Pseudorandom Walks on Regular Digraphs and the $\mathbf{RL}$ vs. $\mathbf{L}$ Problem[*]

PRELIMINARY VERSION

Omer Reingold[†] Luca Trevisan[‡] Salil Vadhan[§]

November 3, 2005

Abstract

We revisit the general $\mathbf{RL}$ vs. $\mathbf{L}$ question, obtaining the following results.

1. Generalizing Reingold's techniques to directed graphs, we present a deterministic, log-space algorithm that given a *regular* (or, more generally, *Eulerian*) directed graph G and two vertices s and t, finds a path between s and t if one exists.
2. If we restrict ourselves to directed graphs that are regular and *consistently labelled*, then we are able to produce *pseudorandom walks* for such graphs in logarithmic space (this result already found an independent application).
3. We prove that if (2) could be generalized to all regular directed graphs (including ones that are not consistently labelled) then $\mathbf{L} = \mathbf{RL}$. We do so by exhibiting a new complete promise problem for $\mathbf{RL}$, and showing that such a problem can be solved in deterministic logarithmic space given a log-space pseudorandom walk generator for regular directed graphs.

We interpret (1) as indicating that it is not *reversibility* per se which Reingold's techniques rely upon, but rather the fact that, in the undirected S-T connectivity problem, the graph may be assumed to be *regular* without loss of generality. On the other hand, as far as derandomizing $\mathbf{RL}$ via pseudorandom walks goes, we obtain by (3) that one can assume regularity without loss of generality. In other words, for this purpose, it is not necessary to develop a theory of pseudorandomness for arbitrary directed graphs with unknown stationary distributions. The combination of (2) and (3) indicates that the only obstacle towards a full derandomization of $\mathbf{RL}$ is in handling arbitrary edge labels. It remains to be seen how difficult this challenge is to overcome.

[*]Earlier version appeared in the Electronic Colloquium on Computational Complexity (ECCC) as Technical Report TR05-022, February 2005. Research supported by US-Israel Binational Science Foundation Grant 2002246.

[†]Incumbent of the Walter and Elise Haas Career Development Chair, Department of Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. omer.reingold@weizmann.ac.il

[‡]Computer Science Division, University of California, Berkeley, CA, USA. Also supported by the National Science Foundation under grant CCR 0515231. luca@cs.berkeley.edu

[§]Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. Also supported by NSF grant CCR-0133096, ONR grant N00014-04-1-0478, and a Sloan Research Fellowship. salil@eecs.harvard.edu

1 Introduction

The research on derandomization of space-bounded computations deals with the tradeoff between two basic resources of computations: memory (or space) and randomness. Can randomness save space in computations? Alternatively, can every randomized algorithm be derandomized with only a small increase in space? These questions received the most attention in the context of log-space computations, and with respect to the following complexity classes: $\mathbf{L}$ (the class of problems solvable in deterministic log-space), $\mathbf{RL}$, and $\mathbf{BPL}$ (the classes of problems solvable by randomized log-space algorithms making one-sided and two-sided errors respectively). It is widely believed that $\mathbf{L} = \mathbf{RL} = \mathbf{BPL}$ and proving this conjecture is the ultimate goal of this body of research.

It turns out that the derandomization of $\mathbf{RL}$ is related to determining the space complexity of one of the most basic graph problems, UNDIRECTED S-T CONNECTIVITY: Given an undirected graph and two vertices, is there a path between the vertices? (The corresponding search problem is to find such a path). The space complexity of this problem and the derandomization of space-bounded computations have been the focus of a vast body of work, and brought about some of the most beautiful results in complexity theory. The connection between the two was made by Aleliunas et. al. [AKL$^+$], who gave an $\mathbf{RL}$ algorithm for UNDIRECTED S-T CONNECTIVITY. The algorithm simply runs a random walk from the first vertex s for a polynomial number of steps, and accepts if and only if the walk visits the second vertex t. This beautifully simple algorithm is undoubtedly one of the most interesting $\mathbf{RL}$ algorithms. It casts the space complexity of UNDIRECTED S-T CONNECTIVITY as a specific example and an interesting test case for the derandomization of space-bounded computations. (In particular, if $\mathbf{RL} = \mathbf{L}$, then UNDIRECTED S-T CONNECTIVITY can be solved in deterministic log-space.) Since then progress on the general and the specific problems alternated with a fluid exchange of ideas (as demonstrated by [Sav, AKS, BNS, Nis2, Nis1, NSW, SZ, ATSWZ], to mention just a few highlights of this research). See the surveys of Saks [Sak] and Wigderson [Wig] for more on these vibrant research areas.

The starting point of our research is a recent result of Reingold [Rei] that showed that UNDIRECTED S-T CONNECTIVITY has a deterministic log-space algorithm. On the other hand, the best deterministic space bound on $\mathbf{RL}$ in general remains $O(\log^{3/2} n)$, established by Saks and Zhou [SZ].

1.1 Our Results

In this paper, we revisit the general $\mathbf{RL}$ vs. $\mathbf{L}$ question in light of Reingold's results, and obtain the following results:

1. Generalizing Reingold's techniques to directed graphs (aka. *digraphs*), we present a deterministic, log-space algorithm that given a Eulerian digraph G (i.e. a graph such that each vertex has an outdegree equal to its indegree) and two vertices s and t, finds a path between s and t if one exists. This involves a new analysis of the zig-zag graph product of [RVW] that generalizes to regular digraphs and the directed analogue of the spectral gap, which may be of independent interest.

2. For the special case of "consistently labelled" regular digraphs we provide a "pseudorandom walk generator." A digraph is *regular* of degree D, or D-regular, if all vertices have indegree D and outdegree D; a D-regular digraph is *consistently labelled* if the D edges leaving each vertex are numbered from 1 to D in such a way that at each vertex, the labels of the incoming edges are all distinct. Roughly speaking, given a random seed of logarithmic length, our generator constructs, in log-space, a "short" pseudorandom walk that ends at an almost-uniformly distributed vertex when taken in any consistently-labelled regular digraph.

Pseudorandom generators that fool space-bounded computations, are very interesting tools even beyond the $\mathbf{RL}$ vs. $\mathbf{L}$ problem (see [Ind, Siv, HVV, HHR] for just a few of their applications). In particular, even the pseudorandom walks given in this paper already found an application in the construction of almost k-wise independent permutations [KNR]. Unfortunately, "oblivious" derandomizations are more difficult, due to the inability to look at the input. For example, while it is true that every regular digraph has a consistent labelling, it is not clear how to transform a pseudorandom generator that works for consistently-labelled regular digraphs such that it would also work for arbitrarily-labelled regular digraphs.

3. We prove that if our pseudorandom generator from Item 2 could be generalized to all regular digraphs (instead of just consistently labelled ones), then $\mathbf{RL} = \mathbf{L}$.

   We do so by exhibiting a new complete problem for $\mathbf{RL}$: S-T CONNECTIVITY restricted to digraphs for which the random walk is promised to have polynomial "mixing time," as measured by a directed analogue of the spectral gap introduced by Mihail [Mih]. We then show that a pseudorandom walk generator for regular digraphs can be used to solve our complete problem deterministically in logarithmic space.

1.2 Perspective

We now discuss possible interpretations of the aforementioned results for the derandomization of $\mathbf{RL}$.

First, let us consider Reingold's algorithm for undirected ST-connectivity. What are the properties of undirected graphs that are intrinsic to this algorithm? One property of undirected graphs is reversibility - a walk on the graph can immediately undo any of its steps by taking the last edge again (in the reverse direction). A second property is that the stationary distribution of the walk on an undirected graph is well behaved (the probability of a vertex is proportional to its degree), and such graphs can easily be reduced to regular graphs where the stationary distribution is uniform. Our result (1), where we extend the algorithm to Eulerian digraphs indicates that the latter property of undirected graphs is much more important here than the former. After all, Eulerian digraphs are non-reversible but their stationary distribution is well behaved and they can easily be reduced to regular digraphs where the stationary distribution is again uniform (the reduction is described in Section 5).

A "pseudorandom walk generator" that works for every consistently labelled regular *undirected* graph is implicit in [Rei]. (Actually, the generator requires a more restrictive form of labelling). Our result (2) formalizes this generator and shows a generalization to regular consistently-labelled digraphs. In order to get a general pseudo-random generator for space-bounded generators (which as mentioned above is a goal of independent interest) there are two restrictions to overcome: regularity and consistency of the labelling.

It is well known that every regular digraph has a consistent labelling . Furthermore, regularity already proved crucial in our result(1). It may therefore seem that the most stringent of the requirements in our construction is regularity rather than consistent labelling. Our final result (3) shows that in this context (of derandomization with pseudorandom walks) regularity is essentially irrelevant. Consistent labelling is in fact the only obstacle towards a full derandomization of $\mathbf{RL}$. It remains to be seen how difficult this is to overcome.

Why is consistent labelling so important? First, as we noted above, *in the context of pseudorandom walks* it is not clear how useful is the mere fact that consistent labelling *exists*. A pseudorandom walk is an operation that is oblivious to the particular input graph, but on the other hand, consistently labelling a graph may not be oblivious (and in fact seems rather "global"). Therefore, it is not clear how to transform a pseudorandom walk for regular consistently-labelled digraphs into one that is pseudorandom for general regular digraphs. An intuitive reason for the importance of the labelling is that for any fixed sequence of edge labels, the corresponding walk on a graph with consistent labels cannot lose entropy (the distribution of the final vertex has as much entropy as the distribution of the start vertex). On the other hand, without the assumption on the labelling, entropy losses may occur. Therefore progress made in one part of a pseudorandom walk (i.e. an increase in entropy) may be lost later in the same walk.

1.3 Techniques

The main technical step in the proof of our results (1) and (2) is an analysis of a zig-zag graph product [RVW] applied to regular digraphs. More specifically, we bound the *spectral gap* (as defined by Mihail [Mih] and Fill [Fil] in the context of nonreversible Markov chains) of the graph obtained by the zig-zag product of two regular digraphs. An analogous bound was proven in [RVW] for undirected regular graphs, but their proof is not immediately applicable to our setting because it uses properties of symmetric matrices. It turns out that our new analysis is actually simpler than the one in [RVW], even though it applies to a more general setting. The proof we present here is even simpler than the one that appeared in the preliminary version of this paper [RTV]. The new proof is based on an approach of Rozenman and Vadhan [RV], who used it to analyze a new 'derandomized squaring' operation.

Another contribution, that may be of independent interest is the new complete promise problem we present for $\mathbf{RL}$. Very loosely, this problem is st-connectivity in rapidly mixing Markov chains (where in the 'Yes' case, both nodes s and t have noticeable probability mass under the stationary distribution of walks starting at s). A complete problem for $\mathbf{RL}$ based on Markov chains was previously known (see the survey of Saks [Sak]). However, in that problem one examines the behavior of a walk at a particular time step t. On the other hand, in the new complete problem we discuss the behavior of the walk *in its limit* (i.e., we are interested in the stationary distribution). Such a problem seems much more amenable to the techniques of [Rei]. In particular, even in the undirected case, we do not know how to space-efficiently and deterministically simulate the distribution reached by a random walk after a fixed number of steps (unless this walk was long enough to approach the stationary distribution).

In the proof of our result (3), we define (as a mental experiment) a regular digraph which can be thought of as a "blow-up" of the input graph in the new complete promise problem for $\mathbf{RL}$. More specifically, every vertex in the input graph corresponds to a block of vertices in the blow-up graphs, with multiplicity that is linearly related to the weight of the original vertex under the stationary distribution. Intuitively, as heavy vertices are split into many more vertices in the blow-up graph, we indeed obtain a graph where the stationary distribution is uniform (and is therefore regular). We are not able to construct this blow-up graph efficiently but we can show (again as a mental experiment) that for some (inconsistent) labelling of the edges in the blow up graph a walk on the blow-up graph naturally "projects" onto the original graph. Furthermore, the projected walk can be easily and efficiently simulated by only referring to the original input graph. By assumption, we know how to generate pseudorandom walks for the blow-up graph and as we show, simulating the projection of such walks on the original graph is sufficient to solve the promise problem.

It is natural to attempt the general framework of derandomization studied here with a different measure of expansion (rather than analogues of eigenvalue gap). We also consider here the combinatorial measure of *edge expansion*. We show that edge expansion is preserved and degree is reduced, by taking a *replacement product* with an expander graph. We show, however, that edge expansion is not necessarily improved by powering in digraphs, and it is not clear that there is any other "local" operation that increases edge expansion. See Appendix A.9 for details.

