(d)}$ is independent across blocks implies the existence of a restriction in $\text{supp}(\mathcal{D}_u^{(d)})$ that fixes exactly the blocks in η_i in a way that does not falsify $T_i \upharpoonright \rho^{(i-1)}$.

This finishes the definition of $\sigma^{(i)}$. We record the following key properties of $\sigma^{(i)}$:

Proposition 5.3. $T_i \upharpoonright \rho^{(i-1)} \sigma^{(i)} \equiv 1$ for $1 \le i < s'$, and $T_{s'} \upharpoonright \rho^{(s'-1)} \sigma^{(s')} \not\equiv 0$.

Proposition 5.4. For every $\beta \in \eta_i$, we have $\rho_\beta^{(i-1)} = \{*\}^{A'}$ whereas $\sigma_\beta^{(i)} \in \{0, 1\}^{A'}$, and

$$\Pr_{\varrho \leftarrow \mathcal{D}_u(q)}[\varrho = \rho_\beta^{(i-1)}] = q \quad \text{whereas} \quad \Pr_{\varrho \leftarrow \mathcal{D}_u(q)}[\varrho = \sigma_\beta^{(i)}] = \frac{1-q}{2u}.$$

5.4 Decodability

Lemma 5.5. The map $\theta: \mathcal{B} \to \{0, 1, *\}^{A(d)} \times \{0, 1\}^s \times \{0, 1\}^{s(\log r + 1)}$, where

$$\theta(\rho) = \left(\rho \sigma^{(1)} \cdots \sigma^{(s')}, \text{ binary}(\pi), \text{ encode}(\eta_1) \circ \cdots \circ \text{encode}(\eta_{s'})\right), \tag{7}$$

is an injection.

We will prove Lemma 5.5 by describing a decoder that can recover $\rho \in \mathcal{B}$ given $\theta(\rho)$ as in (7). Let $\sigma = \sigma^{(1)} \cdots \sigma^{(s')}$. Note that s' can be derived from $\theta_3(\rho)$. To obtain ρ, it suffices to recover the sets η_i, by simply replacing $(\rho\sigma)_{\beta,\tau}$ with $*$ for all $\beta \in \eta_1 \cup \cdots \cup \eta_{s'}$ and all $\tau \in A'$.

To recover η_i's, we assume inductively that the decoder has recovered the "hybrid" restriction

$$\rho^{(i-1)}\sigma^{(i)} \cdots \sigma^{(s')} = \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)})\sigma^{(i)} \cdots \sigma^{(s')} \text{ and the sets } \eta_1, \ldots, \eta_{i-1}, \quad (8)$$

with the base case $i = 1$ being $\rho\sigma^{(1)} \cdots \sigma^{(s')} = \theta_1(\rho)$, which is trivially true by assumption. We will show below how to decode T_i and η_i, and then obtain the next "hybrid" restriction

$$\rho^{(i)}\sigma^{(i+1)} \cdots \sigma^{(s')} = \rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_i \mapsto \pi^{(i)})\sigma^{(i+1)} \cdots \sigma^{(s')}$$

We can recover all s' sets $\eta_1, \ldots, \eta_{s'}$ after repeating this for s' times.

The following lemma shows how to recover T_i, given the "hybrid" restriction in (8).

Proposition 5.6. *For $1 \le i < s'$, we have that T_i is the first term in F such that*

$$T_i \upharpoonright \rho^{(i-1)}\sigma^{(i)} \cdots \sigma^{(s')} \equiv 1.$$

For the special case of $i = s'$, we have that $T_{s'}$ is the first term in F such that

$$T_{s'} \upharpoonright \rho^{(s'-1)}\sigma^{(s')}\rho'' \equiv 1$$

for some $\rho'' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$.

