Improved Depth Lower Bounds for Small Distance Connectivity

Paul Beame*
Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98195
beame@cs.washington.edu

Russell Impagliazzo†
Computer Science and Engineering
UC, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0114
russell@cs.ucsd.edu

Toniann Pitassi‡
Depts. of Math and Computer Science
University of Pittsburgh
Pittsburgh, PA
toni@cs.pitt.edu

Abstract

We consider the problem of determining, given a graph G and specified nodes s and t, whether or not there is a path of at most k edges in G from s to t. We show that solving this problem on polynomial-size unbounded fan-in circuits, requires depth $\Omega(\log\log k)$, improving on a depth lower bound of $\Omega(\log^* k)$ when $k = \log^{O(1)} n$ given in [2, 8]. In addition we show that there is a constant c such that for $k \leq \log n$, any depth d unbounded fan-in circuit for this problem requires size at least $n^{ck^{\epsilon_d}}$ where $\epsilon_d = \phi^{-2d}/3$ and ϕ is the golden mean. This latter result improves on an $n^{\Omega(\log^{(d+3)} k)}$ bound from [2, 8] where $\log^{(i)}$ is the i-fold composition of log with itself.

The key to our technique is a new form of 'switching lemma' which combines some of the features of iteratively shortening terms due to Furst, Saxe, and Sipser [13] and Ajtai [1] with the kinds of switching lemma arguments introduced by Yao [18], Håstad [14], and Cai [9] that have been the methods of choice for subsequent results.

1 Introduction

Connectivity problems in graphs are among the most fundamental in computer science. In particular, the fact that directed st-connectivity and transitive closure are complete problems for NL, nondeterministic logspace, shows the importance of connectivity from the viewpoint of computational complexity. It also points to connectivity problems in general as good candidates for problems in NP that may be proven to lie outside deterministic logspace, L, or NC^1. As well, good complexity bounds for connectivity problems on bounded or unbounded fan-in circuit models or on deterministic Turing machines would give us a better understanding of the relationships in the following chain of complexity classes

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{SAC}^1 = \mathsf{LOGCFL} \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2.$$

The research on graph connectivity is voluminous and even since Wigderson's excellent survey of the state of the art in 1992 [17], there have been significant new developments in connectivity algorithms [4, 12] and lower bounds on restricted models of computation [11, 3, 10, 19].

The key tool in showing that every problem in NL may be solved with circuits of relatively small depth is the 'Repeated Squaring' or 'Pointer Doubling' algorithm for transitive closure. Another way of phrasing some of these complexity questions is to ask whether or not repeated squaring gives an optimal depth for polynomial-size circuits computing transitive closure or st-connectivity. (The $O(\log^{1.5} n)$ space algorithm of Nisan, Szemeredi, and Wigderson [15] shows that for undirected graphs there is an algorithm that uses better *space* than repeated squaring but this does not yield improved depth polynomial-size circuits or say anything about directed connectivity.)

Consider the problem of distance k connectivity, $STCONN(k(n))$: given an unweighted graph G with n vertices and vertices s and t, determine whether or not G contains a path of length at most $k(n)$ from s to t. (Note that distance-bounded connectivity for undirected graphs is just as hard as distance-bounded connectivity in directed graphs via an easy reduction that converts a directed graph into a layered undirected

*Research support by NSF grant CCR-9303017

†Research Supported by NSF YI Award CCR-92-570979, Sloan Research Fellowship BR-3311, and USA-Israel BSF Grant 92-00043

‡Research supported by NSF YI Award CCR-9457782

graph.) Since one can square a Boolean matrix using a polynomial-size circuit of depth 2 consisting of a layer of bounded fan-in $\wedge$-gates feeding into a single unbounded fan-in $\vee$-gate, by using repeated squaring one can solve $STCONN(k(n))$ using polynomial-size (semi)-unbounded fan-in circuits of depth $2\log k$. This also gives polynomial-size fan-in 2 circuits of depth $O(\log n \log k)$ for the problem.

On unbounded fan-in circuits, $STCONN(k(n))$ was first considered by Ajtai [2] who showed that for any function $k(n)$ tending to infinity, $STCONN(k(n))$ requires superpolynomial size on constant-depth circuits. For $k = \log^{\omega(1)} n$, Håstad's parity lower bound [14] implies that polynomial-size unbounded fan-in circuits for $STCONN(k(n))$ require depth $\Omega(\log k/\log\log n)$ but this says nothing about short distances.

Any improvement on repeated squaring for any distance k would result in an improved algorithm for the general directed st-connectivity problem: Suppose that for some k, we could compute distance k connectivity in depth $T_k = o(\log k)$ on polynomial-size unbounded fan-in circuits. Then, by analogy with repeated squaring we would obtain a general directed st-connectivity algorithm of depth $O(T_k \log n/\log k) = o(\log n)$ which would be very surprising and would improve the general simulations of NL both by unbounded fan-in circuits and by fan-in 2 circuits.

This motivated Wigderson in his graph connectivity survey, after discussing Ajtai's result, to suggest a focus on small distance connectivity as an avenue for beating the bounds given by repeated squaring. The question that we investigate is the extent to which this focus can succeed, at least in the case of unbounded fan-in circuits.

As noted above, Ajtai's bound says that for growing k we cannot ever reduce the depth complexity for efficiently computing $STCONN(k(n))$ to a constant. An explicit computation of this non-constant lower bound and a simplification of the key lemma of [2] due to Bellantoni, Pitassi, and Urquhart [8] shows that Ajtai's technique gave an $\Omega(\log^* k)$ depth lower bound for polynomial-size unbounded fan-in circuits solving $STCONN(k(n))$.

Our main result is a substantially improved lower bound for $STCONN(k(n))$ when $k = \log^{O(1)} n$. Namely we show that for polynomial-size unbounded fan-in circuits, computing $STCONN(k(n))$ requires depth $\Omega(\log\log k)$. In addition we show that there is some constant c such that for $k \le \log n$, any depth d unbounded fan-in circuit for $STCONN(k(n))$ requires size at least $n^{ck^{\epsilon_d}}$ where $\epsilon_d = \phi^{-2d}/3$. This latter result improves on an $n^{\Omega(\log^{(d+3)} k)}$ bound from [2, 8] where $\log^{(i)}$ is the i-fold composition of log with itself.