1.4 Organization

We begin by defining notions of expansion for digraphs and giving other technical preliminaries in Section 2. We present in Section 3 our new $\mathbf{RL}$-complete promise problem. The operations of powering, replacement product and zig-zag graph product are defined for digraphs in Section 4, and the effect of powering and zig-zag product on regular digraphs is analyzed in Section 5, leading to our algorithm for finding paths in regular digraphs. Our construction of universal transversal sequences for regular consistently-labelled digraphs, and our pseudorandom walk generator for regular consistently-labelled digraphs are presented in Section 6. In Section 7 we prove that a pseudorandom walk generator for general regular digraphs would imply $\mathbf{L} = \mathbf{RL}$. We give a discussion of other measures of expansion in Appendix A.9.

2 Preliminaries

2.1 Graphs and Markov Chains

In this paper, we consider **directed graphs** (**digraphs** for short), and allow them to have multiple edges, and have self-loops. A graph is **out-regular** (resp., **in-regular**) if every vertex has the same number D of edges leaving it; D is called the **out-degree** (resp., **in-degree**). A graph is **regular** if it is both out-regular and in-regular.

Given a graph G on N vertices, we consider the random walk on G described by the transition matrix M_G whose (v, u)'th entry equals the number of edges from u to v, divided by the outdegree of v.[1]

More generally, if $M^{N\times N}$ is a matrix with non-negative entries such that for every $u \in [N]$ we have $\sum_v M(v,u) = 1$, then we say that M is a **Markov chain** on state space $[N]$. For a Markov chain $M^{N\times N}$, we define the **underlying graph** of M as the graph $G = ([N], E)$ such that $(u, v) \in E$ if and only if $M(v, u) > 0$. A distribution $\pi \in \mathbb{R}^N$ is **stationary** for a Markov chain M if $M\pi = \pi$. Note that if π is stationary for M, then $\mathrm{supp}(\pi) \stackrel{\mathrm{def}}{=} \{v : \pi(v) > 0\}$ is a closed subset of M in the sense that there are no transitions from $\mathrm{supp}(\pi)$ to its complement; thus M is well-defined as a Markov chain restricted to $\mathrm{supp}(\pi)$. A Markov chain M is **time reversible** with respect to a stationary distribution π if for every two vertices $u, v \in [N]$ we have $\pi(u)M(v, u) = \pi(v)M(u, v)$. If G is an undirected graph, then M_G is time reversible with respect to the stationary distribution $\pi(u) = d(u)/2m$, where $d(u)$ is the degree of u and m is the number of edges. A random walk on a directed graph, however, is typically not time reversible.

We are interested in the rate at which a Markov chain M converge to a stationary distribution For a time-reversible Markov chain M, it is well-known that the rate of convergence is characterized by the second largest (in absolute value) eigenvalue $\lambda_2(M)$ of the matrix M. If M is not time-reversible (for example, if M is the random walk on a directed graph), then M need not have real eigenvalues, and the stationary distribution need not have the largest eigenvalue in absolute value, so the time-reversible theory is not immediately applicable.

Following Mihail [Mih] and Fill [Fil], we introduce a parameter $\lambda(M)$ which is equal to $\lambda_2(M)$ if M is time-reversible, but that remains well-defined even for non-time-reversible Markov chain. For a probability distribution $\pi \in \mathbb{R}^N$ on vertices, we define a normalized inner product on $\mathbb{R}^N$ by:

$$\langle x, y\rangle_\pi \stackrel{\mathrm{def}}{=} \sum_{v\in \mathrm{supp}(\pi)} \frac{x(v)\cdot y(v)}{\pi(v)},$$

[1] Often the transition matrix is defined to be the transpose of our definition. Our choice means taking a random walk corresponds to *left*-multiplication by M_G.

and a norm $\|x\|_\pi \stackrel{\text{def}}{=} \sqrt{\langle x, x\rangle_\pi}$. Note that this normalization makes π itself a unit vector (i.e. $\|\pi\|_\pi = 1$), and also implies that x is orthogonal to π iff $\sum_v x(v) = 0$. (Technically, $\langle\cdot,\cdot,\rangle_\pi$ is only an inner product on the subspace $\{x \in \mathbb{R}^N : \mathrm{supp}(x) \subseteq \mathrm{supp}(\pi)\}$, since there are nonzero vectors x outside this subspace such that $\|x\|_\pi = 0$. However, it will be convenient to use this notation for arbitrary vectors in $\mathbb{R}^N$.)

Definition 2.1 *Let M be a Markov chain and π be a stationary distribution for M. We define the* **spectral expansion** *of M with respect to π to be*

$$\lambda_\pi(M) \stackrel{\text{def}}{=} \max_{x\in\mathbb{R}^N:\langle x,\pi\rangle_\pi=0} \frac{\|Mx\|_\pi}{\|x\|_\pi},$$

For a digraph G and a stationary distribution of M_G, we often write $\lambda_\pi(G)$ instead of $\lambda_\pi(M_G)$.

As noted above, when M is time-reversible, then $\lambda_\pi(M)$ equals the second largest eigenvalue (in absolute value) of M (more precisely, the submatrix of M consisting of the rows and columns in $\mathrm{supp}(\pi)$). In general, $\lambda_\pi(M)$ equals the square root of the second largest (in absolute value) of $\tilde{M}M$, where $\tilde{M}(u,v) = \pi(u)M(v,u)/\pi(v)$ (again, restricting to submatrices so that $u, v \in \mathrm{supp}(\pi)$).

The following lemma shows that if $\lambda_\pi(M)$ is small, then the Markov chain converges quickly to π.

Lemma 2.2 *Let π be a stationary distribution of Markov chain M on $[N]$, and let α be any distribution on $[N]$ such that* $\mathrm{supp}(\alpha) \subseteq \mathrm{supp}(\pi)$. *Then*

$$\|M^t\alpha - \pi\|_\pi \le \lambda_\pi(M)^t \cdot \|\alpha - \pi\|_\pi.$$

In particular, if we start at a vertex $v \in \mathrm{supp}(\pi)$ and run M for t steps, then we end at vertex $w \in \mathrm{supp}(\pi)$ with probability at least $\pi(w) - \lambda_\pi(M)^t \cdot \sqrt{\pi(w)/\pi(v)}$.

The above lemma refers to convergence in (normalized) ℓ_2 distance. The following lemma shows that this implies convergence in standard variation distance.

Lemma 2.3 *For any distribution α, the variation distance between α and π is at most $\|\alpha - \pi\|_\pi$.*

It is well-known that (connected, nonbipartite) undirected graphs G always satisfy $\lambda_\pi(G) \le 1 - 1/\mathrm{poly}(N, D)$, where N is the number of vertices and D the degree [Lov]. That is, undirected graphs have at most polynomial mixing time. However, in general directed graphs, $\lambda_\pi(G)$ can be exponentially close to 1, and thus the mixing time exponentially large.

Just as in the undirected case, the spectral expansion can be bounded in terms of the sizes of cuts in the underlying graph.

Definition 2.4 *Let M be a Markov chain with N vertices and π a stationary distribution. The* **conductance** *of M with respect to π is defined to be*

$$h_\pi(M) \stackrel{\text{def}}{=} = \min_{A:0<\pi(A)\le 1/2} \frac{\sum_{u\in A, v\notin A} \pi(u)M(v,u)}{\pi(A)}.$$

Lemma 2.5 ([SJ, Mih, Fil]) *Let M be a Markov chain on N vertices such that $M(u,u) \ge 1/2$ for every u (i.e. M is "strongly aperiodic"), and let π be a stationary distribution of M. Then $\lambda_\pi(M) \le 1 - h_\pi(M)^2/2$.*

When the stationary distribution π is uniform on the vertices of G, then the conductance defined above coincides exactly with the "edge expansion" of G, defined below.[2]

Definition 2.6 *Let $G = (V, E)$ be a directed graph in which every vertex has outdegree D. Then the* **edge expansion** *of G is defined to be*

$$\varepsilon(G) = \min_A \frac{E(A, \overline{A})}{D \cdot \min\{|A|, |\bar{A}|\}},$$

where the minimum is taken over sets of vertices A and $E(A, \overline{A})$ is the set of edges (u, v) where $u \in A$ and $v \notin A$.

2.2 Complexity Classes

We let $\mathbf{L}$, $\mathbf{RL}$, $\mathbf{NL}$, $\mathbf{BPL}$ denote the standard logspace complexity classes. We define $\mathbf{prL}$, $\mathbf{prRL}$ and $\mathbf{prBPL}$ as the respective classes of *promise problems* and $\mathbf{searchL}$, $\mathbf{searchRL}$ and $\mathbf{searchNL}$ as the respective classes of *search problems*. See Appendix A.2 for detailed definitions and for definitions of reductions between search problems. We note the following result.

Proposition 2.7 *If* $\mathbf{prBPL} = \mathbf{prL}$*, then* $\mathbf{searchRL} = \mathbf{searchL}$.

3 A New Complete Problem for $\mathbf{RL}$

S-T CONNECTIVITY and its search version, FIND PATH, both defined below, are two of the most basic problems in computer science.

S-T CONNECTIVITY:

- **Input:** (G, s, t), where $G = (V, E)$ is a directed graph, $s, t \in V$
- **YES instances:** There is a path from s to t in G.
- **NO instances:** There is no path from s to t in G.

FIND PATH:

- **Input:** (G, s, t), where $G = (V, E)$ is a directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **Promise:** There is a path from s to t in G.
- **Output:** A path from s to t in G.

It is well-known that S-T CONNECTIVITY is complete for $\mathbf{NL}$, and the same argument shows that FIND PATH is complete for $\mathbf{searchNL}$. Here we are interested in the complexity of restrictions of these problems. The recent result of Reingold [Rei] shows that their restrictions to *undirected* graphs, UNDIRECTED S-T CONNECTIVITY and UNDIRECTED FIND PATH, are in $\mathbf{L}$ and $\mathbf{searchL}$, respectively.

It was known (see [Sak]) that a certain restriction of S-T CONNECTIVITY was complete for $\mathbf{prRL}$, specifically one where we look at the probability that a random walk of a particular length goes from s to t:

[2]To see that $\varepsilon(G) = h_\pi(G)$ when π is the uniform distribution, note that the fact that the stationary distribution is uniform implies that G is biregular, which in turn implies that $E(A, \overline{A}) = E(\overline{A}, A)$.

SHORT-WALK S-T CONNECTIVITY:

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a directed graph, $s, t \in V$
- **YES instances:** A random walk of length k started from s ends at t with probability at least $1/2$.
- **NO instances:** There is no path from s to t in G.

However, this problem does not seem to capture the properties of UNDIRECTED S-T CONNECTIVITY used in Reingold's algorithm [Rei]. His algorithm uses relies on a measure of expansion, specifically the spectral gap, which refers to the *long-term* behavior of random walks in G (as opposed to walks of a particular length k). We give a complete problem that seems much closer, specifically by restricting to graphs of polynomial mixing time (as measured by $\lambda_\pi(G)$).

POLY-MIXING S-T CONNECTIVITY:

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a out-regular directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **YES instances:** The random walk on G has a stationary distribution π such that $\lambda_\pi(G) \leq 1 - 1/k$, and $\pi(s), \pi(t) \geq 1/k$.
- **NO instances:** There is no path from s to t in G.

POLY-MIXING FIND PATH:

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a out-regular directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **Promise:** $\lambda_s(G) \leq 1 - 1/k$, and $\pi_s(s), \pi_s(t) \geq 1/k$.
- **Output:** A path from s to t in G.

The completeness of these two problems is given by the following theorem.

Theorem 3.1 POLY-MIXING S-T CONNECTIVITY *is complete for* $\mathbf{prRL}$. POLY-MIXING FIND PATH *is complete for* $\mathbf{searchRL}$.

Proof: See Appendix A.3. ■

4 Operations on Directed Graphs

Given the $\mathbf{RL}$-complete problem from the previous section, it is natural to ask whether Reingold's algorithm [Rei] for UNDIRECTED S-T CONNECTIVITY can be generalized to work for the complete problem. Recall that the algorithm works by taking any undirected graph G and applying a sequence of operations to improve its expansion, as measured by spectral gap. Specifically, it relies on a pair of operations that doubles the spectral gap while keeping the degree constant (and increasing the number of vertices by a constant factor). Since the initial (non-bipartite, connected) undirected graph G has spectral gap $\gamma(G) \stackrel{\text{def}}{=} 1 - \lambda(G) \geq 1/\mathrm{poly}(N)$, after $O(\log N)$ operations, we have a graph G' with $\gamma(G') \geq 1/2$. That is, G' is a (constant-degree) expander graph and in particular has diameter $O(\log N)$ (in each connected component). Then s-t connectivity can be decided in logspace by enumerating all paths of $O(\log N)$ from s.

