Bounds on the Size of Sound Monotone Switching Networks Accepting Permutation Sets of Directed Trees

Joshua Brakensiek

under the direction of
Mr. Aaron Potechin
Massachusetts Institute of Technology

Research Science Institute
July 31, 2012

Abstract

Given a directed graph G as input with labeled nodes s and t, the ST-connectivity problem asks whether s and t are connected. The memory efficiency of an algorithm which solves this problem can be analyzed using sound monotone switching networks. This paper concerns bounds on the size of sound monotone switching networks which are restricted to the case that the input graph G is isomorphic to a given graph H. Previously, tight bounds had been found in the cases that H is a special kind of tree and that H is a collection of disjoint paths from s to t. This paper improves these results to find a nearly tight bound which applies to all directed trees. If we let n be the number of vertices in the graph, ℓ be the length of the path from s to t in the tree, and C_1 and C_2 be variables which depend on the distances of the vertices from s and t, then an upper bound on the size is on the order of $n^{\log\log\ell}C_1^{\log\ell}$ and the lower bound is on the order of $C_2^{\log\ell}$. These two bounds are within $\log\log\ell$ times a constant factor in the exponent.

Summary

A graph is a collection of points with edges drawn between some of the nodes. Given a graph with one-way edges, we wish to compute whether there is a path from a starting point to an ending point. If we assume the computation is performed in a certain way, we can calculate roughly the minimal amount of computer memory needed. The problem under consideration is to estimate the amount of memory necessary for special families of graphs. Previously, reasonable estimates had only been found in specific cases. In this paper, we find estimates which apply to a significantly wider range of graphs.

1 Introduction

One major subfield of computer science, *computational complexity theory*, seeks to understand the limits of computation. One long-standing open problem is the minimal space complexity of the ST-connectivity problem. The ST-connectivity problem is formulated as follows: given a directed graph G with starting and ending vertices s and t, is there a way to travel from s to t? This inquiry is not difficult to answer; the challenge is to answer this question with the minimal amount of space (computer memory) necessary. In a celebrated result by Savitch [6], it was shown that one can answer this problem on an n-vertex graph with $O((\log n)^2)$ space[1]. Reingold [5] showed that if G is an undirected graph, then only $O(\log n)$ space is required.

One kind of computation which gives insight into this problem is monotone computation. This kind of computation operates by making deductions from the existence of edges; it does not make deductions from the absence of certain edges. We analyze the ST-connectivity problem using a structure called a *monotone switching network*. Defined more precisely in Section 2, a monotone switching network is an undirected graph with labeled edges based on queries that a program may make about the existence of edges in the input graph. We can think of the vertices of this network as representing possible memory states of the program.

Finding bounds on the size of the monotone switching network roughly corresponds to finding bounds on the amount of space needed to compute ST-connectivity in a monotone computation model. Potechin [2] has shown that in the general case, a monotone switching network needs a size of $n^{\Theta(\log n)}$, which corresponds to needing $\Theta((\log n)^2)$ space. However, finding lower bounds on the size of monotone switching networks does not give us lower bounds on the amount of space needed but does tell us the limits of monotone computation. To obtain general lower bounds on the amount of memory needed, one must analyze a

[1] O, Ω, and Θ notations denote a general measure of growth of a function. They are defined formally in Appendix A.2.

broader class of switching networks, the non-monotone switching networks.

We determine bounds on the sizes of monotone switching networks for special cases of the ST-connectivity problem. In these special cases, we assume that the input graph is isomorphic to a given graph via permutation of the vertices. For example the results in Theorem 4.2 concern the case where every vertex in the given graph has a unique path from s to itself.

In our main result, Theorem 4.4, bounds are found in the case of a general tree. If we define $m(\sigma(G))$ to be the size of the monotone switching network, n to be the number of vertices, ℓ to be the length of the path from s to t, c_i^s and c_i^t be the number of vertices of roughly a certain depth, and $\bar{d}$ the number of vertices not accessible from s or t, then

$$(\ell+\bar{d})^{\Omega(\lg \ell)} \max_{1 \le i \le \lceil \lg\lg \ell \rceil} (c_i^s + c_i^t)^{\Omega(2^i)} \le m(\sigma(G)) \le n^{O(\lg\lg \ell)} (\ell+\bar{d})^{O(\lg \ell)} \prod_{i=1}^{\lceil \lg\lg \ell \rceil} (c_i^s + c_i^t)^{O(2^i)}.$$

1.1 Outline

In Section 2, we formally define monotone switching networks and related terminology. In Section 3, we summarize previous work with monotone switching networks. In that section, we also present existing techniques which are crucial in obtaining the results in this paper. Section 4 provides proof of the main result, bounding the size of sound monotone switching networks in the case of general directed trees.

Appendix A gives definitions of terminology in graph theory and also defines big O, Ω, and Θ notations. Appendix B gives proofs of selected previous results. Appendix C gives a proof of Lemma 4.1, and Appendix D gives a proof of Theorem 4.2.

2 Preliminary Definitions

To discuss monotone switching networks and their properties, the following terminology was introduced by Potechin [2], which we also use.

Definition 2.1. Given a set of vertices $V \cup \{s, t\}$, define a ***monotone switching network*** as an undirected graph G' on the set of vertices $V' \cup \{s', t'\}$. Each edge between two vertices of G' is given a label of the form $a \to b$ where $a, b \in V \cup \{s, t\}$.

An example of a monotone switching network is depicted in Figure 1.

Definition 2.2. Define the ***size*** of a monotone switching network G' as the number of vertices of G'.

We wish to analyze how G' relates to various graphs G on the set of vertices $V \cup \{s, t\}$. Definition 2.3 quantifies this.

Definition 2.3. Given a directed graph G on $V \cup \{s, t\}$, called the ***input graph***, say that a monotone switching network G' ***accepts*** G if and only if there exists a path from s' to t' in G' such that the label of each edge in the path corresponds to an edge of G. For example, the label $a \to b$ corresponds to the directed edge from a to b in G. If G' does not accept G, then G' ***rejects*** G.

For an example, see Figure 1. We analyze monotone switching networks based on which graphs they accept and reject. Definitions 2.4 and 2.5 are general descriptions of what a monotone switching network accepts and rejects.

Definition 2.4. A monotone switching network G' is ***complete*** if it accepts any input graph G for which there is a path from s to t.

Definition 2.5. A monotone switching network G' is ***sound*** if it rejects any input graph G for which there is no path from s to t.

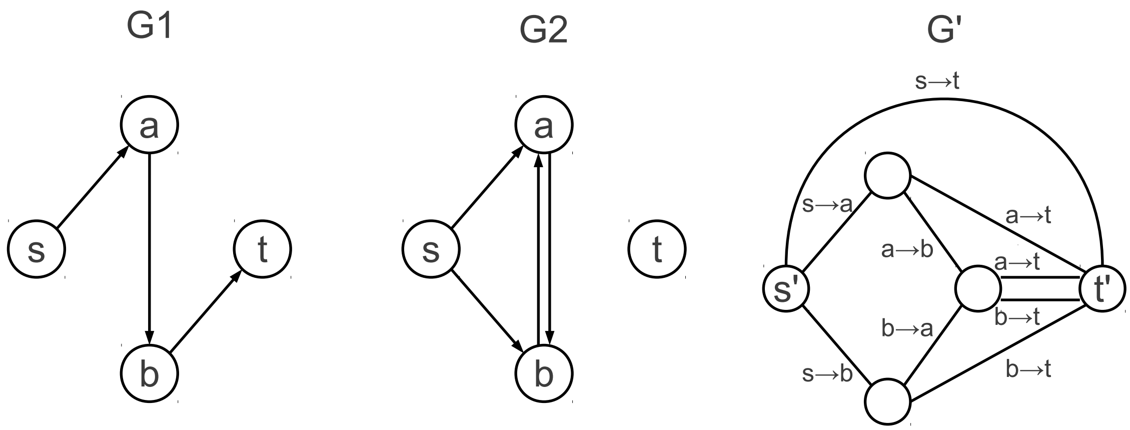

Figure 1: Two input graphs G_1 and G_2 and a monotone switching network G'. Notice that the monotone switching network accepts G_1 because G' has a path from s' to t' with labels $s \to a$, $a \to b$, and $b \to t$, each of which is in G_1. Conversely, G' rejects G_2 because none of the edges to t' have labels which are in G_2. More generally, G' is both complete and sound. The figure of the monotone switching network was obtained from Potechin [4].