The key to our technique is a new form of 'switching lemma' which combines some of the features of the "independent-set-style" switching lemma due to Furst, Saxe, and Sipser [13] and Ajtai [1] with the "Håstad-style" switching lemma arguments introduced by Yao [18], Håstad [14], and Cai [9] that have been the methods of choice for subsequent results. The Håstad-style switching lemmas show that if we are given a particular DNF formula, then a random restriction allows us to represent the restricted formula as a small-depth decision tree, with high probability. The method of converting from the restricted DNF formula to the decision tree is a simple deterministic procedure where queries to the variables are made in the order in which they appear in the unset terms of the DNF formula.

In the independent-set-style switching lemma, one argues that if we are given a particular DNF formula, then a random restriction allows us to find a small set of variables such that after applying the restriction and setting this small set of variables, the remaining DNF formula has term size reduced by at least 1. Using this method, the decision tree is built in r stages, where r is the original term size, and at each stage a successive restriction must be applied. The problem with this type of switching lemma is that we must apply r restrictions in order to build a small-depth decision tree, and this leads to barely superpolynomial final bounds. However, this type of switching lemma involves a global reordering of variables in the construction of the decision tree, and seems applicable in more situations.

Our new switching lemma combines the desirable properties of the above two methods. We show that with high probability a random restriction allows us, for *every* assignment to some of the remaining variables, to find a small set of variables such that after setting this small set of variables (plus applying the assignment and the restriction), the remaining DNF formula has term size reduced by at least 1. Thus, the same restriction can be "reused" at each stage of the tree-building process.

A major conceptual tool in developing this new switching lemma is the recent more direct and simpler formulation of Håstad's argument due to Woods (personal communication) and Razborov [16] which is developed further in [5] for a variety of other examples.

2 Definitions

2.1 Layered Graphs of Permutations

The specific family of graphs we consider is the same as the one that Ajtai considered in [2]: Let $\mathcal{G}(n, k)$ be the set of all graphs with the following properties: Each graph G in $\mathcal{G}(n,k)$ has $k+1$ disjoint *layers* of vertices, $V_0, V_1, \dots, V_k$, with each V_i containing n vertices. The only edges in such a graph G will be between adjacent layers, i.e. between V_i to V_{i+1} for $i < k$, and the induced graph on $V_i \cup V_{i+1}$ will be a perfect bipartite matching. Alternatively, one can view these edges as defining a bijection from V_i to V_{i+1}. Thus, the graph as a whole consists of n disjoint paths of length exactly k from layer V_0 to V_k. For simplicity we will call any member of $\mathcal{G}(n,k)$ a *layered graph.*

As with all graphs, we can represent any layered graph by the variables defining its adjacency matrix but given the structure of layered graphs it is convenient to represent only the relevant entries. Thus, we represent members of $\mathcal{G}(n,k)$ using kn^2 Boolean variables $x_{ij}^{k'}$ for $1 \le i, j \le n$ and $0 \le k' < k$ where $x_{ij}^{k'}$ is 1 if and only if there is an edge in the graph connecting the i-th vertex in $V_{k'}$ to the j-th vertex in $V_{k'+1}$. Note also that over the domain of layered graphs we can eliminate all negated variables since $\neg x_{ij}^{k'} \equiv \bigvee_{j' \ne j} x_{ij'}^{k'} \equiv \bigvee_{i' \ne i} x_{i'j}^{k'}$ for all layered graph input assignments.

2.2 Restrictions $\mathcal{R}_{n,k}^{\ell}$

Using standard terminology we say that a *restriction* is a partial assignment of Boolean values to the input variables. Any variable not assigned we say is *unset.* We will follow standard notation using $f\lceil_\rho, A\lceil_\rho, \dots$ for the application of a restriction ρ to a function, set, etc. and $\rho\sigma$ for the restriction which is the union of the assignments given by ρ and σ (assuming that ρ and σ assign values to different variables.)

Define $\mathcal{R}_{n,k}^{\ell}$ to be the set of all restrictions ρ on graphs from $\mathcal{G}(n,k)$ constructed as follows: For each i, $0 \le i \le k$, choose a set $U_i \subset V_i$ of exactly ℓ *unset* vertices per layer. Then choose a member G' of $\mathcal{G}(n-\ell, k)$ whose vertex layers are $V_0 - U_0, \dots, V_k - U_k$. The variables unset by ρ will be $x_{ij}^{k'}$ such that $i \in U_{k'}$ and $j \in U_{k'+1}$. For the remaining variables $x_{ij}^{k'}$, if both $i \in V_{k'} - U_{k'}$ and $j \in V_{k'} - U_{k'}$ then $x_{ij}^{k'}$ is set to represent G'; otherwise $x_{ij}^{k'}$ is set to 0.

The key motivation for defining this set of restrictions is that for any $\rho \in \mathcal{R}_{n,k}^{\ell}$ we can identify $\mathcal{G}(n,k)\lceil_\rho$ with $\mathcal{G}(\ell,k)$ under a suitable renaming of vertices.

2.3 Decision trees for layered graphs

A *decision tree for layered graphs over* $\mathcal{G}(n,k)$ is defined as follows. It is a rooted tree with each interior node labeled by a query which is a pair consisting of a vertex $v \in V_i$ and either $+$ or $-$ indicating a forward or backward query. For the query $\langle v, + \rangle$, the outedges of the interior node are labelled by the possible choices of the *forward edge* containing v, (v, w), where $w \in V_{i+1}$; similarly for query $\langle v, - \rangle$ the edges are labelled by the choices of the *backward edge* containing v, (u, v), where $u \in V_{i-1}$. There will be one outedge from the interior node for each choice that preserves the property that the edge labels along every path in the decision tree define a partial layered graph over V. Note that if $v \in V_0$ or $v \in V_k$, only one type of query is possible. The leaves of the decision tree are labeled with either "0" or "1".