Attempting to generalize this approach to the **RL**-complete problem POLY-MIXING S-T CONNECTIVITY, we observe that the initial condition $\gamma(G) \geq 1/\mathrm{poly}(N)$ holds by the promise (taking N to be the length of the input). In addition, if we manage to convert G into a constant-degree graph G' with $\gamma(G') \geq 1/2$ while maintaining the fact that s and t have stationary probability at least $1/\mathrm{poly}(N)$, then Lemma 2.2 implies that there is a path of length $O(\log N)$ from s to t and we can solve s-t connectivity by enumerating all such paths.

Thus, the "only" missing part of the algorithm is generalizing the operations used by Reingold to improve expansion (without increasing the degree) to directed graphs. Below we suggest some possibilities.

Labellings. Let G be a digraph with N vertices such that every vertex has outdegree at most D_{out} and indegree at most D_{in}. (Recall that we allow multiple edges and self-loops.) A *two-way labelling* of G provides a numbering of the edges leaving each vertex of G using some subset of the numbers from 1 to D_{out}, as well as a numbering of edges entering each vertex of G using some subset of the numbers from 1 to D_{in}. (No two edges leaving a vertex can have the same number, and no two edges entering a vertex can have the same number.) Such a graph together with its two-way labelling can be specified by a **rotation map** $\mathrm{Rot}_G : [N] \times [D_{\mathrm{out}}] \to ([N] \times [D_{\mathrm{in}}]) \cup \{\perp\}$, where $\mathrm{Rot}_G(v, i) = (u, j)$ if there is an edge numbered i leaving v and it equals the edge numbered j entering u, and $\mathrm{Rot}_G(v, i) = \perp$ if there is no edge numbered i leaving v. The operations below will be defined in terms of 2-way labellings, as specified by rotation maps.

See the Appendix A.4 for definitions of powering, replacement and zig-zag product for digraphs.

5 Regular (and Eulerian) Graphs

We define REGULAR DIGRAPH S-T CONNECTIVITY and REGULAR DIGRAPH FIND PATH to be the problems obtained by restricting S-T CONNECTIVITY and FIND PATH to regular digraphs, and similarly EULERIAN S-T CONNECTIVITY and EULERIAN FIND PATH to be the restrictions to Eulerian digraphs — directed graphs where every vertex has the same in-degree as out-degree. There is no additional promise in these problems. It is not difficult to see that EULERIAN S-T CONNECTIVITY reduces to UNDIRECTED S-T CONNECTIVITY, simply by making all edges undirected. Whether or not s and t are connected is maintained because, in an Eulerian graph, every cut has the same number of edges crossing in both directions. Note, however, that this is *not* a reduction from EULERIAN FIND PATH to UNDIRECTED FIND PATH. Nevertheless, here we give a logspace algorithm for EULERIAN FIND PATH by generalizing the ideas underlying Reingold's algorithm [Rei] to the directed case. (The proof is in Appendix A.5.)

Theorem 5.1 EULERIAN FIND PATH *is in* **searchL**.

6 Oblivious Algorithms for Consistently Labelled Graphs

The algorithm given for REGULAR DIGRAPH FIND PATH in the previous section is in the standard computational model, where the input graph is given explicitly to the logspace algorithm. However, for s-t connectivity problems, it is also interesting to seek "oblivious" algorithms that do not explicitly get the input graph, but are only able to walk on the graph by specifying a sequence of outgoing edge labels. That is, the algorithm is given the parameters of the input graph G (namely, number of vertices N and degree D), and then tries to produce a walk $w \in [D]^*$ such that the walk in G obtained starting at s and following the edge labels in w visits t at some point.

Notice that the behavior of such an oblivious algorithm is sensitive to the labelling of outgoing edges in G, but incoming edge labels are irrelevant. Thus, now we think of our D-regular digraph G as being specified with a *one-way labelling*; that is, the outgoing edges from each vertex are numbered from 1 to D. (In contrast, the algorithm presented in the previous section can be thought of as being given an *unlabelled* graph, then it constructs its own two-way labelling to facilitate the applications of the zig-zag product.)

Here we present two types of oblivious algorithms for regular digraphs, one being a deterministic, logspace construction of "universal traversal sequences" and the other being a logspace-computable "pseudorandom generator" for random walks on the graph.

These algorithms will only work on regular digraphs that are **consistently labelled**, which means that all the edges coming into any vertex of the graph have distinct labels, i.e. no vertex v can be both u's i^{th}-neighbor and w's i^{th}-neighbor (for any distinct vertices u and w). In other words, if we use the same labels to number the edges incoming at each vertex (if (u, v) is the i'th edge leaving u, we consider it to be the i'th edge entering v), we obtain a legal *two-way* labelling of the graph (in that each label in $[D]$ will get used exactly once as an incoming label each vertex). Every regular digraph has a consistent labelling; this is equivalent to the fact that every D-regular bipartite graph is the union of D perfect matchings. . However, finding a consistent labelling may not be feasible in log-space, and in any case an oblivious algorithm does not have the freedom to relabel the graph.

We remark that oblivious algorithms like the ones we describe often have applications that non-oblivious algorithms may not. For example, pseudorandom generators fooling logspace algorithms, such as [Nis2, NZ], have a variety of applications that do not seem to follow arbitrary deterministic simulations of $\mathbf{RL}$, e.g. [Ind, Siv, HVV, HHR]. Even our pseudorandom generator in Section 6.2 below has already found an application in the construction of almost k-wise independent permutations [KNR].

6.1 Universal Traversal Sequences

Definition 6.1 ([AKL$^+$]) *Let D and N be two integers and let $\mathcal{G}$ be a subset of the labelled D-regular connected digraphs on N vertices. We say that a sequence of values in $[D]$ is a* **universal traversal sequence** *for $\mathcal{G}$, if for every graph $G \in \mathcal{G}$, and every vertex $s \in [N]$, the walk that starts in s and follows the edges of G according to the sequence of labels visits all the vertices of the graph.*

We will show how the REGULAR DIGRAPH FIND PATH algorithm described in the previous section also implies a log-space constructible universal traversal sequence for *consistently labelled* regular digraphs. (The proof is in Appendix A.6)

Theorem 6.2 *There exists a log-space algorithm that on input $1^N, 1^D$ produces a universal traversal sequence for all connected, consistently labelled D-regular digraphs G on N-vertices.*

6.2 A Pseudorandom Generator

In this section we show that the path finding algorithm also implies a generator with logarithmic seed length that produces in log-space a "pseudorandom walk" for consistently labelled regular digraphs. This means that from any start vertex, following the pseudorandom walk leads to an almost uniformly distributed vertex. In other words, just as the random walk, the pseudorandom walk converges to the stationary distribution. This seems to be a result of independent interest. In particular, we show in Section 7 that a similar pseudorandom generator (or even weaker), that works for regular digraphs with *arbitrary labels*, would prove that $\mathbf{RL} = \mathbf{L}$.

The intuition for the generator is as follows. In the path-finding algorithm, an expander graph G_{exp} is constructed. In this graph a short random walk converges to the uniform distribution. As in the proof for the universal traversal sequences, the sequence of labels of the (random) walk on G_{exp} can be translated to a (pseudorandom) sequence of labels for a walk on G. Furthermore, this sequence of labels is independent of G (and can be computed in log-space without access to G). Note that all nodes of the original graph G are expanded to "clouds" of equal size. Therefore, the pseudorandom walk converges to the uniform distribution on the vertices of G (which is the projection on G of the uniform distribution on the vertices of G_{exp}). Formalizing the above arguments will indeed imply a generator that produces a pseudorandom walk of length polynomial in the size of the graph. However, a truly random walk will converge faster if G has a larger eigenvalue gap. Theorem 6.3 below takes this into account and implies, in this case, a pseudorandom walk that is shorter as well. The proof is in Appendix A.7.

Theorem 6.3 *For every $N, D \in \mathbb{N}$, $\delta, \gamma > 0$, there is a generator $\mathrm{PRG} = \mathrm{PRG}_{N,D,\delta,\gamma} : \{0,1\}^r \to [D]^\ell$ with seed length $r = O(\log(ND/\delta\gamma))$, and walk length $\ell = \mathrm{poly}(1/\gamma) \cdot \log(ND/\delta)$, computable in space $O(\log(ND/\delta\gamma))$ such that for every* consistently labelled *$(N, D, 1-\gamma)$ regular digraph G and every vertex s in G, talking walk $\mathrm{PRG}(U_r)$ from s ends at a vertex that is distributed δ-close to uniform (in variation distance).*

7 Reducing all of RL to the Regular Case

In this section, we prove that if there exists a pseudorandom generators for walks on regular digraphs *whose edges are arbitrarily labelled*, then $\mathbf{RL} = \mathbf{L}$ and also $\mathbf{searchRL} = \mathbf{searchL}$. Theorem 6.3 implies a generator for walks on regular digraphs with the additional restriction that the labelling of the edges is consistent. Lifting this restriction would imply that $\mathbf{RL} = \mathbf{L}$. In fact, such a generator would also imply $\mathbf{BPL} = \mathbf{L}$. However, we concentrate in this preliminary version on the case of $\mathbf{RL}$.

Theorem 7.1 *There is a universal constant $\alpha > 0$ such that the following holds for every constant $a \in \mathbb{N}$. Suppose that for every $N, D \in \mathbb{N}$, $\delta, \gamma > 0$, there is a generator $\mathrm{PRG} = \mathrm{PRG}_{N,D,\delta,\gamma} : \{0,1\}^r \to [D]^\ell$ with seed length $r = a\log(ND/\delta\gamma)$, and walk length $\ell = (1/(\gamma\delta))^a \cdot (ND)^\alpha$, computable in space $a\log(ND/\delta\gamma)$ such that for every $(N, D, 1-\gamma)$ regular digraph $G = (V, E)$ and every vertex $s \in V$ and every subset $T \subseteq V$ of density at least δ, the walk from s following the labels $\mathrm{PRG}(U_r)$ visits T with probability at least $(\delta\gamma)^a/(ND)^\alpha$. Then $\mathbf{RL} = \mathbf{L}$ and $\mathbf{searchRL} = \mathbf{searchL}$.*

Note that the above theorem requires that the length ℓ of the pseudorandom walks have limited dependence on N and D, being bounded by $(ND)^\alpha$ rather than being polynomial or even linear in ND. Still, this is a much milder requirement than what is achieved by our generator for consistently labelled graphs (Thm. 6.3), which achieves logarithmic dependence. We also note that a pseudorandom generator for logspace algorithms with logarithmic seed length would imply the above theorem, because a truly random walk of length $O(1/\gamma) \cdot O(\log(ND/\delta))$ would end at T with probability at least $\delta/2$, and such a walk can be implemented in space $O(\log(ND/\delta\gamma))$.

Roughly speaking, we will prove Theorem 7.1 by showing that for every poly-mixing graph G, there exists a regular digraph G_{reg} such that the correctness of the generator on G_{reg} implies the correctness of (a modification of) the generator on G. Thus, if we have a generator that works well on regular digraphs, we obtain a generator that works well on instances of POLY-MIXING S-T CONNECTIVITY, which we have shown to be $\mathbf{RL}$-complete (Theorem 3.1). We stress that this construction is only done in the *analysis*, and thus need not be computable in log-space. See the Appendix A.8 for details.

Acknowledgments

We are grateful to Irit Dinur for her invaluable collaboration during the early stages of this work and for her contribution to the results of Section A.9. We also thank David Zuckerman, Eyal Rozenman, David Karger, and Nati Linial for helpful discussions and suggestions.

References

[AKS] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic Simulation in LOGSPACE. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, New York City, 25–27 May 1987.

[AKL$^+$] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, 29–31 Oct. 1979. IEEE.

[ATSWZ] R. Armoni, A. Ta-Shma, A. Wigderson, and S. Zhou. An $O(log(n)^{4/3})$ space algorithm for (s,t) connectivity in undirected graphs. *Journal of the ACM*, 47(2):294–311, 2000.

[BNS] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, pages 204–232, 15–17 May 1989.

[CPS] S. Caracciolo, A. Pelissetto, and A. Sokal. Two Remarks on Simulated Tempering. Unpublished manuscript (see [MR1]), 1992.

[Fil] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible markov chains with an application to the exclusion process. *Annals of Applied Probability*, 1:62–87, 1991.

[HHR] I. Haitner, D. Harnik, and O. Reingold. On the power of the randomized iterate. submitted to this conference, 2005.

[HVV] A. Healy, S. Vadhan, and E. Viola. Using nondeterminism to amplify hardness. In *36th ACM Symposium on Theory of Computing (STOC '04)*, pages 192–201, Chicago, IL, 13–15 June 2004. ACM.