Unless explicitly stated, we assume that all monotone switching networks under consideration are sound. On the other hand, almost none of the monotone switching networks under consideration are complete. We require that the computations which a monotone switching network simulates involve sound logical reasoning. Hence, soundness of a monotone switching network is an important property.

Definition 2.6. Given a set I of input graphs on $V \cup \{s, t\}$, where for each graph $G \in I$ there is a path from s to t, define $m(I)$ to be the smallest possible size of a sound monotone switching network which accepts all the elements of I.

In Sections 3 and 4, we find bounds on the value of $m(I)$ for specific sets of graphs I. The sets of graphs we primarily investigate are permutation sets.

Definition 2.7. Let G be a directed graph on the set of vertices $V \cup \{s, t\}$. For any subset W of $V \cup \{s, t\}$, define $\sigma_W(G)$ to be the set of graphs which are all possible permutations of

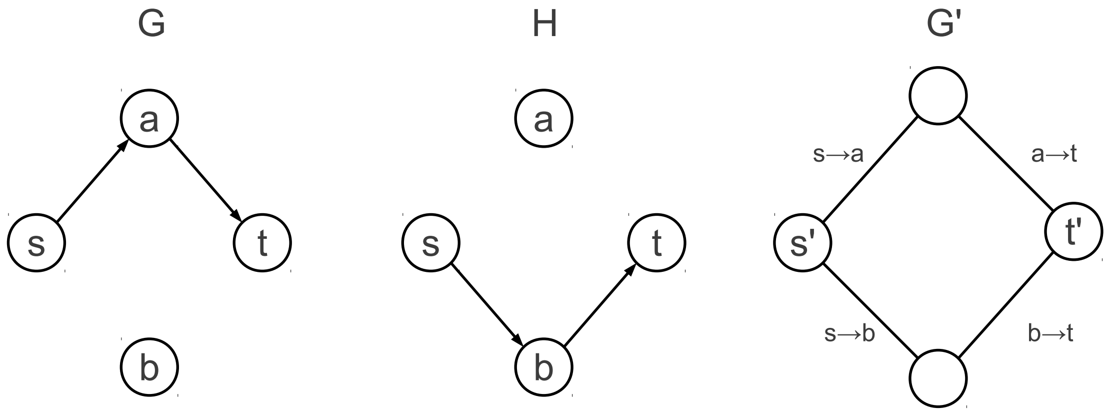

Figure 2: By definition of σ, we have that $\sigma(G) = \{G, H\}$. Notice that the monotone switching network G' accepts both graphs and is sound. Thus, $m(\sigma(G)) \leq 4$. In fact, $m(\sigma(G)) = 4$.

the labels of the vertices $V \cup \{s, t\}$ that fix all vertices in W. Let $\sigma(G) = \sigma_{\{s,t\}}(G)$. These sets are called ***permutation sets***.

We bound the value of $m(\sigma(G))$ for various graphs G. An example is given in Figure 2. To aid in finding these bounds, the results listed in Section 3 are used.

3 Previous Results

Interesting results were obtained by Potechin [2, 3, 4] in bounding the value of $m(I)$ for various input sets I. In Section 3.1, explicit bounds on $m(I)$ by Potechin are listed. In Section 3.2, additional results are given which help bound the value of $m(\sigma(G))$ in the general case.

3.1 Specific Cases

The results discussed in this section, concerning the value of $m(I)$ for various sets of input graphs I, were discovered by Potechin [2, 3, 4]. We assume that for each $G \in I$, its vertices are taken from the set of n vertices $V \cup \{s,t\}$. Let $\mathcal{P}$ be the set of directed graphs such that there is a path from s to t. We then have the following theorem about $\mathcal{P}$.

Theorem 3.1 (Potechin [2])**.** *We have that*

$$m(\mathcal{P}) = n^{\Theta(\lg n)},$$

where $\lg n$ *stands for* $\log_2(n)$*.*

Note. The bound we get for $m(\mathcal{P})$ uses big Θ notation in the *exponent*, instead of as a constant factor. This is because these are the best bounds currently known. These bounds are tight enough for our purposes because they heuristically correspond to an algorithm using $O(\lg(m(\mathcal{P}))) = O((\lg n)^2)$ memory, which is accurate to a constant factor.

Let ℓ be a positive integer less than n. Consider $\mathcal{P}_\ell$, the set of directed graphs such that there is a path from s to t with length ℓ. Length is defined to be the number of edges along the path. Theorem 3.2 gives a bound for this subset of $\mathcal{P}$.

Theorem 3.2 (Potechin [2])**.**

$$m(\mathcal{P}_\ell) = n^{\Theta(\lg \ell)}.$$

Notice the similarity between Theorem 3.2 and Theorem 3.3.

Theorem 3.3 (Potechin [4])**.** *Let G be a graph such that every path from s to t is of length ℓ, and every vertex besides s and t is on exactly one such path. Then*

$$m(\sigma(G)) = n^{\Theta(\lg \ell)}.$$

Figure 4 has an example of G. The asymptotic results of Theorem 3.2 and Theorem 3.3 are identical, although $m(\sigma(G))$ is only a small subset of $\mathcal{P}_\ell$. In some sense, much of the work done by the monotone switching network to accept the elements of $\sigma(G)$ is used to accept

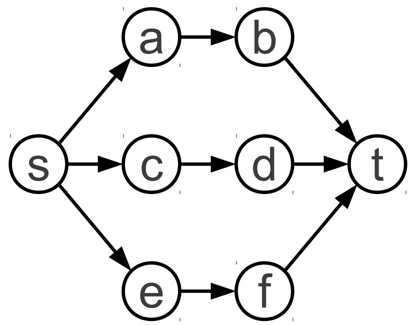

Figure 3: Input graphs G for Theorems 3.3.

the elements of $\mathcal{P}_\ell$. In contrast, Theorem 3.4 shows that some subsets of $\mathcal{P}_\ell$ can be accepted by much smaller sound monotone switching networks.

Theorem 3.4 (Potechin [4])**.** *Let G be a graph for which there is a single path of length ℓ from s to t, and any other vertex v of the graph is on an edge of the form $s \to v$ or $v \to t$. Then*

$$m(\sigma(G)) = n^{\Theta(1)} \ell^{\Theta(\lg \ell)}.$$

Figure 4 depicts an example of G with a single path from s to t where all other vertices are directly connected to s or t. Graphs described in Theorem 3.4 are simpler for monotone computation to recognize than the graphs in Theorem 3.3 because the value of $m(\sigma(G))$ is asymptotically much smaller.

Theorem 3.5 (Potechin [4])**.** *Consider a directed graph G with c vertices. Let ℓ be the length of a path from s to t. Add $n - c$ vertices to G to make n vertices total so that for each vertex v added, there exists an edge of the form $s \to v$ or $v \to t$, connecting v to the rest of G. Then,*

$$m(\sigma(G)) \leq n^{O(1)} c^{O(\lg \ell)}.$$

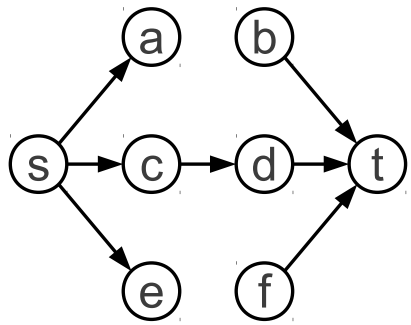

Figure 4: Input graphs G for Theorem 3.4.

In the case G is only a path, Theorem 3.5 is equivalent to Theorem 3.4. We use Theorem 3.5 when constructing upper bounds in the proofs of Theorems 4.2 and 4.4.

3.2 Techniques for Bounding $m(\sigma(G))$

In addition to the bounds for specific graphs in Section 3.1, Potechin [4] provides additional results which aid in the process of determining $m(\sigma(G))$ for arbitrary graphs G. Proofs of these results can be found in Appendix B.