Proof. We first justify the claim for $1 \le i < s'$. Recall that T_i is the first term in F such that $T_i \upharpoonright \rho^{(i-1)}$ is non-constant and $T_i \upharpoonright \rho^{(i-1)}\rho' \equiv 1$ for some restriction $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$. This together with Proposition 5.3 implies that T_i is the first term in F such that $T_i \upharpoonright \rho^{(i-1)}\sigma^{(i)} \equiv 1$: as $\sigma^{(i)} \in \mathrm{supp}(\mathcal{D}_d^{(u)})$, it follows that $\rho^{(i-1)}\sigma^{(i)}$ cannot satisfy any term that occurs before T_i in F. For the same reason, T_i remains the first term in F such that $T_i \upharpoonright \rho^{(i-1)}\sigma^{(i)} \cdots \sigma^{(s')} \equiv 1$ (since $\sigma^{(i+1)}, \cdots, \sigma^{(s')} \in \mathrm{supp}(\mathcal{D}_u^{(d)})$ and so is their composition).

The argument for $i = s'$ is similar. We again recall that $T_{s'}$ is the first term in F such that $T_{s'} \upharpoonright \rho^{(s'-1)}$ is non-constant and $T_{s'} \upharpoonright \rho^{(s'-1)}\rho' \equiv 1$ for some restriction $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$. Since every term in T that occurs before $T_{s'}$ in F is such that $T \upharpoonright \rho^{(s'-1)}\rho' \not\equiv 1$ for all $\rho' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$, certainly $T \upharpoonright \rho^{(s'-1)}\sigma^{(s')}\rho'' \not\equiv 1$ for all $\rho'' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$ as well. On the other hand, by Proposition 5.3 we have that $\sigma^{(s')}$ does not falsify $T_{s'} \upharpoonright \rho^{(s'-1)}$, and so there must exist $\rho'' \in \mathrm{supp}(\mathcal{D}_u^{(d)})$ such that $T_{s'} \upharpoonright \rho^{(s'-1)}\sigma^{(s')}\rho'' \equiv 1$. This completes the proof. $\square$

With T_i in hand we use $\mathrm{encode}(\eta_i)$ to reconstruct η_i by Proposition 5.2. We modify the current "hybrid" restriction $\rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)})\sigma^{(i)} \cdots \sigma^{(s')}$ as follows: for each $\beta \in \eta_i$, set

$$\left(\rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_{i-1} \mapsto \pi^{(i-1)})\sigma^{(i)} \cdots \sigma^{(s')}\right)_{\beta,\tau} = \pi_\beta^{(i)}, \quad \text{for all } \tau \in A'.$$

The resulting restriction is $\rho(\eta_1 \mapsto \pi^{(1)}) \cdots (\eta_i \mapsto \pi^{(i)})\sigma^{(i+1)} \cdots \sigma^{(s')}$ as desired.

Starting with $\rho\sigma$ and repeating this procedure for s' times, we recover all η_i's and then ρ. This completes the proof that θ is an injection.

5.5 Weight increase

Recall that ρ and $\rho\sigma$ differ in exactly s many blocks, and furthermore, ρ is $\{*\}^{A'}$ on all these blocks whereas $\rho\sigma$ belongs to $\{0,1\}^{A'} \cap \mathrm{supp}(\mathcal{D}_u)$ on these blocks.

Lemma 5.7. *For any $\rho \in \mathcal{B}$ and $\rho\sigma = \theta_1(\rho)$, we have*

$$\frac{\mathbf{Pr}[\boldsymbol{\rho} = \rho\sigma]}{\mathbf{Pr}[\boldsymbol{\rho} = \rho]} = \prod_{\substack{blocks\ \beta\ on \\ which\ they\ differ}} \frac{\mathbf{Pr}[\boldsymbol{\varrho} = (\rho\sigma)_\beta]}{\mathbf{Pr}[\boldsymbol{\varrho} = \rho_\beta]} = \left(\frac{1-q}{2qu}\right)^s.$$

Proof. This follows from independence across blocks and Proposition 5.4. $\qquad\square$

6 Proof of Theorem 2

In this section we prove our main technical result, Theorem 2, restated below:

Theorem 2. *There is an absolute constant $c > 0$ such that the following holds. Let $d = d(w)$ and $u = u(w)$ satisfy $d \geq 1$ and $2 \leq u \leq w^{33/100}$. Then for w sufficiently large, any depth-d circuit computing the $\mathsf{SkewedSipser}_{u,d}$ function (recall that this is a formula of depth $2d+1$ over $n = (uw)^d w^{33/100}$ variables) must have size at least $n^{c \cdot (u/d)}$.*

We begin by first observing that the claimed $n^{\Omega(u/d)}$ circuit size lower bound is $o(n)$, and hence vacuous, if $d > u$; thus it suffices to prove the claimed bound under the assumption that $d \leq u$. We make this assumption in the rest of the proof below (see specifically Corollary 6.2). Of course we can also assume that $d \geq 2$, since depth-1 circuits of any size cannot compute $\mathsf{SkewedSipser}_{u,1}$. In the proof we set the parameter q to be $q = w^{-669/1000}$.

In Section 6.1 we establish that our target function $\mathsf{SkewedSipser}_{u,d}$ retains structure with high probability under a suitable random projection. In Section 6.2 we repeatedly apply both this result and our projection switching lemma to prove Theorem 2.

6.1 Target preservation

We start with an easy proposition about what happens to $\mathsf{CNFSipser}_u$ under a random restriction from $\mathcal{D}_u(q)$. The following is an immediate consequence of Definition 12 and Fact 4.1:

Proposition 6.1. *For $\boldsymbol{\varrho} \leftarrow \mathcal{D}_u(q)$, we have that*

$$\big(\mathsf{CNFSipser}_u \restriction \boldsymbol{\varrho}\big) \equiv \begin{cases} \mathsf{CNFSipser}_u & \text{with probability } q \\ 0 & \text{with probability } 1-q. \end{cases}$$

We obtain the following corollary.

Corollary 6.2. *For every $1 \leq \ell \leq d$, we have that $\mathrm{proj}_{\boldsymbol{\rho}}(\mathsf{SkewedSipser}_{u,\ell})$ contains $\mathsf{SkewedSipser}_{u,\ell-1}$ as a subfunction with probability at least 0.9 over a random restriction $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(\ell)}(q)$.*

Proof. Recall that $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(\ell)}(q)$ is drawn by independently drawing $\boldsymbol{\rho}_\beta \leftarrow \mathcal{D}_u(q)$ for each block $\beta \in B(\ell)$. We have that $\text{proj}_{\boldsymbol{\rho}}(\mathsf{SkewedSipser}_{u,\ell})$ contains $\mathsf{SkewedSipser}_{u,\ell-1}$ as a subfunction if the following holds: for each of the $\mathrm{OR}^{(2)}$ gates in $\mathsf{SkewedSipser}_{u,\ell}$, at least $w^{33/100}$ of the w $\mathrm{AND}^{(1)}$ gates (each one corresponding to an independent $\mathsf{CNFSipser}_u$ function) that are its children (say at addresses $\beta_1, \ldots, \beta_w$) have $\boldsymbol{\rho}_{\beta_i} \in \{*\}^{A'}$.

By Proposition 6.1, for a given $\mathrm{OR}^{(2)}$ gate, the expected number of β_i's beneath it that have $\boldsymbol{\rho}_{\beta_i} \in \{*\}^{A'}$ is $qw = w^{331/1000}$. So a multiplicative Chernoff bound shows that at least $w^{33/100}$ of the β_i's beneath it have $\boldsymbol{\rho}_{\beta_i} \in \{*\}^{A'}$ except with failure probability at most $e^{-w^{331/1000}/8}$. By a union bound over the (at most n) $\mathrm{OR}^{(2)}$ gates in $\mathsf{SkewedSipser}_{u,\ell}$, we have that the overall failure probability is at most $n \cdot e^{-w^{331/1000}/8}$. Since

$$n = u^d w^{d + \frac{33}{100}} \leq w^{\frac{133}{100}d + \frac{33}{100}} \leq w^{\frac{133}{100}u + \frac{33}{100}} \leq w^{\frac{133}{100}w^{33/100} + \frac{33}{100}} \ll 0.1 \cdot e^{w^{331/1000}/8},$$

the proof is complete. (In the above we used $u \leq w^{33/100}$ for the first inequality, $d \leq u$ for the second, $u \leq w^{33/100}$ again for the third, and w being sufficiently large for the last.) $\qquad\square$