A decision tree T over V *represents* a function f over the domain of layered graphs $\mathcal{G}(n,k)$ provided that for all leaf nodes l in T, if we let σ be the partial layered graph defined by the edge labels in the path in T from the root to l, then for all (complete) layered graphs α in $\mathcal{G}(n,k)$ that are consistent with σ, $f(\alpha)$ is equal to the label of l.

Note this is just the usual definition of a decision tree, only modified for truth assignments that define layered graphs. Also note that unlike the similar decision trees constructed in [7, 6] this representation is exact in that every graph in $\mathcal{G}(n,k)$ will be consistent with some root to leaf path in T.

Let $V' \subseteq V$. We now define a decision tree that determines the values of all variables with endpoints in V'. The *complete decision tree* for V' is defined as follows. If $V' = \emptyset$ then it consists of a single vertex (leaf). Otherwise, assume that we have created the complete decision tree for $V'' = V' \setminus \{v\}$; we will now extend it to a complete tree for V'. For each leaf node l of the complete decision tree for V'' we do the following: Let p_l be the path from the root to l. The edge labellings along p_l define a partial layered graph where all of the edge variables with endpoints vertices in V'' are completely determined. If $v \notin V_k$ and this partial setting does not fix the forward edge of v, then label l by the query $\langle v, + \rangle$ and add new edges leading out of l, one for every consistent value for the forward edge out of v. Then if $v \notin V_0$ and the partial assignment in p_l does not fix the backward edge of v, label each leaf below l by the query $\langle v, - \rangle$ and add new outedges, one for every consistent value for the backward edge out of v.

3 The lower bound

In this section we prove our main lower bound for connectivity. The overall idea of the proof follows the bottom-up, random-restriction method from [13] although formally, rather than simplifying the circuits after applying restrictions, we follow [6, 5] in showing that after these restrictions are applied the functions computed by the gates of the circuit have some simple property, namely the ability to be represented as a small height decision tree. As in [7, 5], we rephrase the notion of a switching lemma as arguing about the probability that, after the application of a randomly chosen restriction, a disjunctive normal form (DNF) formula, each of whose terms is of bounded size, can be represented by a decision tree of small height. A more traditional switching lemma that converts a DNF formula with small terms to a CNF formula with small clauses is a corollary of our lemma.

Before we proceed to make use of this machinery of restrictions and decision trees, we need to justify its connection to the $STCONN(k(n))$ problem. After all, $STCONN(k(n))$, in addition to an input graph, has as input two distinguished vertices s and t. The idea is to work with a distance-bounded transitive closure problem. Let $CONN(n,k)$ be the following problem with n^2 output bits: On inputs $x_{ij}^{k'}$ representing a graph G in $\mathcal{G}(n,k)$, determine for each pair $s \in V_0$ and $t \in V_k$ whether or not s is connected to t in G.

Lemma 1: If C' is a circuit of size S and depth d solving st-connectivity for all members of $\mathcal{G}(n,k)$ then there is a circuit C of size n^2S and depth d solving $CONN(n,k)$.

Proof: Create n^2 reduced circuits from C' by hard-wiring in each of the n^2 pairs of $s \in V_0$ and $t \in V_k$. C is simply the union of these circuits. $\square$

We now consider some unbounded fan-in circuit C solving $CONN(n,k)$. For convenience we will assume that C has only two kinds of gates, unbounded fan-in $\vee$ gates and fan-in 1 $\neg$ gates. We will only count the $\vee$ gates for size or depth so these measures correspond to the usual ones.

We can represent the literals at the leaves of this circuit by decision trees of height 1. We will show that if C has small size we can apply a random restriction which allows us to represent the depth 1 subfunctions at the bottom level of the circuit by small-depth decision trees. We apply this argument repeatedly to higher and higher levels in the tree until we end up with decision trees that represent each of the functions computed by the outputs of C. If C is not too deep then the final decision trees will all have height less than k and since the restriction will leave some $\mathcal{G}(n',k)$ unset for $n' > 1$, it will be easy to get a contradiction.

Definition 3.1: An *s-disjunction* is a DNF formula in the variables $x_{ij}^{k'}$, each of whose terms contains at most s variables all of which appear positively. Furthermore, each term is consistent with some layered graph in $\mathcal{G}(n,k)$.

(Since we have assumed that our input graphs are from $\mathcal{G}(n,k)$ for some n, as noted in section 2.1, we can remove any negated variables in a DNF formula without changing the lengths of its terms.)

Let T be a decision tree over $\mathcal{G}(n,k)$ that represents a function f. If T has depth d then over the domain $\mathcal{G}(n,k)$, f is equivalent to a d-disjunction f' which has one term t_p for each path p in T that leads to a leaf labelled 1 where t_p is the conjunction of the variables that correspond to the edge labels along p.

Lemma 2: (Connectivity Switching Lemma) Let f be an r-disjunction over $\mathcal{G}(n,k)$ and ρ be a randomly chosen restriction from $\mathcal{R}_{n,k}^{\ell}$ and let s satisfy $4r^2s^2k < \ell$. With probability at least $\gamma = 1 - (3e\ell^{r+1}(2k)^r r/n)^s$, $f\lceil_\rho$ can be represented by a depth $4r^2s$ decision tree over $\mathcal{G}(\ell,k)$.

We postpone the proof of this lemma to the next section and first see how it can be used to obtain our desired lower bound.

The following lemma states that there exists a restriction which allows us to represent a depth d circuit by a small-height decision tree.

Lemma 3: Suppose that C is a circuit of size S and depth d in variables $x_{ij}^{k'}$ for $1 \le i,j, \le n$ and $0 \le k' < k$. Let $n_0 = n$, $r_0 = 4$, $s_0 = 4\log_n S$, and for every $i < d$, let $r_{i+1} = 4r_i^2 s_i$, $s_{i+1} = 4r_i s_i$, and $n_{i+1} = n_i^{1/4r_i}$. If $n_d > (3er_d(2k)^{r_d})^3$ then for each i, $0 \le i \le d$, there is a restriction $\rho_i \in \mathcal{R}_{n,k}^{n_i}$ such that for every gate g of C of depth at most i, $g\lceil_{\rho_i}$ can be represented by a decision tree of height at most r_i.