[HW] S. Hoory and A. Wigderson. Universal Traversal Sequences for Expander Graphs. *Inf. Process. Lett.*, 46(2):67–69, 1993.

[Ind] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 189–197. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

[KNR] E. Kaplan, M. Naor, and O. Reingold. Derandomized Constructions of k-Wise (Almost) Independent Permutations. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM '05)*, number 3624 in Lecture Notes in Computer Science, pages 354 – 365, Berkeley, CA, August 2005. Springer.

[Lov] L. Lovász. *Combinatorial problems and exercises*. North-Holland Publishing Co., Amsterdam, second edition, 1993.

[MR1] N. Madras and D. Randall. Markov chain decomposition for convergence rate analysis. *Annals of Applied Probability*, 12:581–606, 2002.

[MR2] R. A. Martin and D. Randall. Sampling Adsorbing Staircase Walks Using a New Markov Chain Decomposition Method. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 492–502, Redondo Beach, CA, 17–19 Oct. 2000. IEEE.

[Mih] M. Mihail. Conductance and convergence of markov chains: a combinatorial treatment of expanders. In *In Proc. of the 37th Conf. on Foundations of Computer Science*, pages 526–531, 1989.

[Nis1] Nisan. RL $\subseteq$ SC. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 619–623, 1992.

[Nis2] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NSW] N. Nisan, E. Szemeredi, and A. Wigderson. Undirected connectivity in $O(log^{1.5}n)$ space. In *Proceedings of the 30th FOCS*, pages 24–29, Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989. IEEE.

[NZ] N. Nisan and D. Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Sciences*, 52(1):43–52, Feb. 1996.

[Rei] O. Reingold. Undirected ST-Connectivity in Log-Space. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 376–385, 2005.

[RTV] O. Reingold, L. Trevisan, and S. Vadhan. Pseudorandom Walks in Biregular Graphs and the RL vs. L Problem. *Electronic Colloquium on Computational Complexity* Technical Report TR05-022, February 2005. `http://www.eccc.uni-trier.de/eccc`.

[RVW] O. Reingold, S. Vadhan, and A. Wigderson. Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders. *Annals of Mathematics*, 155(1), January 2001. Extended abstract in *FOCS '00*.

[RV] E. Rozenman and S. Vadhan. Derandomized Squaring of Graphs. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM '05)*, number 3624 in Lecture Notes in Computer Science, pages 436–447, Berkeley, CA, August 2005. Springer. See also preliminary version, ECCC TR05-92.

[Sak] M. Saks. Randomization and Derandomization in Space-Bounded Computation. In *IEEE 11th Annual Conference on Structure in Complexity Theory*, 1996.

[SZ] M. Saks and S. Zhou. $\mathrm{BP_HSPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999. 36th IEEE Symposium on the Foundations of Computer Science (Milwaukee, WI, 1995).

[Sav] J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[SJ] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Inform. and Comput.*, 82(1):93–133, 1989.

[Siv] D. Sivakumar. Algorithmic derandomization via complexity theory. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 619–626 (electronic), New York, 2002. ACM.

[Wig] A. Wigderson. The complexity of graph connectivity. In *In Proceedings of the 17th Mathematical Foundations of Computer Science*, pages 112–132, 1992.

A Appendix

A.1 Proofs Omitted from Section 2

A.1.1 Proof of Lemma 2.2

Proof: [Of Lemma 2.2] We note that $\alpha - \pi$ is orthogonal to π and that M preserves orthogonality to π. Thus,

$$\|M^t\alpha - \pi\|_\pi = \|M^t(\alpha - \pi)\|_\pi \le \lambda_\pi(M)^t \cdot \|\alpha - \pi\|_\pi.$$

For the "in particular," we take α to be the distribution concentrated at v and note that

$$\|\alpha - \pi\|_\pi^2 = \frac{(1-\pi(v))^2}{\pi(v)} + \sum_{w \neq v} \frac{\pi(w)^2}{\pi(w)} \le \frac{1}{\pi(v)},$$

and that

$$|(M^t\alpha)(w) - \pi(w)|^2 \le \pi(w) \cdot \|\pi - M^t\alpha\|_\pi^2.$$

■

A.1.2 Proof of Lemma 2.3

Proof: [Of Lemma 2.3] The variation distance between α and π equals

$$\begin{aligned}
\sum_{v:\pi(v)>\alpha(v)} \pi(v) - \alpha(v) &\le \sum_{v \in \mathrm{supp}(\pi)} \frac{|\alpha(v) - \pi(v)|}{\sqrt{\pi(v)}} \cdot \sqrt{\pi(v)} \\
&\le \left(\sum_{v \in \mathrm{supp}(\pi)} \frac{|\alpha(v) - \pi(v)|^2}{\pi(v)} \right)^{1/2} \cdot \left(\sum_{v \in \mathrm{supp}(\pi)} \pi(v) \right)^{1/2} \\
&= \|\alpha - \pi\|_\pi \cdot 1.
\end{aligned}$$

■

A.1.3 Proof of Lemma 2.5

Proof: [Of Lemma 2.5] Mihail [Mih] proves the bound

$$\lambda_\pi(M) \le \sqrt{1 - h_\pi(M)^2} \le 1 - \frac{1}{2} h_\pi^2(M) \; .$$

Here we give a simpler proof, using techniques of Fill [Fil].

First of all, we may assume without loss of generality that $\pi(u) > 0$ for every $u \in [N]$. Otherwise, we can consider the restriction of M to the sub-matrix whose rows and columns are indexed by vertices u such that $\pi(u) > 0$. Such sub-matrix has the same stationary distribution, spectral gap and conductance of M and satisfies our assumption.

Define the *time reverse* of M as the Markov chain $\tilde{M}$ such that $\tilde{M}(u,v) = \pi(u)M(v,u)/\pi(v)$. The following claims are easy to check:

- $\tilde{M}$ is a Markov chain, that is, for every v, $\sum_u \tilde{M}(u,v) = 1$.

  (Note that we have $\sum_u \tilde{M}(u,v) = \sum_u \pi(u)M(v,u)/\pi(v)$ by definition, and $\sum_u \pi(u)M(v,u) = \pi(v)$ by stationarity.)

- π is stationary for $\tilde{M}$.

- $\tilde{M}$ is strongly aperiodic.

Define $P \stackrel{\text{def}}{=} \tilde{M}M$. Then P is also a Markov chain, π is also a stationary distribution for P and, in addition, we have that P is *time reversible*, that is, $\pi(v)P(u,v) = \pi(u)P(v,u)$. The eigenvalues of a time reversible Markov chain R with stationary distribution π all real, are all at most 1, and we denote the second largest eigenvalue in absolute value by $\lambda_2(R)$.

In the particular case of P, it is not hard to see that

$$\lambda_2(P) = (\lambda_\pi(M))^2 \tag{1}$$

and so we are left with the task of proving that $\lambda_2(P) \leq 1 - h_\pi^2(M)$.

Since M and $\tilde{M}$ are strongly aperiodic, we can write them as $M = \frac{1}{2}I + \frac{1}{2}L$ and $\tilde{M} = \frac{1}{2} + \frac{1}{2}\tilde{L}$ where $L, \tilde{L}$ are Markov chains with stationary distribution π. Using this notation, we can write

$$P = \tilde{M}M = \left(\frac{1}{2} + \frac{1}{2}\tilde{L}\right)\left(\frac{1}{2} + \frac{1}{2}L\right) = \frac{1}{4}I + \frac{1}{4}\tilde{L} + \frac{1}{4}L + \frac{1}{4}\tilde{L}L$$

The next observation is that $\frac{1}{2}L + \frac{1}{2}\tilde{L}$ is a time-reversible Markov chain with stationary probability π, and so are $\tilde{L}L$ and I, and that λ_2 is a norm for such matrices, so we have

$$\begin{aligned}\lambda_2(P) &\leq \frac{1}{4}\lambda_2(I) + \frac{1}{2}\lambda_2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right) + \frac{1}{4}\lambda_2(\tilde{L}L) \\ &\leq \frac{1}{2} + \frac{1}{2}\lambda_2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right)\end{aligned}$$

At this point we are ready to use a result of Jerrum and Sinclair [SJ], who prove that for every time-reversible Markov chain R and stationary distribution π we have $\lambda_2(R) \leq 1 - h_\pi^2(R)/2$. Applying this result to $\frac{1}{2}L + \frac{1}{2}\tilde{L}$ we get

$$\lambda_2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right) \leq 1 - \frac{1}{2}h_\pi^2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right)$$

It remains to study the conductance of $\frac{1}{2}L + \frac{1}{2}\tilde{L}$. We first note that

$$h_\pi\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right) = 2h_\pi\left(\frac{1}{2}M + \frac{1}{2}\tilde{M}\right)$$

because every edge that is not a self-loop has twice as much weight in L (respectively $\tilde{L}$) than in M (respectively $\tilde{M}$). Finally, we have

$$h_\pi\left(\frac{1}{2}M + \frac{1}{2}\tilde{M}\right) = h_\pi(M)$$

This identity comes from the fact that, for every cut $S, [N] - S$ of the set of vertices we have

$$\sum_{u\in S, v\notin S} \pi(u)M(v,u) = \sum_{u\in S, v\notin S} \pi(v)M(u,v) = \sum_{u\in S, v\notin S} \pi(u)\tilde{M}(v,u)$$

Collecting all our inequalities together we have

$$\lambda_\pi(M) = \sqrt{\lambda_2(P)} \leq \sqrt{\frac{1}{2} + \frac{1}{2}\lambda_2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right)} \leq \sqrt{\frac{1}{2} + \frac{1}{2} - \frac{1}{4}h_\pi^2\left(\frac{1}{2}L + \frac{1}{2}\tilde{L}\right)} = \sqrt{1 - h_\pi^2(M)} \leq 1 - \frac{1}{2}h_\pi^2(M)$$

■

A.2 Definitions of Complexity Classes

We let $\mathbf{L}$, $\mathbf{RL}$, $\mathbf{NL}$, $\mathbf{BPL}$ denote the standard logspace complexity classes. That is, $\mathbf{L}$ is the class of decision problems solvable by *deterministic* logarithmic space Turing machines, $\mathbf{RL}$ is the class of decision problems solvable by *probabilistic*logarithmic space Turing machines with bounded one-sided error, $\mathbf{BPL}$ is the class of decision problems solvable by *probabilistic*logarithmic space Turing machines with bounded two-sided error, and $\mathbf{NL}$ is the class of decision problems solvable by *non-deterministic*logarithmic space Turing machines. We require our machines to always terminate for every input and for every sequence of random coins or non-deterministic choices. In particular, this implies that every computation terminates within polynomial time.

We also define the *promise version* of log-space complexity classes. A *promise* problem is a pair (Y, N) of disjoint sets of instances. A promise problem (Y, N) is in the class $\mathbf{prL}$ if there is a deterministic log-space Turing machine that accepts all the inputs in Y and rejects all the inputs in N. A promise problem (Y, N) is in $\mathbf{prRL}$ if there is a probabilistic logarithmic space Turing machine that accepts inputs in Y with probability at least $1/2$ and accepts inputs in N with probability 0. A promise problem (Y, N) is in $\mathbf{prBPL}$ if there is a probabilistic logarithmic space Turing machine that accepts inputs in Y with probability at least $3/4$ and accepts inputs in N with probability at most $1/4$. When dealing with promise problems, we require probabilistic machines to halt for every input in $Y \cup N$ and for every sequence of random coins. (We allow infinite loops for inputs not in $Y \cup N$.)

Finally, we define complexity classes of *search problems*. A **search problem** is simply a relation $R \subseteq \Sigma^* \times \Sigma^*$. For a relation R and a string x we define $R(x) \stackrel{\text{def}}{=} \{y : R(x,y)\}$. The computational problem associated with a search problem R is the following: given x such that $R(x) \neq \emptyset$, output a string y in $R(x)$.

A relation (or search problem) R is **log-space** if there is a polynomial p such that $y \in R(x)$ implies $|y| \leq p(|x|)$ and if the predicate $(x, y) \in R$ can be decided by a log-space deterministic Turing machine that has two-way access to x and one-way access to y.

A logspace search problem R is in $\mathbf{searchL}$ if there is a logarithmic space transducer A such that $A(x) \in R(x)$ for every x such that $R(x) \neq \emptyset$. (A transducer is a Turing machine with a read-only input tape, a work tape, and a write-only output tape. The writing head on the output tape is constrained to always move right after writing a symbol, but the machine has two-way access to the input tape.)

A logspace search problem R is in $\mathbf{searchRL}$ if there is a logarithmic space probabilistic transducer A and a polynomial p such that $\Pr\left[A(x) \in R(x)\right] \geq \frac{1}{p(x)}$ for every x such that $R(x) \neq \emptyset$. (We require the transducer to halt for every sequence of random coins and for every x such that $R(x) \neq \emptyset$.)