Theorem 3.6. *Consider two graphs G and H such that every edge of G is also an edge of H. If a monotone switching network G' accepts G, then G' also accepts H.*

Corollary 3.7. *Given a directed graph G, consider the directed graph H which results from adding an edge to G. Then, $m(\sigma(G)) \geq m(\sigma(H))$. Conversely, if an edge is removed from G to yield $\bar{H}$, then $m(\sigma(G)) \leq m(\sigma(\bar{H}))$.*

From Theorem 3.6 and its corollary, we infer that for graphs with the same number of vertices the ones with more edges typically have smaller monotone switching networks.

Theorem 3.8. *Given a directed graph G with an edge $a \to b$, let H be the graph where this edge is replaced with $s \to b$ and $\bar{H}$ be the graph where this edge is replaced with $a \to t$. Then*

$$m(\sigma(G)) \geq \max(m(\sigma(H)),\ m(\sigma(\bar{H}))).$$

Heuristically, Theorem 3.8 implies that when the number of vertices and edges is the same for two graphs, the one with more edges connected from s or to t typically has a smaller monotone switching network.

Although Theorem 3.8 implies that more connectivity from s or to t means more efficiency, Theorem 3.9 shows what happens when there are similar edges but in the reverse direction.

Definition 3.1. An edge is ***useless*** if it is of the form $v \to s$ or $t \to v$ for some vertex v.

This definition is motivated by the fact that having an edge of this form gives no information about whether there is a path from s to t.

Theorem 3.9. *Let G be a graph with useless edges. Let H be G with the useless edges removed. Then $m(\sigma(G)) = m(\sigma(H))$.*

Theorem 3.9 is very useful for proving lower bounds of $m(\sigma(G))$, especially when combined with Theorem 3.10. To introduce this theorem, we first define the concept of a *merge graph*. Figure 5 depicts an example of a merge graph.

Definition 3.2. Given a graph G, let S be a set of vertices such that $s \in S$ and $t \notin S$. Also let T be a set of vertices such that $t \in T$ and $s \notin T$. Consider the graph $G_{(S,T)}$ whose vertex set is identical to G except the vertices of S have been merged into a single vertex $\bar{s}$, and the vertices of T have been merged into a single vertex $\bar{t}$. Any edge between two elements of S or two elements of T is removed. Any edge with exactly one endpoint in S is replaced with an edge whose corresponding endpoint is $\bar{s}$. Likewise, any edge with exactly one endpoint in T is replaced with an edge whose corresponding endpoint is $\bar{t}$. Any remaining edges are kept unchanged. Let $G_{(S,T)}$ be the (S,T)***-merge graph*** of G.

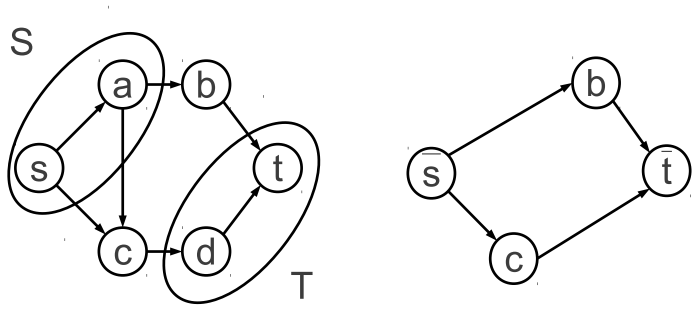

Figure 5: Example showing G and the merge graph $G_{(S,T)}$.

Theorem 3.10. *Given a graph G, let S and T be subsets of G defined as in Definition 3.2. Then $m(\sigma(G)) \geq m(\sigma(G_{(S,T)}))$.*

Theorems 3.9 and 3.10 can work together well, as noted by Potechin [4], to obtain bounds for arbitrary graphs.

A natural question after observing Theorem 3.10 is whether merging arbitrary sets of vertices which do not necessarily contain s or t provides the same result. Potechin [4] showed that this is not the case: there exists a graph G where contracting particular sets of vertices results in an increase in the value of $m(\sigma(G))$.

4 Main Results

We now demonstrate bounds on the value of $m(\sigma(G))$ where G is any tree. First, we find bounds in the case where G is a flow-out tree, whose bounds are the foundation of our proof of the general case.

Definition 4.1. A ***flow-out tree*** G is a tree with a special vertex r such that there is a path from r to every other vertex of G.

Before we explicitly bound $m(\sigma(G))$ in the case of flow-out trees, we state Lemma 4.1, a valuable tool for finding lower bounds.

Lemma 4.1. *Let G be a graph and H be a tree disconnected from G. Let n_H be the number of vertices of H. Let P be a path with $\lceil \sqrt{n_H} \rceil$ vertices which is disconnected from G. Then $m(\sigma(G \cup H)) \geq m(\sigma(G \cup P))$.*

The proof of Lemma 4.1 is given in Appendix C.

Theorem 4.2. *Let G be a flow-out tree with a path of length ℓ from s to t. Let s be the vertex with the property that there is a path from s to every other vertex of G. For $i \geq 1$, define d_i as the number of vertices whose descendants have a maximum distance of i from s. Define an additional sequence $c_1, c_2, \cdots, c_{\lceil \lg \lg \ell \rceil}$ with the property that $c_1 = n$ and for all $i \geq 2$,*

$$c_i = \sum_{j=2^{2^i}}^{n} d_j. \tag{1}$$

We then have that

$$\ell^{\Omega(\lg \ell)} \max_{1 \leq i \leq \lceil \lg \lg \ell \rceil} c_i^{\Omega(2^i)} \leq m(\sigma(G)) \leq n^{O(\lg \lg \ell)} \prod_{i=1}^{\lceil \lg \lg \ell \rceil} c_i^{O(2^i)}. \tag{2}$$

Note. These two bounds are not tight, but are within an exponent of $\lg \lg \ell$ times a constant of each other. This is because the lower bound has the maximum of $\lceil \lg \lg \ell \rceil$ terms, but the upper bound has the product of these terms.

Proof. We first show a lower bound for $m(\sigma(G))$ by proving that $\ell^{\Omega(\lg \ell)} c_i^{\Omega(2^i)} \leq m(\sigma(G))$ for each i. We show this by constructing a graph H, where $m(\sigma(H)) \leq m(\sigma(G))$ and H is a collection of internally disjoint paths of length 2^{2^i-1} from s to t with $c_i^{\Omega(1)}$ total vertices. The lower bound for $m(\sigma(G))$ then follows from Theorem 3.3.

After that, we show the upper bound for $m(\sigma(G))$ by constructing a series of graphs $G_1,\ G_2,\ \cdots,\ G_{\lceil \lg\lg \ell \rceil+1}$, and a recursive series of corresponding monotone switching networks $G'_1,\ G'_2,\ \cdots,\ G'_{\lceil \lg\lg \ell \rceil+1}$, where G'_i is a sound monotone switching network which accepts the elements of $\sigma(G_i)$. A full proof is given in Appendix D. $\square$

A dual structure to the flow-out tree is the flow-in tree.

Definition 4.2. A ***flow-in tree*** G is a tree with a special vertex r such that there is a path to r from every other vertex of G.

Theorem 4.2 has the following corollary which states an analogous bound for flow-in trees.