Proof: We first observe that

$$n_0^{-s_0/3} = n^{-(4/3)\log_n S} = S^{-4/3} < 1/S$$

and note that by our choices of parameters, for each $i \ge 0$, $n_i^{-s_i/3} = n_0^{-s_0/3} < 1/S$. Furthermore, the r_i and s_i values increase with i and the n_i values decrease with i.

Using these facts about our parameters, we now prove the lemma by induction on i. It suffices to argue about $\vee$-gates, because for $\neg$-gates, a decision tree for $\neg g$ is exactly the same as that for g except that the leaf labels 1 and 0 are reversed.

Base Case: $i = 0$. The gates at depth 0 are either inputs or their negations and these can be represented by decision trees of height $1 < r_0$. We thus let ρ_0 be the empty restriction.

Induction Step: Suppose that there is a restriction $\rho_i \in \mathcal{R}^{n_i}_{n,k}$ so that for all gates g of depth at most $i \leq d-1$, $g\lceil_{\rho_i}$ has a decision tree of height at most r_i. Consider any $\vee$-gate g at depth $i+1$. By the inductive hypothesis all the inputs to this gate can be represented by decision trees of height at most r_i. Therefore the functions computed at those gates can be expressed as r_i-disjunctions over $\mathcal{G}(n_i,k)$. Since g is an $\vee$-gate it follows that $g\lceil_{\rho_i}$ can be expressed as an r_i disjunction over $\mathcal{G}(n_i,k)$. Observe that

$$4r_i^2 s_i^2 k \leq 4r_{d-1}^2 s_{d-1}^2 k \leq r_d^2 k < n_d \leq n_{i+1}.$$

Thus we can apply Lemma 2 to $g\lceil_{\rho_i}$ with $r=r_i$, $s=s_i$, $n=n_i$, and $\ell=n_{i+1}$ to show that less than a

$$(3en_{i+1}^{r_i+1} r_i (2k)^{r_i}/n_i)^{s_i}$$

fraction of all restrictions $\rho \in \mathcal{R}^{n_{i+1}}_{n_i,k}$ fail to keep the decision tree height of $g\lceil_{\rho_i\rho}$ at most $4r_i^2 s_i = r_{i+1}$. Now, since $n_d > (3er_d(2k)^{r_d})^3$, by the properties of r_i and n_i we have $n_i > (3er_i(2k)^{r_i})^3$. Thus the failure probability is at most

$$\begin{aligned}
&(3en_{i+1}^{r_i+1} r_i (2k)^{r_i}/n_i)^{s_i}\\
&\leq (n_{i+1}^{r_i+1}/n_i^{2/3})^{s_i}\\
&\leq (n_{i+1}^{5r_i/4}/n_i^{2/3})^{s_i} \qquad \text{since } r_i \geq 4\\
&= (n_i^{5/16}/n_i^{2/3})^{s_i} < n_i^{-s_i/3} < 1/S
\end{aligned}$$

Since there are at most S $\vee$-gates of depth $i+1$, there is some fixed restriction $\rho \in \mathcal{R}^{n_{i+1}}_{n_i,k}$ such that for all gates at depth $i+1$, applying $\rho_i\rho$ leaves their decision tree height at most r_{i+1}. Letting $\rho_{i+1} = \rho_i\rho$ we see that the conditions of the lemma hold. □

Theorem 4: Let $F_{-1}=1$, $F_0=0$ and $F_{i+1}=F_i+F_{i-1}$ for $i \geq 0$ be the Fibonacci numbers. Let $k \leq \log n$. For sufficiently large n and k, any unbounded fan-in, depth d circuit for $CONN(n,k)$ requires size at least $n^{\delta_d k^{1/(3F_{2d})}}$ where $\delta_d = 4^{-(F_{2d+3}-1)/F_{2d}}$.

Proof: Let S be the size of a depth d unbounded fan-in circuit C that computes $CONN(n,k)$. Consider the recurrences from Lemma 3. It is easy to solve them and derive that for $i \geq 0$,

$$\begin{aligned}
s_i &= 4^{F_{2i+2}}(\log_n S)^{F_{2i-1}},\\
r_i &= 4^{F_{2i+3}-1}(\log_n S)^{F_{2i}},\\
n_i &= n^{1/\prod_{j=0}^{i-1} r_j},
\end{aligned}$$

and

$$\prod_{j=0}^{i-1} r_j = 4^{F_{2i+2}-(i+1)}(\log_n S)^{F_{2i-1}-1} < r_i.$$

Suppose that $S < n^{\delta_d k^{1/(3F_{2d})}}$. Then $\log_n S < \delta_d k^{1/(3F_{2d})}$ and thus $r_d = 4^{F_{2d+3}-1}(\log_n S)^{F_{2d}} < k^{1/3}$. Also $(3er_d(2k)^{r_d})^3 \leq k^{4r_d} \leq k^{4k^{1/3}}$. Now $n_d \geq n^{1/r_d} \geq n^{1/k^{1/3}}$ and since $k^{4k^{2/3}} < n$ for $k \leq \log n$ we have $n_d > (cr_d k^{r_d})^3$. Thus we can apply Lemma 3 to find a restriction ρ_d from $\mathcal{R}^{n_d}_{n,k}$ such that every output gate g of C, $g\lceil_{\rho_d}$ can be represented by a decision tree of height less than k over $\mathcal{G}(n_d,k)$. In particular this holds for the output nodes corresponding to pairs composed by taking one of the n_d choices of $s \in V_0$ and one of the n_d choices of $t \in V_k$ that are left unset by ρ_d. But this is impossible because a decision tree of height k on $\mathcal{G}(n_d,k)$ cannot determine if such an st pair is connected. This is a contradiction and thus the theorem holds. □

Corollary 5: Let $\phi = (\sqrt{5}+1)/2$ be the golden mean. Then there is a constant c such that for $k \leq \log n$, any depth d unbounded fan-in circuit for $CONN(n,k)$ requires size at least $n^{ck^{\phi^{-2d}/3}}$.