All **reductions** in this paper are deterministic logspace reductions. The definition of reduction is standard for decision problems and promise problems.

For two search problems R_1 and R_2, we say that R_1 reduces to R_2 if there are two functions $f(\cdot)$ and $g(\cdot,\cdot)$ such that

1. If $R_1(x)$ is non-empty then $R_2(f(x))$ is non-empty;
2. If $z \in R_2(f(x))$, then $g(x,z)$ outputs a sequence $y_1,\ldots,y_k$ such that at least one element y_i of the sequence is in $R_1(x)$;
3. $f()$ is computable by a deterministic log-space transducer
4. $g(x,y)$ is computable by a deterministic log-space transducer with two-way access to x and one-way access to z.

It can be verified that if $\Pi \in \mathbf{searchNL}$ reduces to Π', then $\Pi' \in \mathbf{searchL} \Rightarrow \Pi \in \mathbf{searchL}$.

A.3 Proof of Theorem 3.1

Proof: [Of Theorem 3.1] First, we show that these problems are in $\mathbf{prRL}$ and $\mathbf{searchRL}$, respectively, by giving randomized logspace algorithms for them. Given an instance $(G,s,t,1^k)$, we take a random walk of length $m = 2k \cdot \ln k$ from s, where N is the number of vertices in G. The $\mathbf{searchRL}$ algorithm simply outputs this walk, and the $\mathbf{prRL}$ algorithm accepts if this walk ends at t. If $(G,s,t,1^k)$ is a YES instance, then by Lemma 2.2, the random walk will end at t with probability at least

$$\begin{aligned}\pi(t) - \lambda_\pi(M)^m \cdot \sqrt{\pi(t)/\pi(s)} &\geq \frac{1}{k} - \left(1-\frac{1}{k}\right)^m \cdot \sqrt{k} \\ &\geq \frac{1}{k} - \frac{1}{k^{3/2}} \geq \frac{1}{2k}.\end{aligned}$$

Now we show that every problem in $\mathbf{prRL}$ and $\mathbf{searchRL}$, respectively, reduce to POLY-MIXING S-T CONNECTIVITY and POLY-MIXING FIND PATH. Let M be a randomized logspace machine, running in time at most $p(n) \leq \mathrm{poly}(n)$. Given an input x of length n for M, we construct a graph G whose vertices are of the form (i,τ), where $i \in \{1,\ldots,p(n)\}$ is a "layer", and $\tau \in \{0,1\}^{O(\log n)}$ describes a possible configuration of M (i.e. the state, the contents of the work tape, and the position of the input head). We let $s = (1,\alpha)$ where α is the unique start configuration of M, and $t = (p(n),\beta)$ where β is the (wlog unique) accepting configuration of M. (In the case of a $\mathbf{searchRL}$ algorithm, we have M accept if any of the strings it outputs satisfy the relation R.) We create four outgoing edges from each vertex (i,γ). Two edges are always self-loops. If $i = p(n)$, then the other two edges go to s. If $i < p(n)$, then the they go to vertices of the form $(i+1,\gamma')$ and $(i+1,\gamma'')$, for γ',γ'' as follows. If γ is a configuration where M reads a new random bit, then we take γ' and γ'' to be the two configurations that M would enter depending on the two possible values of the random bit. If γ is a configuration where M does not read a new random bit, then we set $\gamma' = \gamma''$ to be the unique next configuration in M's computation on x. If γ is a halting configuration of M, then we set $\gamma' = \gamma'' = \gamma$.

Let us analyze the stationary distribution and mixing time of a random walk on G. It can be verified that the following distribution π is on vertices (i,τ) is stationary for G: choose i uniformly at random from $\{1,\ldots,p(n)\}$, run M for i steps on input x, and let τ be M's configuration. We see that if $x \in L$, then $\pi(t) > 1/2p(n)$, and if $x \notin L$, then $\pi(t) = 0$. In both cases $\pi(s) = 1/p(n)$.

To bound the mixing time, we observe that a random walk of length $3p(n)$ started at *any* vertex visits s with probability $1 - 2^{-\Omega(p(n))} \geq 1/2$. Lemma A.1 below says that G has a stationary distribution π' such that $\lambda_{\pi'}(G) \leq 1 - 1/(8 \cdot (3p(n))^2)$ and $\pi'(s) > 0$. It follows that π' is the unique stationary distribution on G, since a random walk started at any vertex eventually passes through s and thus converges to π' (by Lemma 2.2). So $\pi' = \pi$.

To conclude, in our reduction, we output $(G, s, t, 1^k)$, where $k = 72p(n)^2$. From the analysis above, this gives a logspace reduction from any problem in **prRL** to POLY-MIXING S-T CONNECTIVITY. Similarly, it gives a reduction from any problem in **searchRL** to FIND PATH, because with one-way access to any path from s to t in G, in logspace we can construct polynomially many computation paths of M, at least one of which is accepting, and this in turn, can be used to obtain a polynomially many strings $y_1, \ldots, y_\ell$ at least of which is in $R(x)$. ∎

The above proof required the following lemma, which says that to show that a Markov chain has polynomial mixing time, it suffices to prove that there is a vertex s such that a random walk of polynomial length started at any vertex will visit s with high probability.

Lemma A.1 *Let M be a Markov chain that is strongly aperiodic (i.e. self-loop probability at least 1/2 at each vertex). Suppose there is a vertex s and a number $\ell \in \mathbb{N}$ such that from every vertex v reachable from s, a random walk of length ℓ from v visits s with probability at least $1/2$. Then M has a stationary distribution π such that $\lambda_\pi(G) \leq 1 - 1/8\ell^2$ and $\pi(s) \geq 1/2\ell$.*

Proof: Let M' be the restriction of M to the set of all vertices reachable from s. Let π be a stationary distribution of the random walk on on M'. Because of the self-loops and the fact that $\pi(s) > 0$ (since every vertex in M' has a path to s), we can bound $\lambda_\pi(M)$ by computing the conductance $h_\pi(M)$ and applying Lemma 2.5. To lower-bound the conductance, we need to lower bound $\Pr[X' \notin A | X \in A] = \Pr[X \in A \wedge X' \notin A] / \pi(A)$, where X is chosen according to π, X' is a random step from X, and A is any set such that $0 < \pi(A) \leq 1/2$. To bound this, we consider a random walk $X_1, \ldots, X_\ell$ of length ℓ started in the stationary distribution π, and separate into two cases depending on whether $s \in A$.

If $s \notin A$, then the following holds:

$$\begin{aligned} \ell \cdot \Pr\left[X \in A \wedge X' \notin A\right] &\geq \Pr\left[\exists i X_i \in A \wedge X_{i+1} \notin A\right] \\ &\geq \Pr\left[X_1 \in A, s \in \{X_2, \ldots, X_\ell\}\right] \\ &\geq \pi(A) \cdot (1/2), \end{aligned}$$

where the last inequality holds because a random walk of length ℓ (from any vertex in G') visits s with probability at least $1/2$ by hypothesis.

If $s \in A$, then the following holds:

$$\begin{aligned} \ell \cdot \Pr\left[X \in A \wedge X' \notin A\right] &= \ell \cdot \Pr\left[X \notin A \wedge X' \in A\right] \\ &\geq \Pr\left[\exists i X_i \notin A \wedge X_{i+1} \in A\right] \\ &\geq \Pr\left[X_1 \notin A, s \in \{X_2, \ldots, X_\ell\}\right] \\ &\geq \pi(\overline{A})/2 \\ &\geq \pi(A)/2 \end{aligned}$$

Thus, we conclude that $\Pr[X' \notin A | X \in A] \geq 1/(2\ell)$ for every A such that $0 < \pi(A) \leq 1/2$, and hence $h_\pi(G) \geq 1/(2\ell)$. By Lemma 2.5, $\lambda_\pi \leq 1 - 1/(2 \cdot (2\ell)^2)$.

To lower bound $\pi(s)$, we note that the expected number of times s is visited in $X_1, \ldots, X_\ell$ equals $\pi(s) \cdot \ell$ on one hand, and is at least $1/2$ on the other. Thus $\pi(s) \cdot \ell \geq 1/2$. ∎

In fact, the converse is also true — if a Markov chain has polynomial mixing time then there is a vertex s such that a random walk of polynomial length started at any vertex will visit s with high probability. Indeed, if $\lambda_\pi(M) \leq 1-\gamma$ and we take s to be any vertex such that $\pi(s) \geq 1/N$ (where N is the number of states), then Lemma 2.2 says that a random walk of length $\ell = O((1/\gamma)\cdot\log(N/p_{min}))$ will end at s with probability at least $1/2N$, where p_{min} is the minimum (nonzero) probability mass under π. Repeating $O(N)$ times, we visit s with high probability. In cases we are interested in (e.g. random walks on graphs), p_{min} is only exponentially small, so the walk length $\ell \cdot N$ is polynomial.

We note that the proof of Theorem 3.1 can be modified to give a complete problem for $\mathbf{prBPL}$, specifically where the NO instances are replaced with instances such that $\lambda_\pi(G) \leq 1-1/k$, $\pi(s) \geq 1/k$ and $\pi(t) \leq 1/2k$. We also note that, following [AKL$^+$], the randomized algorithm for POLY-MIXING S-T CONNECTIVITY also gives a nonconstructive existence proof of polynomial-length universal traversal sequences for the corresponding class of graphs:

Proposition A.2 *There is a polynomial p such that for every N, D, k, there exists a sequence $\psi \in [D]^{p(N,D,k)}$ such that for every N-vertex labelled directed graph G of outdegree D and vertex s in G such that $\lambda_s(G) \leq 1-1/k$, following the walk ψ from s visits all vertices v of G for which $\pi_s(v) \geq 1/k$.*

A.4 Operations on Directed Graphs

The first operation used by Reingold [Rei] to improve expansion is powering, simply replaces the edge set with all walks of length t in the graph.

Definition A.3 (powering) *Let G be a two-way labelled graph given by rotation map $\mathrm{Rot}_G : [N]\times[D] \to [N]\times[B]$. The* **$t$'th power** *of G is the graph G^t with rotation map is given by $\mathrm{Rot}_{G^t} : [N]\times[D]^t \to [N]\times[B]^t$ defined by $\mathrm{Rot}_{G_t}(v_0,(k_1,k_2,\ldots,k_t)) = (v_t,(\ell_t,\ell_{t-1},\ldots,\ell_1))$, where these values are computed via the rule $(v_i,\ell_i) = \mathrm{Rot}_G(v_{i-1},k_i)$ (and if any of these evaluations yield $\perp$, then the final output is also $\perp$).*

In directed graphs, powering improves expansion (i.e. reduces mixing time) as well as it does in undirected graphs:

Lemma A.4 *For any stationary distribution π of G, $\lambda_\pi(G^t) \leq \lambda_\pi(G)^t$.*[3]

Powering alone does not suffice, because it increases the degree of the graph. Thus, Reingold [Rei] requires an additional operation to reduce the degree while maintaining the expansion. For this, there are two possibilities — the replacement product and zig-zag product. These operations were defined and analyzed in [RVW] for undirected regular graphs, and it is not clear what is the 'right' generalization to irregular directed graphs (particularly non-Eulerian graphs, where the indegree and outdegree of an individual vertex may be unequal). Here we suggest one possibility. For simplicity, we restrict to rotation maps where the outdegree bound D is the same as the indegree bound B.

In the replacement product, we combine a graph G_1 with N_1 vertices and a rotation map of degree D_1 with a graph G_2 that has D_2 vertices and a rotation map of degree D_2. The product graph has D_1N_1 vertices, that we think of as being grouped into N_1 "clouds" of size D_1, one cloud for each vertex of G_1. Each cloud is a copy of the graph G_2. In addition, if the i-th outgoing edge from vertex v in G_1 was the j-th incoming edge in w (that is, if $\mathrm{Rot}_{G_1}(v,i) = (w,j)$, then, in the product graph, there is an edge from the i-th vertex in the cloud of v to the j-th vertex in the cloud of w. The formal definition follows.

[3] In undirected graphs this is actually an equality, but in digraphs it need not be.