Corollary 4.3. *Let G be a flow-in tree with a path of length ℓ from s to t. Let t be the vertex with the property that there is a path to it from every other vertex of G. For $i \geq 1$, define d_i as the number of vertices whose ancestors, including itself, have a maximum distance of i from s. Define an additional sequence $c_1,\ c_2,\ \cdots,\ c_{\lceil \lg\lg \ell \rceil}$ with the property that $c_1 = n$ and for all $i \geq 2$, the element c_i satisfies (1). Hence, $m(\sigma(G))$ satisfies the bounds (2).*

Proof. We reverse the direction of every edge, and swap the labels of s and t. The obtained tree satisfies the hypothesis of Theorem 4.2. Thus, the same bound holds. $\square$

Theorem 4.4. *Let G be an arbitrary directed tree with a path from s to t. Define d_i^s to be the number of vertices which are accessible from s with i as the maximum distance of its descendants from s. Let d_i^t to be the number of vertices which can access t with the maximum distance of its ancestors to t being i. Define the sequence $c_1^s, \ldots, c_{\lceil \lg\lg \ell \rceil}^s$ such that*

$$c_1^s = \sum_{i=1}^{n} d_i^s \qquad \text{and} \qquad c_k^s = \sum_{i=2^{2^k}}^{n} d_i^s,\ (k \geq 2).$$

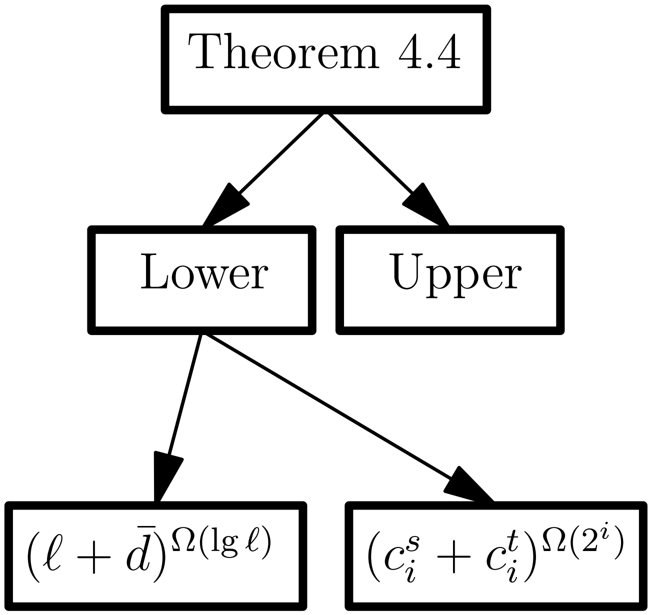

Figure 6: Flow chart of Proof of Theorem 4.4.

Define $c_1^t, \ldots, c_{\lceil \lg \lg \ell \rceil}^t$ *similarly. Let* $\bar{d}$ *be the number of vertices which are not accessible from* s *or* t*. Let* ℓ *be the length of the path from* s *to* t*. Then* $m(\sigma(G))$ *can be bounded by*

$$(\ell + \bar{d})^{\Omega(\lg \ell)} \max_{1 \leq i \leq \lceil \lg \lg \ell \rceil} (c_i^s + c_i^t)^{\Omega(2^i)} \leq m(\sigma(G)) \leq n^{O(\lg \lg \ell)} (\ell + \bar{d})^{O(\lg \ell)} \prod_{i=1}^{\lceil \lg \lg \ell \rceil} (c_i^s + c_i^t)^{O(2^i)}.$$

Note. These bounds are within $\lg \lg \ell$ times a constant factor of each other.

Proof. Like the proof of Theorem 4.2, this proof is divided into two parts, the first is to prove the lower bound, and the second is to prove the upper bound. Refer to Figure 6 for a general outline of the proof.

For this graph G, let H_s be the graph induced by the set of vertices v for which a path from s to v exists. Let H_t be the graph induced by the set of vertices v from which a path from v to t exists. In both cases, we do not include the vertices on the path from s to t.

Lower Bound:

The proof of the lower bound is divided into two parts. The first is to show that $m(\sigma(G)) \geq (\ell + \bar{d})^{\Omega(\lg \ell)}$. The second is to show that $m(\sigma(G)) \geq (c_i^s + c_i^t)^{\Omega(2^i)}$ for all i. We then take the geometric mean of these two bounds.

For the first part, we can merge all of the vertices of H_s with s and all of the vertices of H_t with t. In the graph $G_{(H_s,H_t)}$, there may be edges not on the path from s to t which are connected to s or t. These edges are directed towards s or away from t, so they are useless. We can remove these useless edges to obtain a graph consisting of a single path from s to t and a collections of trees which are disconnected from the main path. Let the size of these k trees be $a_1,\ a_2,\ \ldots\ ,\ a_k$. Notice that $a_1 + a_2 + \cdots + a_k = \bar{d}$. From Lemma 4.1, we can reduce these trees to paths of length $\lceil \sqrt{a_1} \rceil,\ \lceil \sqrt{a_2} \rceil,\ \ldots\ ,\ \lceil \sqrt{a_k} \rceil$. We can link these paths into a long path of length at least

$$\sum_{i=1}^{k} \sqrt{a_i} \geq \sqrt{\bar{d}}.$$

We can then merge this long path with the path from s to t to create a collection of disjoint paths of length ℓ from s to t. The total number of vertices is at least $\sqrt{\bar{d}}/3 + \ell$. From Theorem 3.3 we get a lower bound of

$$m(\sigma(G)) \geq \left(\frac{\sqrt{\bar{d}}}{3} + \ell\right)^{\Omega(\lg \ell)} = (\bar{d} + \ell)^{\Omega(\lg \ell)},$$

as desired.

For the second part, merge all the vertices of H_t with t. Additionally, merge any subgraphs not on the path from s to t which are connected to H_t with t. Merge with t any subgraphs connected to H_s but are not on the path from s to t. We are then left with a flow-out tree. By Theorem 4.2, $m(\sigma(G)) \geq (c_i^s)^{\Omega(2^i)}$. Using a symmetric argument with a flow-in tree, by Corollary 4.3, we have $m(\sigma(G)) \geq (c_i^t)^{\Omega(2^i)}$. Thus, $m(\sigma(G)) \geq (c_i^s + c_i^t)^{\Omega(2^i)}$, as desired.

Upper Bound:

We now prove the upper bound for $m(\sigma(G))$. First, take the vertices not in H_s, H_t, or the path from s to t, and remove any edges connected to them. Thus, G is a flow-in tree, a flow-out tree, and a collection of disconnected vertices. We use a construction very similar to that used in the proof of Theorem 4.2.

Let $\bar{c}$ be the number of points which are at a distance of more than ℓ from s or to t. Construct a new graph G_1 where all the edges connected to these $\bar{c}$ points are removed. We now construct graphs $G_2, \ \ldots, \ G_{\lceil \lg \lg \ell \rceil + 1}$ inductively as follows. Given G_{i-1}, let P_i^s be the set of vertices which are at a distance of less than $2^{2^{i-1}}$ from s and do not have any children. Let P_i^t be the set of vertices which are at a distance of less than $2^{2^{i-1}}$ to t and do not have any ancestors. Also, add to P_i^s any vertices which are on the paths from s to P_i; and add to P_i^t any vertices which are on paths from P_i^t to t. To create G_i, remove the edges connecting P_i^s to G_{i-1} and add edges directly from s to P_i^s. Similarly, remove edges connected P_i^t to G_{i-1} and add edges directly to t. From the definition of $c_i^s + c_i^t$, there are $n - c_i^s - c_i^t$ vertices which are directly connected to s or t. Refer to Figure 9 in Appendix D, for an example of this construction.

In the graph $G_{\lceil \lg \lg \ell \rceil + 1}$, every vertex is directly connected via a single edge to s or t, on the path from s to t, or disconnected from the graph. From Theorem 3.5, there is a monotone switching network $G'_{\lceil \lg \lg \ell \rceil + 1}$ of size $n^{O(1)}(\ell + \bar{c} + \bar{d})^{O(\lg \ell)}$ accepting the elements of $\sigma(G_{\lceil \lg \lg \ell \rceil + 1})$.