Corollary 6: For any $k(n) \leq \log n$, any depth d unbounded fan-in circuit for $STCONN(k(n))$ requires size $n^{\Omega(k^{\phi^{-2d}/3})}$.

Corollary 7: For $k(n) \leq \log^{O(1)} n$, any polynomial-size unbounded fan-in circuit for $STCONN(k(n))$ requires depth $\Omega(\log\log k(n))$.

4 The connectivity switching lemma

The idea of our switching lemma proof is as follows. Let f be an r-disjunction. We want to show that with extremely high probability, a random restriction chosen from the distribution of restrictions has the property that for *any* consistent partial assignment T, $(f\lceil_\rho)\lceil_T$ has at most s 'independent' terms. (This is the main technical sub-lemma.) If this is true, then we can build a decision tree for $f\lceil_\rho$ that queries all variables in the s independent terms. Since all other terms are dependent, we show that this reduces the overall term-size by at least one. Then applying the sub-lemma again, we can find another set of at most s independent terms and query all of the variables in these s terms. After continuing this process at most r times, we are guaranteed to terminate since all terms have been reduced to size 0.

The main ideas of the argument may be used in a simpler form to prove a switching lemma for restrictions under the uniform distribution. Although this is much weaker than the switching lemma from [14] the argument illustrates the essential features of our technique more clearly than does the proof of the connectivity switching lemma. We include it for pedagogical purposes in an Appendix.

4.1 A restriction lemma

We will first state and prove the main technical sublemma for the switching lemma for connectivity restrictions. Let ρ be a restriction, and let T be a set of edges. Let $C_1, \ldots, C_s$ be sets of edges. We say that $C_1, \ldots, C_s$ are T-*consistent* if there is a layered graph containing $\rho \cup T \cup C_1 \cup \cdots \cup C_s$. We say that the collection $C_1, \ldots, C_s$ is T-*independent* if any edge in more than one C_i is from $\rho \cup T$, and if there are no $i \neq j$ and edges $e \in C_i - \rho - T$, $e' \in C_j - \rho - T$ so that e and e' are on the same path in $T \cup \{e, e'\}$.

Lemma 8: Let f be an r-disjunction over $\mathcal{G}(n,k)$ and ρ be a randomly chosen restriction from $\mathcal{R}^{\ell}_{n,k}$. Let s be any integer with $4s^2r^2k < \ell$. Then the probability that there is a set of edges T such that
(a) f restricted by $\rho \cup T$ is not identically 1, and
(b) there are s T-consistent and T-independent terms from f,
is at most $(3er\ell^{r+1}(2k)^r/n)^s$.

Proof: Call a restriction ρ *bad* if there is such a T and set of s terms. We will show that any bad ρ can be recovered from a non-negligible fraction of layered graphs consistent with it, and a small amount of additional information. On the other hand, there are many restrictions consistent with any possible layered graph, so in general it is impossible to recover an arbitrary restriction from a layered graph consistent with it. Thus, the bad restrictions form only a small fraction of the possible restrictions.

More precisely, let ρ be a bad restriction and let T be the set of edges and $C_1, \ldots, C_s$ be a set of T-independent and T-consistent terms. (We assume that T and $C_1, \ldots, C_s$ are chosen in some canonical fashion as a function of ρ.) Let $T' = \cup_{i=1}^{i=s}(T \cap C_i - \rho)$, i.e., those edges of T that actually occur in one of the terms but not in ρ. It is easy to see that $C_1, \ldots, C_s$ are T'-independent and T'-consistent, and that $\rho \cup T'$ does not force f to be identically 1. Let $S = (C_1 \cup \cdots \cup C_s) - \rho$. Note that $|T'| \leq rs$, since $T' \subseteq S$, and $|S| \leq rs$ because the fact that f is an r-disjunction implies that for each i, $|C_i| \leq r$. Moreover, since $C_1, \ldots, C_s$ are T'-consistent, S consists of several disjoint paths in the unrestricted variables. Since $C_1, \ldots, C_s$ are T'-independent, each such path P is contained in $C_i \cup T'$ for some i. Otherwise there would be some edge e from some $C_i - \rho - T'$ in P, and an edge e' from some $C_j - \rho - T'$, $i \neq j$ also in P. Taking the pair of such edges closest together, all intermediate edges would have to be from T', which contradicts T'-independence.

We say that a layered graph G consistent with ρ is an *encoding* if it contains S and if no two paths in S are part of the same path in G. We can pick a random encoding G as follows: let $P_1, \ldots, P_p$, $p \leq rs$, be the paths in S. Extend P_1 backwards and forwards one edge at a time to get a path from the first layer to the last, avoiding any node in any of the other paths. Then repeat with P_2 on the remaining nodes. When all paths have been extended, pick a random layered graph on the remaining $\ell - p$ nodes at each layer. For each of the $pk - |S|$ extension phases, we will have at least $\ell - p$ choices for the edge at that layer. We then have $(\ell - p)!^k$ choices after the extension phases are over. Thus, each bad ρ has at least

$$\begin{aligned}
&(\ell-p)^{pk-|S|}(\ell-p)!^k \\
&\quad\geq\ (\ell-p)^{pk}(\ell-p)!^k/\ell^{|S|} \\
&\quad=\ (1-p/\ell)^{pk}\ell^{pk}(\ell-p)!^k/\ell^{|S|} \\
&\quad\geq\ (1-p/\ell)^{pk}(\ell!)^k/\ell^{|S|} \\
&\quad\geq\ e^{-4p^2k/\ell}(\ell!)^k/\ell^{|S|} \\
&\quad\geq\ e^{-1}(\ell!)^k/\ell^{|S|}
\end{aligned}$$

encodings since $4p^2k \leq 4(rs)^2k \leq \ell$. Thus, for any bad ρ, the fraction of extensions that are encodings is at least $e^{-1}(\ell)^{-|S|}$. (In other words, the fraction of consistent extensions that are encodings is approximately the same as the fraction containing S, and each edge of the at most rs edges in S is included with probability $1/\ell$.)