6.2 Completing the Proof of Theorem 2

Most of the proof is devoted to showing that the required size for a depth-d circuit that computes $\mathsf{SkewedSipser}_{u,d}$ is at least

$$S \stackrel{\text{def}}{=} \frac{0.1}{(16u^2q)^{u-1}}. \tag{9}$$

We prove (9) by contradiction; so assume there is a depth-d circuit C of size at most S that computes $\mathsf{SkewedSipser}_{u,d}$. As noted in Section 2.2 we assume that C is alternating and leveled.

We "get the argument off the ground" by first hitting both $\mathsf{SkewedSipser}_{u,d}$ and C with $\text{proj}_{\boldsymbol{\rho}}(\cdot)$ for $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d)}(q)$, where $q = w^{-669/1000}$. (By Remark 17, we can apply our projection switching lemma, Theorem 14, both to r-DNFs and r-CNFs.) Applying Theorem 14 (with $r = 1$ and $s = u - 1$) to each of the gates at distance 1 from the inputs in C,[2] we have that the resulting circuit $\text{proj}_{\boldsymbol{\rho}}(C)$ has depth d, bottom fan-in $u - 1$, and at most S gates at distance at least 2 from the inputs[3] with failure probability at most $S \cdot (16qu)^{u-1} < 0.1$. On the other hand, taking $\ell = d$ in Corollary 6.2 we have that $\text{proj}_{\boldsymbol{\rho}}(\mathsf{SkewedSipser}_{u,d})$ contains $\mathsf{SkewedSipser}_{u,d-1}$ as a subfunction with failure probability at most 0.1. By a union bound, with probability at least 0.8, a draw of $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d)}(q)$ satisfies both of the above, and we fix any such restriction $\kappa^{(d)} \in \text{supp}(\mathcal{D}_u^{(d)}(q))$. A further deterministic "trimming" restriction (by only setting certain variables to 0; note that this can only simplify $\text{proj}_{\rho^{(d)}}(C)$ further) causes the target $\text{proj}_{\rho^{(d)}}(\mathsf{SkewedSipser}_{u,d})$ to become exactly $\mathsf{SkewedSipser}_{u,d-1}$. Let us write C_d to denote the resulting simplified version of the original circuit C after the combined "project-and-trim". As C is supposed to compute $\mathsf{SkewedSipser}_{u,d}$, C_d must compute $\mathsf{SkewedSipser}_{u,d-1}$.

Next, we consider what happens to $\mathsf{SkewedSipser}_{u,d-1}$ and C_d if we hit them both with $\text{proj}_{\boldsymbol{\rho}}(\cdot)$ for $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d-1)}(q)$. Applying Theorem 14 (with $r = s = u - 1$) to each of the gates at distance 2 from the inputs and taking a union bound, the resulting circuit $\text{proj}_{\boldsymbol{\rho}}(C_d)$ has depth $(d-1)$, bottom

---

[2]In this initial application we view C as having an extra layer of gates of fan-in 1 next to the input variables, so we have a valid application of Theorem 14 with $r = u - 1 \geq 1$.