Next, we construct inductively the sequence of sound monotone switching networks $G'_{\lceil \lg \lg \ell \rceil}, \ \ldots, \ G'_2, \ G'_1$ such that G'_i accepts the elements of $m(\sigma(G_i))$. In the monotone switching network G'_{i+1}, consider an edge with a label of the form $s \to a$ or $a \to t$. Consider a graph $\bar{G}_{i+1} \in \sigma(G_{i+1})$ which crosses that edge in its accepting path. The corresponding graph $\bar{G}_i$ may not be able to cross that edge because the edge was deleted and became a path of length at most $2^{2^{i-1}}$ in $\bar{G}_i$. Thus, to construct G'_i, we replace that edge with a monotone switching network which checks if there is a path of length at most $2^{2^{i-1}}$ from s to a or a to t. This monotone switching network will contain up to n distinct monotone switching networks which check each possible path length. From Theorem 3.5, the number of vertices in this subgraph is $n^{O(1)}(c_i^s + c_i^t + \bar{d})^{O(2^i)}$. Thus, the number of edges in this subgraph is at most

$$n^2 \left(n^{O(1)}(c_i^s + c_i^t + \bar{d})^{O(2^i)} \right)^2 = n^{O(1)}(c_i^s + c_i^t + \bar{d})^{O(2^i)}.$$

The number of edges of G_i' is at most

$$|E(G_i')| \leq |E(G_{i+1}')| n^{O(1)}(c_i^s + c_i^t + \bar{d})^{O(2^i)}.$$

Thus,

$$\begin{aligned} V(G') \leq E(G') \leq E(G_1') = E(G'_{\lceil \lg \lg \ell \rceil + 1}) &\prod_{i=1}^{\lceil \lg \lg \ell \rceil} n^{O(1)}(c_i^s + c_i^t + \bar{d})^{O(2^i)} \\ &= n^2 (n^{O(1)}(\ell + \bar{c} + \bar{d})^{O(\lg \ell)})^2 n^{O(\lg \lg \ell)} \prod_{i=1}^{\lceil \lg \lg \ell \rceil} (c_i^s + c_i^t + \bar{d})^{O(2^i)} \\ &= n^{O(\lg \lg \ell)}(\ell + \bar{c} + \bar{d})^{O(\lg \ell)} \prod_{i=1}^{\lceil \lg \lg \ell \rceil} (c_i^s + c_i^t + \bar{d})^{O(2^i)}. \end{aligned} \tag{3}$$

Because $\bar{c}^{O(\lg \ell)} \leq (c_i^s + c_i^t + \bar{d})^{O(2^i)}$ when $i = \lceil \lg \lg \ell \rceil$, and

$$\prod_{i=1}^{\lceil \lg \lg \ell \rceil} \bar{d}^{O(2^i)} \leq \bar{d}^{O(\lg \ell)},$$

Inequality (3) is equivalent to

$$m(\sigma(G)) \leq n^{O(\lg \lg \ell)}(\ell + \bar{d})^{O(\lg \ell)} \prod_{i=1}^{\lceil \lg \lg \ell \rceil} (c_i^s + c_i^t)^{O(2^i)}.$$

□

5 Conclusion

Sound monotone switching networks provide an insightful way of analyzing monotone computation. Previously, Potechin [2, 3, 4] found tight bounds in the case where the inputs were the permutation sets of very specific kinds of trees and acyclic graphs. From these earlier results, we proved in Theorem 4.4 nearly tight bounds for all directed trees. These bounds give us insight into the structure of efficient monotone computation. From Theorem 4.4, the exponent for $c_1^s + c_1^t$ is orders of magnitude smaller than the exponent for $c_{\lceil \lg \lg \ell \rceil}^s + c_{\lceil \lg \lg \ell \rceil}^t$. We can infer from this that monotone computation is more effective at analyzing vertices closer to s and t than vertices which are farther. This suggests that the optimal algorithm for ST-connectivity in a monotone computation model is akin to a breadth-first search.

Future research will involve the following:

- Generalize the bounds to permutation sets of all acyclic graphs and eventually all graphs.

- Improve known bounds. Currently, these bounds are within a factor of $O(\lg \lg \ell)$ in the exponent. Can this be improved to a factor of $O(1)$ in the exponent?

- Find algorithms corresponding to these monotone switching networks. The existence of a monotone switching network of size m heuristically implies that an algorithm with $O(\log m)$ memory use exists, but such an algorithm may not necessarily exist [4]. Much work can be devoted to determining whether or not these algorithms exist and finding elegant implementations if they indeed exist.

- Extend these results to non-monotone switching networks. These more general structures account for all possible classical computations. According to Potechin [2, 4], obtaining tight bounds in this case would result in significant progress toward solving the open log-space versus nondeterministic log-space problem.

6 Acknowledgments

I would like to express my sincerest gratitude to my mentor Mr. Aaron Potechin for his guidance while undergoing this research. I would also like to thank Dr. Tanya Khovanova for her helpful suggestions pertaining to this project; Dr. Pavel Etingof for organizing the RSI mathematics projects; Dr. John Rickert, Sitan Chen, and other RSI staff members and students for giving suggestions to improve my paper. I would also like to acknowledge the Massachusetts Institute of Technology for hosting the Research Science Institute; my sponsors Dr. and Mrs. Daniel Dahua Zheng and the American Mathematical Society for funding the Research Science Institute; D.E. Shaw & Co., L.P. for naming me a D.E. Shaw scholar; and the MIT Mathematics Department and the Center for Excellence in Education, for making this paper possible. Finally, I would like to thank my parents, Warren and Kathleen, for their support.

References

[1] N. Immerman. Nondeterministic space is closed under complementation. *SIAM J. Comput.*, 17(5):935–938, Oct. 1988.

[2] A. Potechin. Bounds on monotone switching networks for directed connectivity. *arXiv:0911.0664v4*.

[3] A. Potechin. Monotone switching networks for directed connectivity are strictly more powerful than certain-knowledge switching networks. *arXiv:1111.2127v1*.

[4] A. Potechin. Personal Conversation, July 2012.

[5] O. Reingold. Undirected st-connectivity in log-space. *STOC*, 2005.

[6] W. J. Savitch. Relaionship between nondeterministic and deterministic tape classes. *J.CSS*, 4:177–192, 1970.

A Background

This appendix is a listing of terminology and notations not defined in the main body of the paper. This appendix is divided into two sections. The first, Graph Theory, formally defines what a graph is and some related concepts. The second, Big O, Ω, and Θ notations, gives formal definitions of these notations.

A.1 Graph Theory

Because monotone switching networks are graphs, much graph theory is used throughout this paper. In our context, graphs come in two varieties, undirected and directed.

Definition A.1. An ***undirected graph*** G is a pair of sets V and E. The elements of E are two-element sets $\{a, b\}$ where $a, b \in V$. The elements of V are called ***vertices*** and the elements of E are called ***edges***.

Definition A.2. An ***directed graph*** G is a pair of sets V and E. The elements of E are two-element *ordered pairs* (a, b) where $a, b \in V$. The elements of V are called ***vertices*** and the elements of E are called ***edges***.

If there are multiple graphs mentioned in the same context, for each graph G we refer to the sets V and E as $V(G)$ and $E(G)$.

The key distinction between directed and undirected graphs is that the elements of each edge of a directed graph are ordered while the elements of each edge of an undirected graph are not.

Definition A.3. Consider two undirected graphs G and H. Graph G is a ***subgraph*** of H if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$. The definition of subgraph is identical in the case that G and H are directed.

Definition A.4. In an undirected graph G, define a ***path*** between two vertices a and b in G to be a sequence of vertices

$$a = v_0,\ v_1,\ v_2,\ \dots,\ v_{\ell-1},\ v_\ell = b$$

such that for all $i \in \{0, \dots, \ell - 1\}$, we have that $\{v_i, v_{i+1}\} \in G$.

In a directed graph G, there is a ***path*** from vertex a to vertex b if there exists a sequence of vertices

$$a = v_0,\ v_1,\ v_2,\ \dots,\ v_{\ell-1},\ v_\ell = b$$

such that for all $i \in \{0, \dots, \ell - 1\}$, we have that $(v_i, v_{i+1}) \in G$.

Notice that in a directed graph, there can be a path from a to b but no path from b to a.

Definition A.5. The **length** of a path is the number of edges along the path.

A type of graph which is investigated is trees.

Definition A.6. A ***tree*** is an undirected graph G such that for any two vertices $a, b \in V(G)$, there exists a unique path between those two vertices. A ***directed tree*** is a directed graph G whose corresponding undirected graph is a tree.

If the context makes it clear that we are discussing directed graphs, a directed tree may be referred to as a tree.

Definitions A.7 and A.8 quantify which vertices are considered accessible from a particular vertex in a directed graph.

Definition A.7. Given a vertex v of a directed graph G, the ***children*** of G is the set of vertices C such that for any $w \in C$, there is an edge from v to w. We define ***parents*** similarly, except the edges are in the reverse direction.