Given any encoding G of ρ, we can give a small amount of additional information that will allow us to compute ρ using the fact that we know f. This is equivalent to finding which ℓ of the n paths in G were unrestricted by ρ. We show how from G and $s(\log r+1)+rs(\log k+1)$ bits of well-chosen information (advice), we can compute s of the unrestricted paths in ρ. We can then specify explicitly which of the $\binom{n-s}{\ell-s}$ choices for the set of remaining paths is correct.

The decoding method is as follows. Since G contains S and ρ, G forces all of $C_1, \ldots, C_s$ to be true, and possibly other terms from f. Let C'_1 be the first such term. Since $\rho \cup T'$ did not force f to be true, there must be some edge e_1 in $C'_1 - \rho - T'$, and the accompanying path P_1 must be unrestricted by ρ. We use the first $\log r$ bits to specify e_1 among the r edges in C'_1. The location of all the edges in $P_1 \cap T'$ will be given to us by the layer each such edge occurs at, using $\log k$ bits per such edge. (Technically, we use the first bit to say if there are any such edges; if there are we use the next $\log k$ bits to obtain the first edge, and the next bit tells us whether there are any more along the same path.) We delete all the edges in $P_1 - T'$ from G, and find another term C_2' still forced to true. As before, this gives us another path P_2 which was unrestricted in ρ. We repeat until s paths are found.

The process would only be forced to stop before s paths are found if at some previous stage there were no terms from f forced to true. But originally all of $C_1, \ldots, C_s$ are set to true by G, i.e., contained in G. By the property of the encoding, each P_j contains at most one path from S, and so $P_j - T'$ is contained in some C_k, and is thus disjoint from the other C_i's. Therefore, by deleting the edges in $P_j - T'$, we cause at most one of the C_i's to be no longer contained in G. Hence, we can repeat the process at least s stages, since at any point before then, at least one C_i remains forced to true.

The total number of advice bits we use is $\log r+1$ per stage, and an additional $\log k + 1$ per edge of T' found. Since there are s stages and there are at most rs edges in T', this is at most $s(\log r + 1) + rs(\log k + 1)$.

Thus, each layered graph G can be an encoding of at most $(2r)^s(2k)^{rs}\binom{n-s}{\ell-s}$ bad restrictions, out of $\binom{n}{\ell}$ restrictions consistent with G. Thus, the probability, picking a random pair G and ρ with G extending ρ, that G is an encoding of ρ is at most

$$\begin{aligned}\frac{(2r)^s(2k)^{rs}\binom{n-s}{\ell-s}}{\binom{n}{\ell}} &= \frac{(2r)^s(2k)^{rs}(n-s)!(\ell)!}{n!(\ell-s)!}\\ &\leq \frac{(2r)^s(2k)^{rs}(\ell)^s}{(n-s)^s}\\ &\leq (3r(2k)^r\ell/n)^s\end{aligned}$$

since $n - s \geq 2n/3$. On the other hand, the conditional probability that G is an encoding of ρ, given that ρ is bad, is at least $e^{-1}(\ell)^{-|S|} \geq e^{-1}(\ell)^{-rs}$. Thus, the probability that ρ is bad is at most $\frac{(3r(2k)^r\ell/n)^s}{e^{-1}\ell^{-rs}} = (3er(2k)^r\ell^{r+1}/n)^s$ as required. $\square$

4.2 Proof of the switching lemma

We are now ready to prove Lemma 2.

Proof: (Proof of Lemma 2.) By Lemma 8, with probability γ, a random restriction ρ drawn from $\mathcal{R}^{\ell}_{n,k}$ has the following property, $\mathcal{P}$: For all consistent collections T of unrestricted edges so that $f\lceil_{\rho\cup T}$ is not identically 1, any maximal collection of T-independent, T-consistent terms in $f\lceil_\rho$ has size at most s. We will show that from any ρ with property $\mathcal{P}$, we can construct a depth $4r^2s$ decision tree for $f\lceil_\rho$.

Fix ρ with property $\mathcal{P}$. We will implicitly describe the decision tree for $f\lceil_\rho$ by giving a procedure to decide the value of $f\lceil_\rho$ by making queries to the predecessors and successors of unrestricted nodes; the depth of the tree will be the worst case number of queries.

The procedure operates in r stages, and in each stage at most $4rs$ queries are made. When a successor query concerning node u is made, and the answer v is obtained, then we say that edge (u, v) has been *discovered* by the procedure, and similarly for predecessor queries. Let T_i be the set of edges discovered in the first i stages, and define $T_0 = \emptyset$. Stage $i + 1$ of the procedure is as follows: If $f\lceil_{\rho\cup T_i}$ is identically 1, halt and accept. If no terms of f are consistent with $\rho \cup T_i$, halt and reject. Otherwise, there must be a maximal collection of T_i-independent and T_i-consistent terms $C_1, \ldots, C_{s'}$ with $s' \leq s$. Let S_{i+1} be the set of nodes mentioned in some edge of some C_j. Since each C_j is an r-conjunction, and each variable involves 2 vertices, $|S_{i+1}| \leq 2rs$. Then for each $v \in S_{i+1}$, make the following queries: if v is not in any edge of T_i, then query the predecessor and successor of v. Otherwise, v is in some path P of T_i; query the predecessor of the first node of P and the successor of the final node of P. In either case, at most 2 queries are made per vertex, so the total number of queries in stage i is at most $2|S_{i+1}| \leq 4rs$.

To see that the above procedure halts within r stages, consider any term C of f. We claim that if C is consistent with $\rho \cup T_i$ then at least i edges of C have been discovered in the first i stages. In particular, until C becomes inconsistent with $\rho \cup T_i$, at least one new edge of C is discovered in each stage. Assume C is consistent with $\rho \cup T_i$, and let $C_1, \ldots, C_{s'}$ be the maximal collection of T_i-consistent and T_i-independent terms found in stage $i + 1$. If C is one of the terms in this collection, all of its nodes are queried. Thus, either it will become inconsistent or all of its edges will be discovered.