Definition A.5 (replacement product) *If G_1 is a two-way labelled graph on N_1 vertices with rotation map $\mathrm{Rot}_{G_1} : [N_1] \times [D_1] \to [N_1] \times [D_1]$ and G_2 is a two-way labelled graph on D_1 vertices with rotation map $\mathrm{Rot}_{G_2} : [D_1] \times [D_2] \to [D_1] \times [D_2]$, then their* **replacement product** *$G_1 \textcircled{r} G_2$ is defined to be the graph on $[N_1] \times [D_1]$ vertices whose rotation map $\mathrm{Rot}_{G_1 \textcircled{r} G_2} : ([N_1] \times [D_1]) \times [D_2+1] \to ([N_1] \times [D_1]) \times [D_2+1]$ is as follows:*

$\mathrm{Rot}_{G_1 \textcircled{r} G_2}((v,k),i)$**:**

1. *If $i \le D_2$, let $(m,j) = \mathrm{Rot}_{G_2}(k,i)$ and output $((v,m),j)$.*
2. *If $i = D_2+1$, output $(\mathrm{Rot}_{G_1}(v,k),i)$.*
3. *If the computation of Rot_{G_2} or Rot_{G_1} yields $\perp$, then the output is $\perp$.*

A variant, called the **balanced replacement product** $G_1 \textcircled{b} G_2$ in [RVW], gives equal weight to the edges coming from G_1 and from G_2, by duplicating edges that go between clouds (ie edges of the type 2) D_2 times, for a total degree of $2D_2$.

The zig-zag product, introduced in [RVW], combines, as before, a graph G_1 with N_1 vertices and a rotation map of degree D_1 with a graph G_2 that has D_1 vertices and degree D_2. The product graph has N_1D_1 vertices as in the replacement product, but now there is an edge between two vertices if there is a length-three path in the replacement product graph between them, and the middle edge in the path crosses between two clouds. In particular, the degree of the zig-zag product graph is D_2^2, instead of D_2+1. The formal definition is below.

Definition A.6 (zig-zag product [RVW]) *If G_1 is a labelled graph on N_1 vertices with rotation map $\mathrm{Rot}_{G_1} : [N_1] \times [D_1] \to [N_1] \times [D_1]$ and G_2 is a labelled graph on D_1 vertices with rotation map $\mathrm{Rot}_{G_2} : [D_1] \times [D_2] \to [D_1] \times [D_2]$, then their* **zig-zag** *product $G_1 \textcircled{z} G_2$ is defined to be the graph on $[N_1] \times [D_1]$ vertices whose rotation map $\mathrm{Rot}_{G_1 \textcircled{z} G_2} : ([N_1] \times [D_1]) \times [D_2^2] \to ([N_1] \times [D_1]) \times [D_2^2]$ is as follows:*

$\mathrm{Rot}_{G_1 \textcircled{z} G_2}((v,k),(i,j))$**:**

1. *Let $(k',i') = \mathrm{Rot}_{G_2}(k,i)$.*
2. *Let $(w,\ell') = \mathrm{Rot}_{G_1}(v,k')$.*
3. *Let $(\ell,j') = \mathrm{Rot}_{G_2}(\ell',j)$.*
4. *Output $((w,\ell),(j',i'))$.*

In typical applications of the zig-zag or replacement products (e.g. [RVW, Rei], G_2 is taken to a constant-degree expander graph (i.e. $\gamma(G_2) = \Omega(1)$). Then, for the case of undirected graphs, it is known that the zig-zag product and the balanced replacement product have spectral gap that is at most a constant factor smaller than the spectral gap of G_1 [RVW, MR2].[4] Thus they roughly maintain expansion while reducing the degree to a constant, and this suffices for Reingold's algorithm [Rei].

Unfortunately, we do not know how to analyze the effect of the zig-zag and/or replacement products (or variants) on spectral gap for directed graphs in general. Indeed, even the stationary distribution is not well-behaved under these products; we can construct examples where the stationary probability of a vertex t goes

[4] Actually, the undirected definitions of these products are restricted to two-way labellings that are *undirected* in the sense that every edge $\{u,v\}$ has the same label as an edge leaving u as it does entering u. That is, $\mathrm{Rot} \circ \mathrm{Rot}$ is the identity.

from being noticeable (e.g. $1/N^2$) to exponentially small. In Section A.9, we show that the replacement product can actually be analyzed with respect to *edge expansion*, but then it turns out that powering no longer behaves well.

We can analyze these products (and thus extend Reingold's algorithm) for the case of *regular* digraphs, and these results are presented in the next section.

A.5 Proof of Theorem 5.1

To prove Theorem 5.1, it suffices to provide a logspace algorithm for REGULAR DIGRAPH FIND PATH, because Eulerian digraphs can be reduced to the case of 2-regular digraphs by replacing each vertex v with a directed cycle C_v of $\deg(v)$ vertices, where we connect one outgoing edge of v and one incoming edge of v to each of the vertices in C_v. Thus in the rest of this section we focus on regular digraphs.

A.5.1 Basic Facts

In a regular digraph of degree D, the rotation map $\mathrm{Rot}_G : [N] \times [D] \to [N] \times [D]$ is a permutation. Note that the uniform distribution is a stationary distribution of the random walk on a regular digraph. Thus, when working with regular digraphs, the inner product $\langle \cdot, \cdot \rangle_\pi$ and the spectral expansion $\lambda_\pi(G)$ will always be with respect to π being the uniform distribution, and we will usually omit π from the notation.

First, we note that regular digraphs have nonnegligible spectral gap, just like in the undirected case, provided every vertex has a self-loop.[5]

Lemma A.7 *Let G be a connected, D-regular digraph on N vertices in which every vertex has at least αD self-loops. Then $\lambda(G) \leq 1 - \Omega(\alpha/DN^2)$.*

Proof: We reduce to the undirected case using a technique of Fill [Fil]. Let $M = M_G$. The spectral expansion $\lambda(M)$ equals the square root of $\lambda_2(M^T M)$, i.e. the second largest eigenvalue (considering sign) of the symmetric matrix $M^T M$. Because of the self-loops in G, we can write $M = \alpha I + (1-\alpha)L$, where L is the transition matrix for the random walk on G with the αD self-loops removed from each vertex. Then

$$M^T M = \alpha^2 I + 2\alpha \cdot (1-\alpha) \cdot (L + L^T)/2 + (1-\alpha)^2 L^T L.$$

Now, $(L + L^T)/2$ is the transition matrix for the connected, undirected $2D$-regular graph obtained by taking the edges of G together with their reversals. Thus, by the known bound on the second eigenvalue of undirected graphs [Lov], we have $\lambda_2((L + L^T)/2) \leq 1 - \Omega(1/DN^2)$. Thus,

$$\begin{aligned} \lambda(M)^2 &= \lambda_2(M^T M) \\ &\leq \alpha^2 + (1-\alpha)^2 + 2\alpha \cdot (1-\alpha) \cdot \lambda_2((L + L^T)/2) \\ &\leq 1 - \Omega(\alpha/DN^2), \end{aligned}$$

as desired. ■

[5]In the preliminary version of this paper [RTV], we erroneously used the standard notion of aperiodicity (i.e. gcd of all cycle lengths is 1) instead of requiring self-loops. However, the lemma is false in this case; see [RV].

A.5.2 Zig-zag Product

In this section, we generalize the Zig-Zag Theorem of [RVW] to regular digraphs.

Theorem A.8 *If $\lambda(G_1) \leq 1-\gamma_1$ and $\lambda(G_2) \leq 1-\gamma_2$, then $\lambda(G_1 \textcircled{z} G_2) \leq 1-\gamma_1 \cdot \gamma_2^2$.*

Our algorithm, like [Rei], we will only use the following consequence of the second bound above: if G_2 is a good expander in the sense that $\lambda(G_2)$ is bounded by a constant less than 1 and $\lambda(G_1) \leq 1-\gamma_1$, then $\lambda(G_1 \textcircled{z} G_2) \leq 1-\Omega(\gamma_1)$. In the preliminary version of this paper [RTV], we presented a proof of this $1-\Omega(\varepsilon_1)$ that was conceptually simpler than the previous proofs of this bound in the undirected case, for either the zig-zag or replacement products.[6] Here we present an even simpler proof, based on an approach of Rozenman and Vadhan [RV], who used it to analyze a new 'derandomized squaring' operation (that gives an alternative to Reingold's algorithm as well our generalization to Eulerian digraphs). The key to this approach is the following lemma:

Lemma A.9 ([RV]) *Let M be a Markov chain with stationary distribution π, and suppose that $\lambda_\pi(M) \leq \lambda$. Then $M = (1-\lambda)J_\pi + \lambda \cdot E$, where J_π is the matrix such that every column equals π and E has norm at most 1 with respect to $\|\cdot\|_\pi$. (That is, $\|Ex\|_\pi \leq \|x\|_\pi$ for all x.*

Intuitively, this lemma says that we can view a random step on a Markov chain with spectral expansion λ as jumping to a random vertex under π with probability λ and "not getting any further from π" with probability $1-\lambda$. This intuition would be precise if E were stochastic, but it is not guaranteed to be so. Nevertheless, the intuition will work in the proof below.

Proof: (of Theorem A.8) Let M be the transition matrix of the random walk on $G_1 \textcircled{z} G_2$. Following [RVW], we relate M to the transition matrices of G_1 and G_2, which we denote by A and B, respectively. First, we decompose M into the product of three matrices, corresponding to the three steps in the definition of $G_1 \textcircled{z} G_2$'s edges. Let $\tilde{B}$ be the transition matrix for taking a random G_2-step on the second component of $[N_1] \times [D_1]$, i.e. $\tilde{B} = I_{N_1} \otimes B$, where I_{N_1} is the $N_1 \times N_1$ identity matrix. Let $\tilde{A}$ be the permutation matrix corresponding to Rot_{G_1}. By the definition of $G_1 \textcircled{z} G_2$, we have $M = \tilde{B}\tilde{A}\tilde{B}$.

By Lemma A.9, $B = \gamma_2 J + (1-\gamma_2)E$, where every entry of J equals $1/D_1$ and E has norm at most 1. Then $\tilde{B} = \gamma_2 \tilde{J} + (1-\gamma_2)\tilde{E}$, where $\tilde{J} = I_{N_1} \otimes J$ and $\tilde{E} = I_{N_1} \otimes E$ has norm at most 1.

This gives

$$M = \left(\gamma_2 \tilde{J} + (1-\gamma_2) tildeE\right) \tilde{A} \left(\gamma_2 \tilde{J} + (1-\gamma_2)\tilde{E}\right) = \gamma_2^2 \tilde{J}\tilde{A}\tilde{J} + (1-\gamma_2^2)F,$$

where F has norm at most 1.

Now, the key observation is that

$$\tilde{J}\tilde{A}\tilde{J} = A \otimes J.$$

The left-hand side is the stochastic matrix corresponding to the Markov chain that does the following from state (v,i): choose i' uniformly in $[D_1]$, let $(w,j') = \mathrm{Rot}_{G_1}(v,i')$, choose j uniformly in $[D_1]$ and go to state (w,j). The right-hand side corresponds to: let w be a random neighbor of v in G_1, choose j uniformly in $[D_1]$ and output (w,j). These two processes are identical by the definition of a rotation map.

[6] The basic analysis of the undirected zig-zag product in [RVW] only gives a bound of $1-\Omega(\varepsilon_1^2)$. Only a much more complicated and less intuitive analysis, that uses the undirectedness of G_1 in additional ways, gives the $1-\Omega(\varepsilon_1)$ bound. The Martin–Randall [MR2] decomposition theorem for Markov chains also implies a $1-\Omega(\varepsilon_1)$ bound for the undirected replacement products, but its full proof (relying on [CPS]) is also fairly involved.

Combining the above, we have

$$M = \gamma_2^2 \cdot A \otimes J + (1-\gamma_2^2)\tilde{F},$$

and thus

$$\begin{aligned}\lambda(M) &\leq \gamma_2^2 \cdot \lambda(A \otimes J) + (1-\gamma_2^2)\\ &\leq \gamma_2^2 \cdot (1-\gamma_1) + (1-\gamma_2^2)\\ &= 1-\gamma_1\gamma_2^2,\end{aligned}$$

as desired. ■

A.5.3 The Path-Finding Algorithm

We have seen that powering and the zig-zag graph product has essentially the same affect on regular digraphs as on undirected graphs. Therefore, both the decision and search versions of the st-connectivity algorithm of [Rei] can be extended (without any substantial change) to regular digraphs. This implies Theorem 5.1, which states that REGULAR DIGRAPH FIND PATH is in $\mathbf{searchL}$. As the algorithm here is essentially the same as in [Rei], we only provide a sketch of the proof.

Proof Sketch: [of Theorem 5.1] We describe a log-space algorithm $\mathcal{A}$ that gets as input a D-regular (i.e. both the indegree and the outdegree of each vertex is D) graph G on N-vertices and two vertices s and t and outputs a path from s to t if such a path exists (otherwise, it will output 'not connected').