[3]Note that $\text{proj}_{\boldsymbol{\rho}}(C)$ may have a large number of gates at distance 1 from the inputs but it suffices for our purpose to bound the number of gates at distance at least 2 from the inputs.

fan-in $u - 1$, and at most S gates at distance at least 2 from the inputs with failure probability at most $S \cdot (16ruq)^{u-1} < S \cdot (16qu^2)^{u-1} \leq 0.1$. On the other hand, taking $\ell = d - 1$ we can again apply Corollary 6.2 to $\mathsf{SkewedSipser}_{u,d-1}$ and we have that $\mathrm{proj}_{\boldsymbol{\rho}}(\mathsf{SkewedSipser}_{u,d-1})$ contains $\mathsf{SkewedSipser}_{u,d-2}$ as a subfunction with failure probability at most 0.1. Once again by a union bound, with probability at least 0.8 a draw of $\boldsymbol{\rho} \leftarrow \mathcal{D}_u^{(d-1)}(q)$ satisfies both of the above, and we fix any such restriction $\kappa^{(d-1)} \in \mathrm{supp}(\mathcal{D}_u^{d-1}(q))$. As before we perform a deterministic trimming restriction that causes the target $\mathrm{proj}_{\kappa^{(d-1)}}(\mathsf{SkewedSipser}_{u,d-1})$ to become exactly $\mathsf{SkewedSipser}_{u,d-2}$ and we let C_{d-1} be the resulting simplified version of C_d after the combined project-and-trim. As C_d computes $\mathsf{SkewedSipser}_{u,d-1}$ we have that C_{d-1} must compute $\mathsf{SkewedSipser}_{u,d-2}$.

Repeating the argument above, each time taking $r = s = u - 1$ in Theorem 14, there exist a sequence of restrictions $\kappa^{(d-2)} \in \mathrm{supp}(\mathcal{D}_u^{(d-2)}(q)), \dots, \kappa^{(1)} \in \mathrm{supp}(\mathcal{D}_u^{(1)}(q))$ and their resulting circuits $C_{d-2}, \dots, C_1$ such that

- **Hard function retains structure.** For $1 \leq \ell \leq d - 2$, $\mathrm{proj}_{\kappa^{(\ell)}}(\mathsf{SkewedSipser}_{u,\ell})$ contains $\mathsf{SkewedSipser}_{u,\ell-1}$ as a subfunction, and hence there exists a deterministic trimming restriction that results in $\mathrm{proj}_{\kappa^{(\ell)}}(\mathsf{SkewedSipser}_{u,\ell})$ becoming exactly $\mathsf{SkewedSipser}_{u,\ell-1}$.

- **Circuit collapses.** For $2 \leq \ell \leq d - 2$, the circuit $\mathrm{proj}_{\kappa^{(\ell)}}(C_{\ell+1})$ has depth ℓ, bottom fan-in $u - 1$, and has at most S gates at distance at least 2 from the inputs. Furthermore, C_ℓ is the simplified version of $\mathrm{proj}_{\kappa^{(\ell)}}(C_{\ell+1})$ after the deterministic trimming restriction associated with $\mathrm{proj}_{\kappa^{(\ell)}}(\mathsf{SkewedSipser}_{u,\ell})$. Finally, the circuit $\mathrm{proj}_{\kappa^{(1)}}(C_2)$ can be expressed as a depth-$(u - 1)$ decision tree, and C_1 is the simplified version of $\mathrm{proj}_{\kappa^{(1)}}(C_2)$ after the deterministic trimming restriction associated with $\mathrm{proj}_{\kappa^{(1)}}(\mathsf{SkewedSipser}_{u,1})$.

The above implies that C_ℓ computes $\mathsf{SkewedSipser}_{u,\ell-1}$ for all $1 \leq \ell \leq d - 2$. This yields the desired contradiction since C_1, a decision tree of depth at most $u-1$, cannot compute $\mathsf{SkewedSipser}_{u,0}$, the OR of $w^{33/100} \geq u$ many variables. Hence any depth-d circuit computing $\mathsf{SkewedSipser}_{u,d}$ must have size at least S, where S is the quantity defined in (9). The following calculation showing that $S = n^{\Omega(u/d)}$ completes the proof of Theorem 2:

Claim 6.3. $S = n^{\Omega(u/d)}$.