Now suppose that C is not among $C_1, \ldots, C_{s'}$. Then $C, C_1, \ldots, C_{s'}$ is either T_i-inconsistent or T_i-dependent.

In the first case, since C is consistent with T_i, and all the other terms together are consistent with T_i, $C - T_i$ must be inconsistent with some $C_j - T_i$. Thus there must be an edge $e = (u, v)$ of $C - T_i$ and an edge $e' = (u', v')$ from $C_j - T_i$ with $u = u'$ or $v = v'$. Assume $u = u'$; the case $v = v'$ is similar. Then since $e' \notin T_i$, u's successor will be queried in stage $i+1$, and so either C will become inconsistent or e will be discovered.

In the other case, $C, C_1, \ldots, C_{s'}$ are T_i-consistent but not T_i-independent. Then there is an edge $e = (u, v) \in C - T_i$ and an edge $e' = (u', v')$ in some $C_j - T_i$ connected via a (possibly empty) path $P \subseteq T_i$. Assume that P starts at v and ends at u'; the reverse case is similar. Then the queries for stage $i + 1$ involving u' will include the predecessor of v. Thus either e and hence C will become inconsistent, or e will be discovered in stage $i + 1$.

Thus, if the procedure continues for r stages, every term has either become inconsistent, or has had all of its edges discovered. If any term has all of its edges discovered, the function is forced to 1; if all terms have become inconsistent, the function is forced to 0. In either case, the procedure halts after the r-th stage. $\square$

5 Concluding Remarks

It would be very nice to close the gap significantly further between the $O(\log k(n))$ upper bound and our $\Omega(\log\log k(n))$ lower bound for polynomial-size circuits solving $STCONN(k(n))$. It is easy to improve our lower bound slightly by replacing the $4r^2s$ in the statement of Lemma 2 by $4\binom{r+1}{2}s$ but this has virtually no effect on the asymptotics of the depth lower bound. However, if one could improve this $4r^2s$ bound to a function of r and s whose total degree were $1+o(1)$ then one would obtain a substantial improvement in the depth lower bound.

One of our original motivations for considering $STCONN(k(n))$ was that the lower bound of [2] was the only one using an independent-set-style switching lemma that seemed impossible to improve using a Håstad-style switching lemma. We view our new switching lemma as progress towards developing a general switching lemma that might give a simple characterization of the properties on a family of restrictions that permit a switching lemma to be proven.

Acknowledgements

We would like to thank Avi Wigderson, Noam Nisan, Pavel Pudlak, Jan Krajicek, Petr Savicky, Peter Clote, and Jiri Sgall for helpful discussions about this work.

From @yonge.cs.toronto.edu,@dvp.cs.toronto.edu:beame@cs.toronto.edu Thu Aug 3 21:07:44 1995 Received: from yonge.cs.toronto.edu by odin.ucsd.edu; id AA20308 sendmail 5.67/UCSDPSEUDO.4-CS via SMTP Thu, 3 Aug 95 21:07:41 -0700 for russell Received: from dvp.cs.toronto.edu ([128.100.1.15]) by yonge.cs.toronto.edu with SMTP id ¡86519¿; Fri, 4 Aug 1995 00:07:27 -0400 Received: by dvp.cs.toronto.edu id ¡15327¿; Fri, 4 Aug 1995 00:07:12 -0400 From: Paul Beame ¡beame@cs.toronto.edu¿ To: russell@cs.ucsd.edu, toni@cs.pitt.edu Subject: conn.bbl Content-Length: 4960 Message-Id: ¡95Aug4.000712edt.15327@dvp.cs.toronto.edu¿ Date: Fri, 4 Aug 1995 00:07:10 -0400 Status: R

References

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

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

[3] Greg Barnes and Jeff A. Edmonds. Time-space lower bounds for directed s-t connectivity on JAG models. In *Proceedings 34th Annual Symposium on Foundations of Computer Science*, pages 228–237, Palo Alto, CA, November 1993. IEEE.

[4] Greg Barnes and Uri Feige. Short random walks on graphs. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 728–737, San Diego, CA, May 1993.

[5] Paul W. Beame. A switching lemma primer. Technical Report 95–07–01, Department of Computer Science and Engineering, University of Washington, November 1994.

[6] Paul W. Beame, Russell Impagliazzo, J. Krajíček, T. Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 794–806, Santa Fe, N.M., November 1994. IEEE.

[7] Paul W. Beame, Russell Impagliazzo, J. Krajíček, T. Pitassi, Pavel Pudlák, and A. Woods. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 200–220, Victoria, B.C., Canada, May 1992.

[8] S. Bellantoni, T. Pitassi, and A. Urquhart. Approximation and small depth Frege proofs. In *Proceedings, Structure in Complexity Theory, Sixth Annual Conference*, pages 367–391, Chicago, IL, June 1991. IEEE.

[9] Jin-Yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 21–29, Berkeley, California, May 1986.

[10] Jeff Edmonds and Chung Keung Poon. A nearly optimal time-space lower bound for the graph connectivity problem on the NNJAG. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 147–156, Las Vegas, NV, May 1995.

[11] Jeff A. Edmonds. Time-space trade-offs for undirected ST-connectivity on a JAG. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 718–727, San Diego, CA, May 1993.

[12] Uriel Feige. A randomized time-space tradeoff of $\tilde{O}(m\hat{R})$ for USTCON. In *Proceedings 34th Annual Symposium on Foundations of Computer Science*, pages 238–246, Palo Alto, CA, November 1993. IEEE.

[13] Merrick L. Furst, J. B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. In *22nd Annual Symposium on Foundations of Computer Science*, pages 260–270, Nashville, TN, October 1981. IEEE.

[14] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 6–20, Berkeley, CA, May 1986.

[15] Noam Nisan, Endre Szemerédi, and Avi Wigderson. Undirected connectivity in $O(\log^{1.5} n)$ space. In *Proceedings 33rd Annual Symposium on Foundations of Computer Science*, pages 24–29, Pittsburgh, PA, October 1992. IEEE.