Definition A.8. Given a vertex v of a directed graph G, the ***descendants*** of G is the set of vertices D such that for any $w \in D$, there is a path from v to w. We define ***ancestors*** similarly.

A.2 Big O, Ω, and Θ notations

Big O, Ω, and Θ notations are used to present the results of this paper. Understanding these notations is essential to understanding Section 3 and 4. Roughly, these notations give one a bound on the relative growth rate of a function.

Definition A.9. Given a function $f(n)$ from the positive integers to the positive reals, let $O(f(n))$ be the set of functions from the positive integers to the positive reals such that for any $g(n) \in O(f(n))$, there exists a constant $c > 0$ and an integer m such that $g(n) \leq cf(n)$ for all $n \geq m$. Instead of writing $g(n) \in O(f(n))$, mathematicians typically write $g(n) = O(f(n))$.

Definition A.10. Given a function $f(n)$ from the positive integers to the positive reals, let $\Omega(f(n))$ be the set of functions from the the positive integers to the positive reals such that for any $g(n) \in \Omega(f(n))$, there exists a constant $c > 0$ and an integer m such that $g(n) \geq cf(n)$ for all $n \geq m$. Again, $g(n) = \Omega(f(n))$ is preferred notation to $g(n) \in \Omega(f(n))$.

Definition A.11. Given a function $f(n)$ from the positive integers to the positive reals, let $\Theta(f(n))$ be the set of functions $g(n)$ from the the positive integers to the positive reals such that $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$.

Saying $g(n) = O(f(n))$ connotes roughly that g grows at most as much as f does. Similarly, connotes $g(n) = \Omega(f(n))$ means roughly that g grows at least as much as f does. Saying $g(n) = \Theta(f(n))$ implies that g grows about the same as f does.

B Proofs of Previous Results from Section 3

For the readers interest, this Appendix gives elementary proofs of some of the results in Section 3. These proofs are based on the work of Potechin [4].

Proof of Theorem 3.6. Since G' accepts G, there is a path from s' to t' such that for any edge on this path, its label is an edge of G. Since all of these edges are in H, we must have that H is accepted also. $\square$

Proof of Theorem 3.7. Consider a minimal-size sound monotone switching network G' which accepts all elements of $\sigma(G)$. For any element of $\sigma(G)$ there is a corresponding element of $\sigma(H)$ with the same edges. Hence, by Theorem 3.6, G' also accepts the elements of $\sigma(H)$. Therefore, $m(\sigma(G)) \geq m(\sigma(H))$. In the case where an edge is removed from G, replace G with $\bar{H}$ and H with G in the first inequality to obtain $m(\sigma(G)) \leq m(\sigma(\bar{H}))$. $\square$

Proof of Theorem 3.8. We prove that $m(\sigma(G)) \geq m(\sigma(H))$. The inequality $m(\sigma(G)) \geq m(\sigma(\bar{H}))$ is a symmetric argument by reversing every edge and swapping s and t. Consider the minimal-size sound monotone switching network G' which accepts the elements of $\sigma(G)$. Construct a new monotone switching network H' which contains the same vertices and edges as G'. For any edge e with a label $v_1 \to v_2$ in H', add an additional edge, parallel to e, with the label $s \to v_2$. We now prove two properties about H'.

Lemma B.1. *Every element of $\sigma(H)$ is accepted by H'.*

Proof. Let H_1 be an element of $\sigma(H)$. Let G_1 be a corresponding element of $\sigma(G)$. Because G_1 is accepted by G', there exists a path from s' to t' using only edge labels of G_1. Since no edges were deleted in the construction of H, this same path exists in H. On the other hand, some of the edge labels may not be edges of H. Let $v_1 \to v_2$ be one of these edges. Because $v_1 \to v_2$ is an edge of G_1, we must have that $s \to v_2$ is an edge of H. Consider the path in H' which bypasses the edge $v_1 \to v_2$ and instead passes through the parallel edge $s \to v_2$.

If this modification is made for all necessary edges, then we have an accepting path for H_1. Thus, all the elements of $\sigma(H)$ are accepted by H'. $\square$

Lemma B.2. *H' is sound.*

Proof. Assume for sake of contradiction that H' is not sound. Then there exists a graph K with no path from s to t but is accepted by H'. This implies there is a path P from s' to t' in H' which uses only edge labels in K. Because G' is sound, it rejects K. Thus, some of the edges of P must not be edges of G'. Hence, these are edges of the form $s \to v_2$ which are parallel with edges of the form $v_1 \to v_2$. Consider the new graph $\bar{K}$ where these edges of the form $s \to v_2$ are reverted to $v_1 \to v_2$. Because these original edge labels are in G', this new graph is accepted by G'. Since G' is sound, there exists a path from s to t using only the edges of $\bar{K}$. Let S be the set of points reachable from s in K, and let $\bar{S}$ be the set of points reachable from s in $\bar{K}$. Because there is no path from s to t in K, we have that $t \notin S$. Also note that each edge replacement does not add elements to S, as it replaces the condition 'v_2 is reachable from s' with 'v_2 is reachable from s if v_1 is reachable from s'. Thus, $t \in \bar{S} \subseteq S \not\ni t$. This is a clear contradiction, so H' is sound. $\square$

Since there exists a sound monotone switching network H' of size $m(\sigma(G))$ which accepts every element of $\sigma(H)$, we have that

$$m(\sigma(G)) \geq m(\sigma(H))$$

as desired. $\square$

Proof of Theorem 3.9. This result follows from Lemmas B.3 and B.4.

Lemma B.3. *Let G be a graph with the edge $a \to s$. Consider H, an identical graph except $a \to s$ is removed. Then $m(\sigma(G)) = m(\sigma(H))$.*

Proof. Since an edge was removed from G to yield H, from Theorem 3.7, we have that $m(\sigma(G)) \leq m(\sigma(H))$. Consider a sound monotone switching network G' of minimal size which accepts all the elements of $\sigma(G)$. Replace every edge of the form $a \to s$, for some a, with an unlabeled edge. An unlabeled edge can be traversed under any condition. This monotone switching network accepts all the elements of $\sigma(H)$. □

Lemma B.4. *Let G be a graph with the edge $t \to a$. Consider H, an identical graph except $t \to a$ is removed. Then $m(\sigma(G)) = m(\sigma(H))$.*

Proof. Since an edge was removed from G to yield H, from Theorem 3.7, we have that $m(\sigma(G)) \leq m(\sigma(H))$. Consider a sound monotone switching network G' of minimal size which accepts all the elements of $\sigma(G)$. Replace every edge of the form $t \to a$, for some a, with an unlabeled edge. This monotone switching network accepts all the elements of $\sigma(H)$. □

□

Proof of Theorem 3.10. Because $\sigma_{S\cup T}(G) \subset \sigma(G)$, we have that $m(\sigma_{S\cup T}(G)) \leq m(\sigma(G))$. Consider the sound monotone switching network H' which accepts all of the elements of $\sigma_{S\cup T}(G)$. Construct a new monotone switching network $\bar{H}'$ by taking every edge label of this monotone switching network which has an endpoint in S or T and replace it with $\bar{s}$ or $\bar{t}$, respectively. This new monotone switching network accepts all elements of $\sigma(G_{(S,T)})$. As any graph without a path from $\bar{s}$ to $\bar{t}$ cannot correspond to a graph with a path from s to t, we have that $\bar{H}'$ is sound. Thus,

$$m(\sigma(G_{(S,T)})) \leq m(\sigma_{S\cup T}(G)) \leq m(\sigma(G)).$$

□

C Proof of Lemma 4.1

Proof. Give the vertices of H a depth labeling $d(v)$ in the following way: pick an arbitrary vertex v and let $d(v)$ be 0. We can now define d recursively. For any edge $v_1 \to v_2$, we have $d(v_2) - d(v_1) = 1$. If there is a path from vertex w_1 to vertex w_2, then $d(w_2) - d(w_1) > 0$. Thus, any two vertices of the same depth do not have a directed path between them. Let $d_{\min}$ be the minimum depth and $d_{\max}$ be the maximum depth. Note that $d_{\min}$ may be negative if there are edges directed toward v. We now have two cases to consider.