The algorithm will rely on a constant size (undirected) expander graph H, given by its rotation map Rot_H, with rather weak parameters. More specifically, H will be D_e-regular, for some constant D_e, it will have $(D_\mathrm{e})^{80}$ vertices (no attempt was made to optimize the constants), and $\lambda(H) \leq 1/2$. The expander H can be obtained via exhaustive search or any one of various explicit constructions.

The first step of the algorithm, will be to reduce the input G, s, t into a new input G_{reg}, s', t' where G_{reg} is $(D_\mathrm{e})^{80}$-regular on $N \cdot D$ vertices, every connected component of G_{reg} is aperiodic, and s and t are connected in G if and only if s' and t' are connected in G_{reg}. Furthermore, a path from s' to t' in G_{reg} can be translated in log-space into a path from s to t in G. The reduction itself is quite standard: Each vertex of G is replaced with a cycle with D vertices. In addition, the i^{th} vertex (v,i) in the cycle that corresponds to v is connected to $(w,j) = \mathrm{Rot}_G(v,i)$ in the cycle that corresponds to w. Up to now, both the indegree and the outdegree of each vertex is three. Therefore, we add to each vertex $(D_\mathrm{e})^{80} - 3$ self loops (this also guarantees that each connected component of G_{reg} is aperiodic). The vertices s' and t' are arbitrary vertices from the cycles that correspond to s and t. A path from s' to t' in G_{reg} can easily be projected down to a path from s to t in G.

The next step is a reduction of G_{reg}, s', t' to a new input $G_{\mathrm{exp}}, s'', t''$ of REGULAR DIGRAPH FIND PATH, such that each connected component of G_{exp} is an expander (and in particular has a logarithmic diameter), and s' and t' are connected in G_{reg} if and only if s'' and t'' are connected in G_{exp}. Furthermore, this is a log-space reduction and a path from s'' to t'' in G_{exp} can be translated in log-space into a path from s' to t' in G_{reg}. This step is the heart of the algorithm, and it essentially completes the algorithm. All that is left to do is enumerate all logarithmically-long paths from s'' in G_{exp} and output one of them if it reaches t'' (after translating it in two steps to a path from s to t in G).

The transformation from G_{reg} to G_{exp} is defined recursively. Set G_0 to equal G_{reg}, and for $i > 0$ define G_i recursively by the rule:

$$G_i = (G_{i-1} \textcircled{z} H)^{40}.$$

Finally, define $G_{\text{exp}} = G_\ell$ for $\ell = O(\log(N \cdot D))$ (that will be determined by the analysis). It follows inductively that each G_i is a $(D_{\text{e}})^{80}$-regular digraph over $[N] \times [D] \times [(D_{\text{e}})^{80}]^i$. In particular, the zig-zag product of G_i and H is well defined. In addition, since D_{e} is a constant, and ℓ is logarithmic then G_ℓ has $poly(N \cdot D)$ vertices.

Assume that G_{reg} is connected, then by Lemma A.7, $\lambda(G_{\text{reg}}) \leq 1 - 1/\text{poly}(N \cdot D)$. By Lemma A.4 and Theorem A.8 (properties of powering and the zig-zag product for regular digraphs), we have that unless $\lambda(G_i)$ is already smaller than some fixed constant then $\lambda(G_i) \leq (\lambda(G_{i-1}))^2$. This means that for some $\ell = O(\log(N \cdot D))$, we have that $\lambda(G_\ell)$ is guaranteed to be smaller than some fixed constant. In other words, G_{exp} is an expander. What if G_{reg} has several connected components? Since both powering and the zig-zag product operate separately on each connected component, we have that for every $S \subseteq [N] \times [D]$, if S contains the vertices of a connected component of G_{reg} then $S \times [(D_{\text{e}})^{80}]^\ell$ contains the vertices of a connected component of G_{exp}, and the subgraph of G_{exp} induced by these vertices is an expander. By this argument, it is natural to select s'' to be any vertex in $\{s'\} \times [(D_{\text{e}})^{80}]^\ell$ and similarly regarding t''. This choice indeed satisfies the requirements of the reduction.

It remains to argue that the transformation of G_{reg} to G_{exp} is log-space and that a path on G_{exp} translates in log-space into a path on G_{reg}. The intuition is that taking a step on G_i translates to a constant number of operations, some of which are taking a step on G_{i-1} and the rest require constant space. As the space used for each one of these operations can be reused for the subsequent operations, the space needed to walk on G_i is only larger by a constant than the space needed to walk on G_{i-1}. Furthermore, this evaluation in particular translates a step on G_i to a path of constant length between the corresponding vertices of G_{i-1}. The space-efficiency requirements follow by induction. $\square$

A.6 Proof of Theorem 6.2

Consistent labelling is the weakest restriction for which efficiently constructible universal traversal sequences are known *even for undirected expander graphs* [HW]. For general undirected graphs, the st-connectivity algorithm of [Rei] gives efficiently constructible universal traversal sequences, but these require an even stronger restriction on the labelling. So in fact, the generalization to regular digraphs is useful even from the point of view of undirected graphs.

Our first step is to argue that the universal traversal sequences for expanders given by Hoory and Wigderson [HW] can be extended to the case of *directed* expanders.

Lemma A.10 *For every two constants D and λ where D is a positive integer and $\lambda < 1$, there exists a log-space algorithm that on input 1^N produces a universal traversal sequence for all connected, consistently labelled D-regular digraphs G on N-vertices with $\lambda(G) \leq \lambda$.*

Proof Sketch: The universal traversal sequence of Hoory and Wigderson [HW] works just as well in the regular case. The only properties used in their analysis are that (1) A walk that starts at two distinct vertices and follows the same set of labels ends in two distinct vertices (this is where the consistent labelling is used). (2) For two sets of vertices A and B one of size K and the other of size $N-K$, either the intersection $A \cap B$ or the number of edges from A to B are $\Omega(\min\{K, N-K\})$ (this is where the expansion is used). Both of these properties also hold in the regular case. $\square$

Now we proceed to construct our universal traversal sequences.

Proof Sketch: (of Theorem 6.2) Consider some connected, consistently labelled D-regular digraphs G on N-vertices. We will show a log-space algorithm $\mathcal{A}$ that produces a universal traversal sequence for $\{G\}$.

We will then argue that the algorithm does not need access to G which will imply the theorem (as the output of $\mathcal{A}$ will be good for any such graph G).

The crucial observation is that, as noted above, given a consistently labelled graph G, we can assume without loss of generality that every edge (u,v) has the same label as an out going edge from u and as an incoming edge to v. Observe that, for the purpose of universal traversal sequence, the only labels that matter are the outgoing labels from each vertex (the incoming labels, which define the rotation map of the graph, are ignored during the walk - therefore any legal labelling will do). In other words, we can assume without loss of generality that whenever $(u,j) = \mathrm{Rot}_G(v,i)$ we have that $i = j$. From now on, our proof follows the same lines as the construction of universal traversal sequence in [Rei], and is therefore only sketched here.

Consider the two graphs G_{reg} and G_{exp} that are obtained from G (and *do depend* on Rot_G) in the proof of Theorem 5.1. By the analysis in that proof, G_{exp} is an expander. Furthermore, as powering and the zig-zag product preserve the property of consistent labeling, we have that G_{exp} is also consistently labelled. Lemma A.10 now implies that there exists a universal traversal sequence $\vec{a}$ for $\{G_{\mathsf{exp}}\}$ and its log-space construction is independent of G. Now consider the walk on G_{exp}, following $\vec{a}$ from some vertex $(s, 1^{\ell+1})$, where $s \in [N]$. This walk covers all of the vertices of G_{exp}. By the construction of G_{exp}, the sequence of labels $\vec{a}$ can be translated in log-space (again, without access to G) into a sequence $\vec{b}$ of labels, such that the walk from $(s,1)$ (for any $s \in [N]$) which follows these labels, visits all the vertices of G_{reg}.

The next step is to translate $\vec{b}$ into a universal traversal sequence for $\{G\}$. Consider the walk from $(s,1)$ on G_{reg}. We want to simulate this walk without knowing s and without access to G. On the other hand, at each step all we want to know is a value $c \in [D]$ such that we are now at some vertex (v,c). To begin with c is set to one. It is easy update c (one up or one down) when taking a step on one of the cycles in the definition of G_{reg}. Labels that correspond to self loops can be ignored. We are left with edges that cross between two different cycles (that correspond to two vertices of G). By our assumption above, in such a case c remains unchanged. Furthermore, the values of c when an edge between cycles is taken, are exactly the labels of edges in G that are traversed by the projection on G of the walk defined by $\vec{b}$. To conclude, the sequence $\vec{c}$ is simply the sequence of values of c in the simulation described above, when edges between cycles are traversed. $\square$

A.7 Proof of Theorem 6.3

Proof Sketch: Let G be a *consistently labelled* $(N, D, 1-\gamma)$ regular graph and s any vertex of G. We will construct a distribution on a sequence of labels such that taking a walk from s on G according to these labels, ends at a vertex that is distributed δ-close to uniform (in variation distance). Since the distribution of labels will be independent of G and s (and will only depend on N, D, δ, and γ) this will imply a pseudorandom generator.

As in the proof of Theorem 5.1, we consider in our analysis two additional graphs G_{reg} and G_{exp}. Their definition will be slightly modified here. First, G_{reg} will be obtained by a zig-zag product (or a replacement product) with a constant degree expander on D vertices. Adding self loops we get an $(ND, (D_{\mathsf{e}})^{80}, 1 - \Omega(\gamma))$ regular graph. The advantage of doing that (instead of a replacement product with a cycle as in the proof of Theorem 5.1), is that the eigenvalue gap of G_{reg} is only smaller by a constant than the eigenvalue gap of G. We now define G_{exp} similarly to the proof of Theorem 5.1, by recursively applying the zig-zag product and powering. However, since we start with a stronger guarantee on the eigenvalue gap of G_{reg} we only need $\ell = O(\log(1/\gamma))$ levels of recursion to bring spectral gap to a constant. The size of the final expander G_{exp} is thus $N_{\mathrm{fin}} = N \cdot D \cdot 2^{O(\ell)} = ND \cdot \mathrm{poly}(1/\gamma)$.

Consider now a random walk of length $m_{\text{fin}} = O(\log(N_{\text{fin}}/\delta)) = O(\log(ND/\delta\gamma))$ in G_{exp}. Such a walk starting from any vertex in the vertices in G_{exp} which correspond to s will converge to the uniform distribution on the vertices of G_{exp}, up to variation distance δ. As in the proof of Theorem 6.2, this walk projects to a walk on G. Since the uniform distribution on vertices of G_{exp} projects to the the uniform distribution on vertices of G, we get that the walk in G also converges to the uniform distribution on the vertices of G_{exp}, up to variation distance δ. As in the proof of Theorem 6.2, we note that we can assume without loss of generality that in the rotation map of G the label of an edge (u, v) is identical both as an outgoing edge from u and as an incoming edge to v. This implies (as in that proof), that the edge labels taken by the walk on G are actually independent of G and s and can be computed in the required small space, just knowing N, D, γ, and δ.

We make the following observations:

- The randomness required is $r = O(m_{\text{fin}}) = O(\log(ND/\delta\gamma))$.
- The walk length is $\ell = m_{\text{fin}} \cdot 2^{O(\ell)} = \log(ND/\delta\gamma) \cdot \text{poly}(1/\gamma) = \log(ND/\delta) \cdot \text{poly}(1/\gamma)$.

□

A.8 Proof of Theorem 7.1

We will prove Theorem 7.1 by showing that for every poly-mixing graph G, there exists a regular digraph G_{reg} such that the correctness of the generator on G_{reg} implies the correctness of (a modification of) the generator on G. Thus, if we have a generator that works well on regular digraphs, we obtain a generator that works well on instances of POLY-MIXING S-T CONNECTIVITY, which we have shown to be **RL**-complete (Theorem 3.1). The construction of G_{reg} from G is given by the following lemma. We stress that this construction is only done in the *analysis*, and thus need not be computable in log-space.

Lemma A.11 *There is a universal constant c such that the following holds. Let $G = (V, E)$ be any d-outregular graph on n vertices with vertices $s, t \in V$ and stationary distribution π such that $\pi(s) \geq 1/k$, $\pi(t) \geq 1/k$, and $\lambda_\pi(G) \leq 1 - 1/k$. Then for every $\varepsilon > 0$, if we set $N_{\text{reg}} = (ndk/\varepsilon)^c$, $D_{\text{reg}} = c \cdot N_{\text{reg}}/\varepsilon$, $\gamma = 1/(ndk)^c$, there is a $(N_{\text{reg}}, d \cdot D_{\text{reg}}, 1 - \gamma)$-regular digraph G_{reg} such that the following holds. The vertex set of G_{reg} can be decomposed into "clouds" $V_{\text{reg}} = \bigcup_{v \in V} C_v$ with $|C_s|, |C_t| \geq |V_{\text{reg}}|/2k$. There is a bad set of edge labels $B \subseteq [d] \times [D_{\text{reg}}]$ of density ε such that for every $u \in V$, vertex $\hat{u} \in C_u$ and edge label $(i, j) \in ([d] \times [D_{\text{reg}}]) \setminus B$, the (i, j)'th neighbor of $\hat{u}$ in G_{reg} is in cloud C_v where v is the i'th neighbor of u in G_{reg}.*

Before proving this lemma, let's see how it implies Theorem 7.1.