[16] A. Razborov. Bounded arithmetic and lower bounds in Boolean complexity. Submitted to Feasible Mathematics II, 1993.

[17] Avi Wigderson. The complexity of graph connectivity. In I. M. Havel and V. Koubek, editors, *Mathematical Foundations of Computer Science 1992: Proceedings, 17th Symposium*, volume 629 of *Lecture Notes in Computer Science*, pages 112–132, Prague, Czechoslovakia, August 1992. Springer-Verlag.

[18] A. C. Yao. Separating the polynomial hierarchy by oracles: Part I. In *26th Annual Symposium on Foundations of Computer Science*, pages 1–10, Portland, OR, October 1985. IEEE.

[19] A. C. Yao. A lower bound for the monotone depth of connectivity. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 302–308, Santa Fe, N.M., November 1994. IEEE.

Appendix

In this appendix we prove the switching lemma for the case of the uniform distribution using our new technique. It is independent of the rest of the paper. Stronger results appear in the literature (Håstad [14], Cai [9]) but we include this proof to illustrate our technique.

Let f be a DNF formula with term size $\leq r$ over $\{x_1, \ldots, x_n\}$, and ρ be chosen uniformly from $\mathcal{R}_n^\ell$, where $\mathcal{R}_n^\ell$ is the set of all restrictions on $\{x_1 \ldots, x_n\}$ with exactly ℓ unset variables. We say a set of literals is *consistent* if it does not contain both a variable and its negation. For T, $C_1, \ldots, C_s$ sets of literals, we say $C_1, \ldots, C_s$ are T-*consistent* if their union with each other and T is consistent, and we say that they are T-independent if $C_i \cap C_j \subseteq T$ for every $i \neq j$. We identify a set of literals with the minimal restriction that forces each literal to 1. We say that ρ is s-bad for f if there is a set of literals T unset by ρ so that $f\lceil_{\rho \cup T}$ is not identically 1, and there are s T-consistent and T-independent terms $C_1, \ldots, C_s$ in $f\lceil_\rho$

Lemma 9: Let f and ρ be as above, and assume $s \leq \ell \leq n/2$. Then the probability that ρ is s-bad for f is at most $(2r2^r\ell/n)^s$.

Proof: Let ρ be an s-bad restriction for f. Let T be the set of unset literals and $C_1, \ldots, C_s$ be a set of terms from f whose restrictions are T-independent and T-consistent terms from $f\lceil_\rho$. (Given a ρ that is s-bad for f, we choose T and $C_1, \ldots, C_s$ in a canonical way, e.g. the lexicographically first such choices that work.) Let $S = (C_1 \cup \cdots \cup C_s) - \rho$. Note that $|S| \leq rs$.

A truth assignment consistent with ρ is an *encoding* if it makes all literals in S true. The number of encodings is thus at least $2^{\ell - rs}$ out of 2^ℓ consistent truth assignments, and thus for any ρ s-bad for f, the fraction of extensions that are encodings is at least 2^{-rs}.

Given any encoding G of ρ, and a relatively small amount of additional information (*advice*), we will be able to find s unset variables in ρ as follows: Because G contains S and ρ, G forces all of $C_1, \ldots, C_s$ to be true, and possibly other terms from f as well. Let C'_1 be the lexicographically first such term. Since $\rho \cup T$ did not force f to be true, there must be some literal in $C'_1 - \rho - T$. We use the first $\log r$ bits of advice to find this literal among the r literals in C'_1. Then we unset this variable from G, and find the next term C'_2 in f that is still forced to true. We repeat this process until s literals have been found.

This process would only be forced to stop before s literals are found if at some previous stage there were no terms from f forced to true. But originally all of $C_1, \ldots, C_s$ are set to true by G. Because the C_i's are T-independent, by unsetting any one literal not in T, we cause at most one of the C_i's to be no longer contained in G. Thus, this process continues for at least s stages.

Because the total number of advice bits we use is

$\log r$ per stage, the total number of advice used in the decoding is $s \log r$. Thus each total assignment can be an encoding of at most $r^s \binom{n-s}{\ell-s}$ restrictions that are s-bad for f, out of $\binom{n}{\ell}$ restrictions consistent with G, since after we find s unset variables, we need to specify the assignment to the remaining $\ell - s$ unset variables to totally specify ρ.

Thus the probability, when picking a random pair G and ρ with G extending ρ, that G is an encoding of ρ is at most

$$\frac{r^s \binom{n-s}{\ell-s}}{\binom{n}{\ell}} = \frac{r^s (n-s)!\, \ell!}{n!\, (\ell-s)!} \leq \frac{r^s \ell^s}{(n-s)^s} \leq (2r\ell/n)^s$$

since $n - s \geq n/2$. On the other hand, the conditional probability that G is an encoding of ρ, given that ρ is s-bad for f, is at least 2^{-rs}. Thus the probability that ρ is s-bad for f is at most $\frac{(2r\ell/n)^s}{2^{-rs}} = (2r2^r\ell/n)^s$, as required. $\square$

Lemma 10: Let f and ρ be as above. If ρ is not $(s+1)$-bad for f, then $f\lceil_\rho$ has a Boolean decision tree of height at most $r^2 s$.

Proof: Since ρ is not $(s+1)$-bad for f, $f\lceil_\rho$ has a maximal consistent and independent set of at most s terms. Query the at most rs unset variables mentioned in these terms. Any answers to these queries shortens every term in $f\lceil_\rho$ by at least one, since no term is disjoint from these variables. Let T_1 be the set of literals corresponding to the answers. If $f\lceil_{\rho \cup T_1}$ is not identically 1, find and query a maximal set of T_1-independent and consistent terms, and let T_2 add the set of answers to these queries to T_1. Repeat until $f\lceil_{\rho \cup T_i}$ is identically 1 or 0. Since each stage shortens every term by 1, this will occur within r stages, for a total of at most $r^2 s$ queries. $\square$