Case 1: $d_{\max} - d_{\min} + 1 < \lceil \sqrt{n_H} \rceil$.

By the pigeonhole principle, there exists a depth $\bar{d}$ which has at least $\lceil \sqrt{n_H} \rceil$ vertices. We can merge all of the vertices of depth less than $\bar{d}$ with t and all the vertices of depth greater than $\bar{d}$ with s. After removing useless edges, these vertices are isolated. We then add edges between these vertices to create a path of length $\lceil \sqrt{n_H} \rceil$, as desired. If there are additional vertices, they can be merged with s.

Case 2: $d_{\max} - d_{\min} + 1 \geq \lceil \sqrt{n_H} \rceil$.

Let $w_{\min}$ be a vertex at the depth $d_{\min}$ and let $w_{\max}$ be a vertex at the depth $d_{\max}$. The undirected path from $w_{\min}$ to $w_{\max}$ has length at least $d_{\max} - d_{\min} + 1$. The number of edges along this path directed towards $w_{\max}$ is $d_{\max} - d_{\min} + 1$ plus the number of edges directed away from $w_{\max}$. For any vertex v where there are two vertices directed towards v, merge v with s. In the case where there are two vertices directed away from v, merge v with t. After removing useless edges, there are at least $\lceil \sqrt{n_H} \rceil$ vertices on disjoint paths. Edges can then be added between these paths to make a path of length at least $\lceil \sqrt{n_H} \rceil$, as desired. Any additional vertices can be merged with s. $\square$

D Proof of Theorem 4.2

Proof. To prove the lower bound, We show that for all $i \leq \lceil \lg \lg \ell \rceil$,

$$m(\sigma(G)) \geq \ell^{\Omega(\lg \ell)} c_i^{\Omega(2^i)}.$$

Consider the set of vertices which are not on the path from s to t. If we merge these vertices with s and remove useless edges directed toward s, we are left with a single path of length ℓ. From Theorem 3.10 and Theorem 3.3, we obtain

$$m(\sigma(G)) \geq \ell^{\Omega(\lg \ell)}.$$

Thus, it is sufficient to prove that

$$m(\sigma(G)) \geq c_i^{\Omega(2^i)}$$

and take the geometric mean of these two bounds. If $i = 1$, take every edge $a \to b$ not on the path from s to t and change it to $s \to b$. By Theorems 3.4 and 3.8, we get

$$m(\sigma(G) \geq n^{\Omega(1)} \ell^{\Omega(\lg \ell)} \geq c_1^{\Omega(2^1)}$$

Now consider $i \geq 2$, let $k = 2^{2^i}$. Let S_1 be the set of vertices v such for any vertex w which is a descendant of v, the distance from s to w is less than k. Merge the elements of S_1 with s. The remaining vertices have descendants which are a distance of at least k from s. Thus, there are exactly c_i vertices remaining by definition of c_i.

We split the problem into two cases. Let $\bar{d}_k$ be the number of vertices with a depth at most $\lceil k/2 \rceil$.

Case 1: $\bar{d}_k \geq \sqrt{c_i}$.

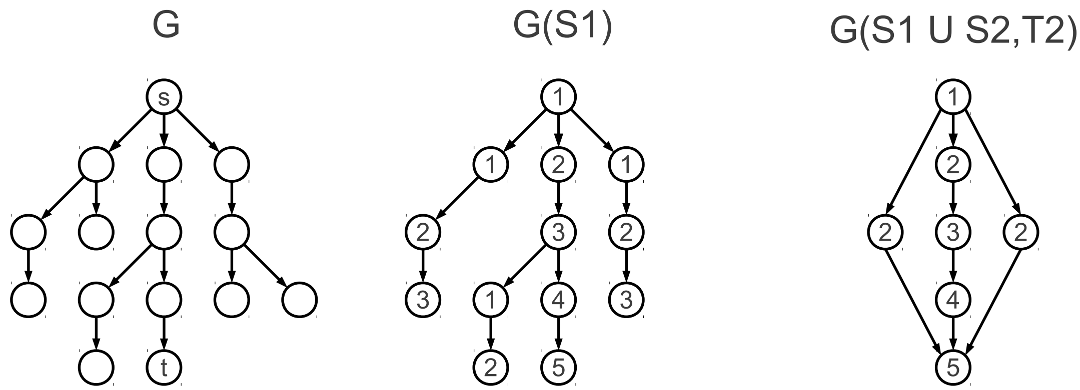

Figure 7: Examples of G, $G_{(S_1,\{\})}$, and $G_{(S_1\cup S_2,T_2)}$. The numbers written inside the vertices are values of f. In this case, $G_{(S_1\cup S_2,T_2)} = G_{(S_1\cup S_2\cup S_3,T_2)}$.

Define a function f from the set of remaining vertices to the positive integers as follows

(a) For the path from s to t defined by

$$s = v_0 \to v_1 \to \cdots \to v_{\ell-1} \to v_\ell = t,$$

$f(v_i) = i+1$. If v_i has p children $v_{i+1}, w_1, \ldots, w_{p-1}$, then $f(w_j) = 1$ for all positive integers j at most $p-1$.

(b) If v has j children $w_1, \ldots, w_j$, and is not on the path from s to t, then arbitrarily pick one child w_p so that it satisfies $f(w_p) = f(v)+1$. For all $q \neq p$, we have that $f(w_q) = 1$.

Let S_2 be the set of vertices v where $f(v) = 1$. Let T_2 be the set of vertices v where $f(v) \geq \lceil k/2 \rceil$. Merge the elements of S_2 with s and the elements of T_2 with t. By the nature of f, every path from s to t must have length at least $\lceil k/2 \rceil$. Let S_3 be the set of vertices from which there is no path to t. Merge those vertices with s.

Lemma D.1. *$G_{(S_1\cup S_2\cup S_3,T_2)}$ has at least $\bar{d}_k/2$ vertices.*

Proof. Let $\bar{d}'_k$ be the number of vertices v at a depth of at most $\lceil k/2 \rceil$ such that $f(v) = 1$. The remaining $\bar{d}_k - \bar{d}'_k$ vertices are not merged with s and are in $V(G_{(S_1 \cup S_2 \cup S_3, T_2)})$. Additionally, for any v such that $f(v) = 1$, there is an adjacent vertex w such that $f(w) = 2$ and w is in $V(G_{(S_1 \cup S_2 \cup S_3, T_2)})$. Thus,

$$|V(G_{(S_1 \cup S_2 \cup S_3, T_2)})| \geq \max(\bar{d}'_k, \bar{d}_k - \bar{d}'_k) \geq \frac{\bar{d}_k}{2},$$

as desired. □

Thus, from Theorem 3.10, Theorem 3.3, and Lemma D.1, we obtain

$$m(\sigma(G)) \geq m(\sigma(G_{(S_1 \cup S_2 \cup S_3, T_2)})) = \left(\frac{\bar{d}_k}{2}\right)^{\Omega(\lg(k/2))} = (c_i)^{\Omega(2^i)}.$$

Case 2: $\bar{d}_k < \sqrt{c_i}$.

Let $D_{\lceil k/2 \rceil}$ be the set of points at a distance of $\lceil k/2 \rceil$ from s. Let $v_{\lceil k/2 \rceil}$ be the element of $D_{\lceil k/2 \rceil}$ with the largest subtree H. Merge $v_{\lceil k/2 \rceil}$ with the vertex t. We now have two subcases to consider.

Subcase 1: t is in H. Merge the path from $v_{\lceil k/2 \rceil}$ to t with t. There may now be subtrees disconnected from G.

Subcase 2: t is not in H. Let w be the vertex which is at a distance of $\lceil k/2 \rceil$ away from t. Merge w with s.

We now treat both subcases identically. Let S_1 be the set of vertices which are not in H and are not on a path from s to t. Merge S_1 with s. Because H is the largest subtree, it has at least

$$\frac{c_i}{\bar{d}_k} \geq \sqrt{c_i}$$

vertices. Some of these $\ell - k/2$ vertices were removed in the subcase where $t \in H$, so G has at least $\sqrt{c_i} - \ell$ vertices.