Proof. We first observe that

$$n = u^d w^{d + \frac{33}{100}} \leq w^{\frac{133}{100}d + \frac{33}{100}} \leq w^{(\frac{133}{100} + \frac{33}{200})d} < w^{\frac{3d}{2}}, \quad \text{and hence} \quad n^{\frac{2}{3d}} < w,$$

where we used $u \leq w^{33/100}$ for the first inequality and $d \geq 2$ for the second. As a result we have

$$S = 0.1 \left(\frac{w^{669/1000}}{16u^2} \right)^{u-1} \geq 0.1 \left(\frac{w^{9/1000}}{16} \right)^{u/2} \geq 0.1 \left(\frac{n^{\frac{2}{3d} \cdot \frac{9}{1000}}}{16} \right)^{u/2} = n^{\Omega(u/d)},$$

where we used $q = w^{-669/1000}$ for the first equality, $2 \leq u \leq w^{33/100}$ for the first inequality, and $w > n^{\frac{2}{3d}}$ for the final equality. $\qquad \square$

Remark 18. We remark that a straightforward construction yields small-depth circuits computing $\mathsf{SkewedSipser}_{u,d}$ that nearly match the lower bound given by Theorem 2. This construction simply applies de Morgan's law to convert a u-way AND of w-way ORs into a w^u-way OR of u-way ANDs. This is done for all of the $\mathrm{AND}^{(d)}, \mathrm{AND}^{(d-2)}, \mathrm{AND}^{(d-4)}, \dots$ gates in $\mathsf{SkewedSipser}_{u,d}$. Collapsing adjacent layers of gates after this conversion, we obtain a depth-$(d + 1)$ circuit of size $n^{O(u/d)}$ that computes the $\mathsf{SkewedSipser}_{u,d}$ function.

References

[Ajt83]   Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983. 2

[Ajt89]   Miklós Ajtai. First-order definability on finite structures. *Annals of Pure and Applied Logic*, 45:211–225, 1989. 1, 2, 3, 7

[Bea95]   Paul Beame. A switching lemma primer. University of Washington, Dept. of Computer Science and Engineering, Technical Report UW-CSE-95-07-01, 1995. 16

[BIP98]   Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth lower bounds for small distance connectivity. *Computational Complexity*, 7:325 –345, 1998. 1, 2, 3, 6, 7, 8

[BPU92]   Stephen Bellantoni, Toniann Pitassi, and Alasdair Urquhart. Approximation and small depth Frege proofs. *SIAM Journal on Computing*, 21(6):1161–1179, 1992. 1, 2, 3, 7

[Cai86]   Jin-Yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–29, 1986. 2

[FSS84]   Merrick Furst, James Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. 2

[Hås86]   Johan Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, Cambridge, MA, 1986. 1, 2, 3, 4, 8

[IPS97]   Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size–depth tradeoffs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997. 7

[IS01]   Russell Impagliazzo and Nathan Segerlind. Counting axioms do not polynomially simulate counting gates. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 200–209, 2001. 7

[KPPY84]   Maria Klawe, Wolfgang Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC)*, pages 480–487, 1984. 3

[Ros14]   Benjamin Rossman. Formulas vs. circuits for small distance connectivity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 203–212. ACM, 2014. 1, 2, 3, 6, 7, 8

[RST15]   Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for Boolean circuits. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015. to appear. 1, 7

[Sav70]   Walter Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970. 1

[Sip83]   Michael Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 61–69, 1983. 1, 4

[Sta]   Theoretical Computer Science StackExchange. Available at http://cstheory.stackexchange.com/questions/7672/ most-efficient-way-to-convert-an-textac0-circuit-to-a-circuit-of-any-dep. 8

[Wig92]   Avi Wigderson. The complexity of graph connectivity. In *Proceedings of the 17th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 112–132. Springer-Verlag, 1992. 1

[Yao85]   Andrew Yao. Separating the polynomial time hierarchy by oracles. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 1985. 1, 2, 4