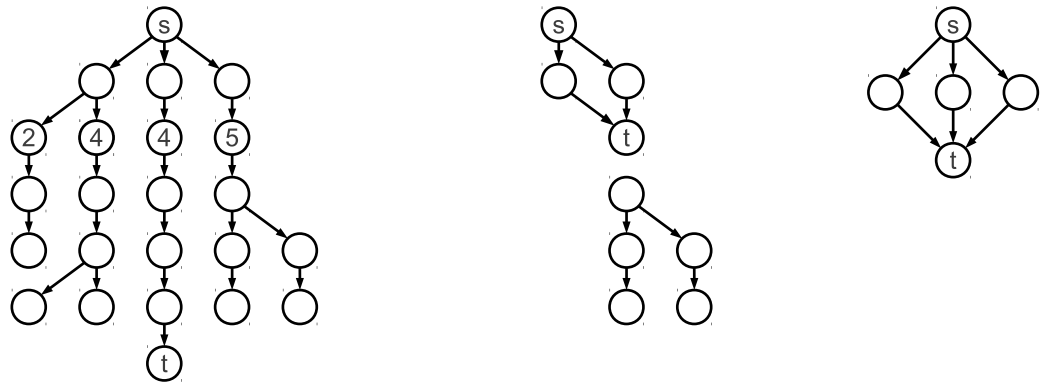

Figure 8: Example of the manipulation of G in Case 2, Subcase 2 of the proof of Theorem 4.2. In the first graph, the numbered vertices are the sizes of their subtrees. Notice that $k = 4$.

After removing useless edges from the mergings, we have a collection of disconnected trees. Let j be the number of such trees and let the sizes of these trees be

$$a_1, a_2, \ldots, a_j.$$

By Lemma 4.1, we can reduce these trees to paths of length

$$\lceil\sqrt{a_1}\rceil,\ \lceil\sqrt{a_2}\rceil,\ \ldots,\ \lceil\sqrt{a_j}\rceil.$$

We can then merge these paths with s and t to get a collection of paths from s to t of length $\lceil k/2\rceil$ containing at least

$$\frac{\sqrt{\sqrt{c_i} - \ell}}{3}$$

vertices. From Theorem 3.3, we get a lower bound of

$$m(\sigma(G)) = \left(\frac{\sqrt{\sqrt{c_i} - \ell}}{3}\right)^{\Omega(\lg k/2)} = (\sqrt{c_i} - \ell)^{\Omega(\lg k)}.$$

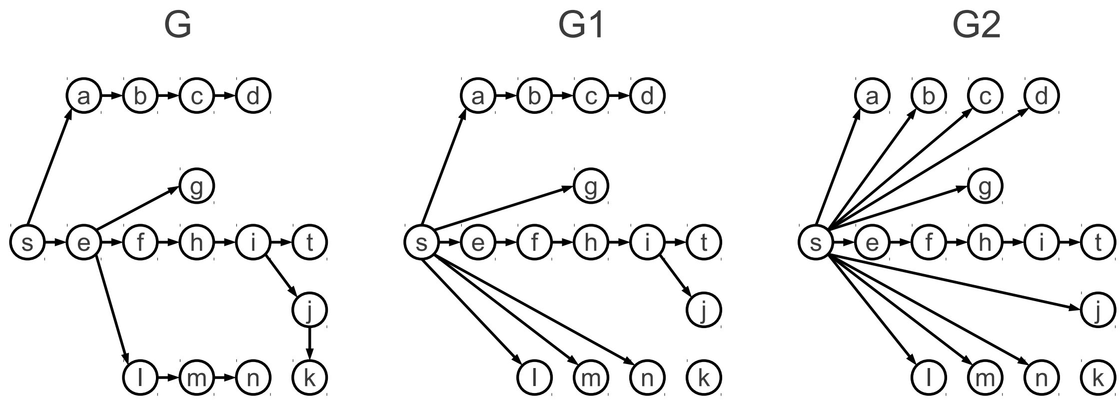

Figure 9: Sequence of graphs G, G_1, G_2 used in proving the upper bound in Theorem 4.2.

If $\sqrt{c_i} \geq 2\ell$, then $m(\sigma(G)) = c_i^{\Omega(\lg k)}$. Otherwise, $m(\sigma(G)) = \ell^{\Omega(\lg \ell)}$ is a better lower bound as $\ell \geq k$.

This completes our proof of the lower bound.

To prove the upper bound on the value of $m(\sigma(G))$, we need to construct a sound monotone switching network which accepts all the elements of $\sigma(G)$.

Let $\bar{c}$ be the number of points which are at a distance of more than ℓ from s. Construct a new graph G_1 where all the edges connected to these $\bar{c}$ points are removed. We next construct the graphs $G_2, \ldots, G_{\lceil \lg \lg \ell \rceil + 1}$ inductively as follows. Let P_i be the set of vertices which are at a distance less than $2^{2^{i-1}}$ from s which do not have any children. Also add to P_i any vertices which are on the paths from s to P_i. To create G_i, remove the edges connecting P_i to G_{i-1} and add edges directly from s to P_i. From the definition of c_i, there are $n - c_i$ vertices which are directly connected to s. See Figure 9 for an example of this construction.

In the graph $G_{\lceil \lg \lg \ell \rceil + 1}$, every vertex is directly connected via a single edge to s, or on the path from s to t, or disconnected from the graph. From Theorem 3.5, there is a monotone switching network $G'_{\lceil \lg\lg \ell \rceil+1}$ of size

$$n^{O(1)}(\ell + \bar{c})^{O(\lg \ell)}$$

accepting the elements of $\sigma(G_{\lceil \lg\lg \ell \rceil+1})$.

We now construct inductively the sequence of sound monotone switching networks

$$G'_{\lceil \lg\lg \ell \rceil},\ \ \ldots,\ G'_2,\ G'_1$$

such that G'_i accepts the elements of $\sigma(G_i)$. In the monotone switching network G'_{i+1}, consider an edge with a label of the form $s \to a$. Consider a graph $\bar{G}_{i+1} \in \sigma(G_{i+1})$ which crosses that edge in its accepting path. The corresponding graph $\bar{G}_i$ may not be able to cross that edge because the edge in $\bar{G}_{i+1}$ turned into a path of length at most $2^{2^{i-1}}$ in $\bar{G}_i$. Thus, to construct G'_i, we replace that edge with a monotone switching network which checks if there is a path of length at most $2^{2^{i-1}}$ from s to a. This monotone switching network will contain up to n distinct monotone switching networks which check each possible path length. From Theorem 3.5, the number of vertices in this subgraph is $n^{O(1)}c_i^{O(2^i)}$. Thus, the number of edges in this subgraph is at most

$$n^2\left(n^{O(1)}c_i^{O(2^i)}\right)^2 = n^{O(1)}c_i^{O(2^i)}.$$

The number of edges of G'_i is at most

$$|E(G'_i)| \leq |E(G'_{i+1})| n^{O(1)}c_i^{O(2^i)}.$$

Thus,

$$\begin{aligned}
|V(G')| \le |E(G')| \le |E(G'_1)| &= |E(G'_{\lceil \lg\lg \ell \rceil + 1})| \prod_{i=1}^{\lceil \lg\lg \ell \rceil} n^{O(1)} c_i^{O(2^i)} \\
&= n^2 (n^{O(1)} (\ell + \bar{c})^{O(\lg \ell)})^2 n^{O(\lg\lg \ell)} \prod_{i=1}^{\lceil \lg\lg \ell \rceil} c_i^{O(2^i)} \\
&= n^{O(\lg\lg \ell)} (\ell + \bar{c})^{O(\lg \ell)} \prod_{i=1}^{\lceil \lg\lg \ell \rceil} c_i^{O(2^i)}. \qquad (4)
\end{aligned}$$

Because $(\ell + \bar{c})^{O(\lg \ell)} \ge c_i^{O(2^i)}$ when $i = \lceil \lg\lg \ell \rceil$, Inequality (4) is equivalent to

$$m(\sigma(G)) = n^{O(\lg\lg \ell)} \prod_{i=1}^{\lceil \lg\lg \ell \rceil} c_i^{O(2^i)},$$

as desired. □