Proof of Theorem 7.1: Let $(G, s, t, 1^k)$ be any instance of POLY-MIXING FIND PATH, where G is d-outregular, has n vertices, and has (promised) stationary distribution π with $\pi(s), \pi(t) \geq 1/k$ and $\lambda_\pi(G) \leq 1 - 1/k$. Set $\delta = 1/2k$, and $\varepsilon = 1/(ndk)^b$ for a large constant b to be specified later, and let $N_{\text{reg}} = (ndk/\varepsilon)^c$, $D_{\text{reg}} = c \cdot N_{\text{reg}}/\varepsilon$, $\gamma = 1/(ndk)^c$ be the parameters of the regular digraph guaranteed by Lemma A.11. Let $\text{PRG} = \text{PRG}_{N_{\text{reg}}, d \cdot D_{\text{reg}}, \delta, \gamma} : \{0,1\}^r \to ([d] \times [D_{\text{reg}}])^\ell$ be the generator hypothesized in Theorem 7.1, with seed length $r = a \log(N_{\text{reg}} D_{\text{reg}}/\delta\gamma) = O(abc \log(ndk))$. and walk length

$$\ell = (1/\gamma\delta)^a \cdot (N_{\text{reg}} \cdot dD_{\text{reg}})^\alpha = (ndk)^{O(ac)} \cdot (ndk/\varepsilon)^{O(\alpha c)} = (ndk)^{O(ac)}/\varepsilon^{O(\alpha c)}$$

Without loss of generality we may assume that each component in $\text{PRG}(U_r)$ is uniformly distributed in $[d] \times [D_{\text{reg}}]$. (Shift each component of the output by adding a random element $s \leftarrow [d] \times [D_{\text{reg}}]$. This only increases the seed length by a constant factor and preserves the pseudorandomness of the output because it is equivalent to shifting all labels in the regular digraph by $-s$.)

The algorithm for Poly-Mixing Find Path works as follows. We enumerate the $2^r = (nkd)^{O(abc)}$ seeds of PRG, for each obtaining a walk $\hat{w} \in ([d] \times [D_{\mathrm{reg}}])^\ell$ of length $\ell = (nkd)^{O(abc)}$. Taking the first components of each step in $\hat{w}$, we obtain an induced walk $w \in [d]^\ell$, which we perform on G, starting at s. If any of these walks end at t, we output that walk.

To analyze this algorithm, we consider a walk $\hat{w} \leftarrow \mathrm{PRG}(U_r)$ taken in G_{reg}, starting at any vertex of C_s. Since $\lambda(G_{\mathrm{reg}}) \leq 1 - \gamma$, C_t has density at least $1/2k$, and $\delta = 1/2k$, such a walk will end in C_t with probability at least

$$(1/\delta\gamma)^a \cdot (N_{\mathrm{reg}} \cdot dD_{\mathrm{reg}})^\alpha = \varepsilon^{O(\alpha c)}/(ndk)^{O(ac)}.$$

We now argue that the induced walk w in G will end at t with nearly the same probability. By the properties of G_{reg}, this will be the case provided the walk $\hat{w}$ does not use any edge label from B. Since B has density at most ε and each edge label in $\hat{w}$ is uniformly distributed, the probability any label from B is used is at most

$$\ell \cdot \varepsilon = (ndk)^{O(ac)} \cdot \varepsilon^{1-O(\alpha c)}.$$

Thus the walk w in G ends at t with probability at least

$$\frac{\varepsilon^{O(\alpha c)}}{(ndk)^{O(ac)}} - (ndk)^{O(ac)} \cdot \varepsilon^{1-O(\alpha c)} > 0,$$

provided $\alpha \leq c/\kappa$ and $\varepsilon \leq (1/ndk)^b$ for a $b > \kappa ac$, where κ is a sufficiently large universal constant. In particular, there exists a seed of PRG that will produce a walk from s to t. ■

Defining the regular digraph G_{reg}

Proof of Lemma A.11: Let n be the number of vertices in G, d the out-degree of G, and $\pi = \pi_s$ be the stationary distribution of G (actually the induced subgraph on vertices reachable from s). By adding self-loops and applying Lemma 2.2, we may assume that G has the following properties:

1. $\pi(s) \geq 1/k$, $\pi(t) \geq 1/k$.
2. At least half of the edges leaving each vertex are self-loops.
3. For any vertex v reachable from s, a random walk of length $\ell = O(n \cdot k^3 \cdot \log d)$ from v visits s with probability at least $1/2$.

(For Item 3, we note that $\pi(v) \geq d^{-n}$, so Lemma 2.2 says that a walk of length $O((1/(1-\lambda_\pi(G))) \cdot \log(2/\sqrt{\pi(v)\pi(s)})) = O(k \cdot (n \log d + k))$ ends at s with probability at least $\pi(s)/2 \geq 1/2k$. Repeating $O(k)$ times increases the probability to 1/2.)

The desired regular digraph G_{reg} will essentially be a blow-up of G, with each vertex of G repeated a number of times proportional to its stationary probability, with small "corrections" to remove low-probability vertices and to fix slight irregularities (due to round-off errors).

We construct G_{reg} in several phases.

Step 1: Make all state probabilities nonnegligible. Let ε be the given error parameter. Without loss of generality, we will assume that $\varepsilon < 1/\mathrm{poly}(n, k, d, \ell)$ for a polynomial to be specified later. Then let $D' = \mathrm{poly}(n, \ell, 1/\varepsilon)$ for a polynomial to be specified later. Define a graph $G' = (V, E')$ on the same vertex set as G, but with degree $d \cdot D'$. For every vertex v and edge label $(i, i') \in [d] \times [D']$, we set the (i, i')'th neighbor of v in G' to be the i'th neighbor of v in G, except that we modify up to n of the edges leaving s in order to ensure that every vertex reachable from s has at least one incoming edge directly from s. (The edges to modify should be chosen so as to maintain the property that at least half of the edges from s are self-loops.) Thus a random step on G' is identical to a random step on G, except with probability at most n/D' when at vertex s.

Observe that Property 3 of G also holds in G', because any walk from a vertex v in G that visits s also visits s in G'. Thus, by Lemma A.1, we have $\lambda_{\pi'}(G') \leq 1 - 1/8\ell^2$ for some stationary distribution π'. Moreover, if we take $\ell' = O(\ell^2 \log(\ell/\varepsilon))$, then by Lemmas 2.2 and 2.3, a random walk of length ℓ' from s in G (resp., G') ends at a vertex distributed ε-close to π (resp., π'). Thus,

$$\begin{aligned}
\pi'(t) &\geq \Pr[\text{r.w. in } G' \text{ of length } \ell' \text{ from } s \text{ ends at } t] - \varepsilon \\
&\geq \Pr[\text{r.w. in } G \text{ of length } \ell' \text{ from } s \text{ ends at } t] - \ell' \cdot (n/D') - \varepsilon \\
&\geq \pi(t) - \varepsilon - \varepsilon - \varepsilon \\
&\geq 1/2k,
\end{aligned}$$

provided we take $\varepsilon \leq 1/6k$ and $D' \geq \ell' n/\varepsilon$. Similarly, we have $\pi'(s) \geq 1/2k$. And for every vertex v reachable from s, we have $\pi'(v) \geq (1/2k) \cdot (1/D')$ since there is at least one edge from s to v.

To summarize, we have established the following properties of $G' = (V, E')$:

1. For any vertex v reachable from s, a random walk of length $\ell = O(n \cdot k^3 \cdot \log d)$ from v visits s with probability at least $1/2$.
2. $\lambda_{\pi'}(G') \leq 1 - 1/8\ell^2$.
3. $\pi'(s) \geq 1/2k$, $\pi'(t) \geq 1/2k$.
4. At least half of the edges leaving each vertex are self-loops.
5. For every vertex v reachable from s, $\pi'(v) \geq 1/(2kD')$
6. For every vertex v and every edge label $(i, i') \in [d] \times [D']$, the (i, i')'th neighbor of v in G' equals the i'th neighbor of v in G, unless $v = s$ and $(i, i') \in B'$ where $B' \subseteq [d] \times [D']$ is a set of labels of density at most $n/D' \leq \varepsilon$.

Step 2: Blow up G' to a nearly regular digraph G'' We blow up each vertex v of G' to a "cloud" C_v consisting of $N_v = \lceil \pi'(v) N \rceil$ vertices, for a sufficiently large $N = O(kD'/\varepsilon)$. By Property 5 of G', we have $N_v \in [\pi'(v) \cdot N, (1 + \varepsilon) \cdot \pi'(v) \cdot N]$. The vertex set of G'' is $V'' = \bigcup_v C_v$ for a total of $N'' = \sum_v N_v \in [N, (1 + \varepsilon) \cdot N]$ vertices. Every vertex in G'' has degree $d \cdot D' \cdot D''$, for a sufficiently large $D'' = O(N/\varepsilon)$. For $(i, i', i'') \in [d] \times [D'] \times [D'']$, the (i, i', i'')'th edge leaving any vertex in C_u goes to the $(i'' \bmod N_v)$'th vertex of C_v, where v is the (i, i')'th neighbor of u in G'.

We now argue that G'' is nearly biregular, in the sense that all of the indegrees are close to $d \cdot D' \cdot D''$. Consider any vertex $\hat{v}$ in cloud C_v. Each edge (u, v) in G' induces either $N_u \cdot \lfloor D''/N_v \rfloor$ or $N_u \cdot \lceil D''/N_v \rceil$

edges into $\hat{v}$. Note that $D''/N_v \geq D''/((1+\varepsilon)N) \geq 1/\varepsilon$, if we choose $D'' \geq (1+\varepsilon)N/\varepsilon$. So the indegree of $\hat{v}$ is at most

$$\begin{aligned}
\sum_{(u,v)\in E'} N_u \cdot \left(\frac{D''}{N_v}+1\right) &\leq \sum_{(u,v)\in E'} N_u \cdot (1+\varepsilon)\cdot \frac{D''}{N_v} \\
&\leq \sum_{(u,v)\in E'} [(1+\varepsilon)\pi'(u)N]\cdot(1+\varepsilon)\cdot\frac{D''}{\pi'(v)N} \\
&= \frac{(1+\varepsilon)^2 D''}{\pi'(v)} \cdot \sum_{(u,v)\in E'} \pi'(u) \\
&= \frac{(1+\varepsilon)^2 D''}{\pi'(v)} \cdot (d\cdot D' \cdot \pi'(v)) \\
&= (1+\varepsilon)^2 \cdot d \cdot D' \cdot D'' = (1+O(\varepsilon))\cdot d\cdot D' \cdot D''
\end{aligned}$$

By Property 3 of G', we observe that $|C_s| \geq \pi'(s)N \geq (1+\varepsilon)\cdot N''/2k$ and similarly $|C_t| \geq (1+\varepsilon)\cdot N''/2k$.

We now enumerate the properties of G'' established above.

1. Every vertex in G'' has out-degree $d\cdot D'\cdot D''$ and in-degree at most $(1+O(\varepsilon))\cdot d\cdot D'\cdot D''$.

2. For every vertex $\hat{u}$ in cloud C_u and every $(i,i',i'') \in [d]\times\times[D']\times[D'']$, the (i,i',i'')'th edge leaving $\hat{u}$ leads to a vertex $\hat{v}$ in cloud C_v, where v is the (i,i')'th neighbor of u in G'. By Property 6 of G', v also equals the i'th neighbor of u in the original graph G unless $u=s$ and $(i,i',i'')\in B''$, where $B''=B\times[D'']$ is a set of labels of density at most ε.

3. The number of edges between any two such vertices $\hat{u}$ and $\hat{v}$ equals either $e_{uv}\cdot\lfloor D''/N_v\rfloor$ or $e_{uv}\cdot\lceil D''/N_v\rceil$, where e_{uv} is the number of edges between u and v in G'.

4. C_s and C_t are both of density at least $1/2k$.

Step 3: Add edges to G'' to make a regular digraph G_{reg}. Property 1 of G'' implies that we can make the graph biregular by adding $O(\varepsilon\cdot d\cdot D'\cdot D'')$ edges leaving each vertex. Specifically, we obtain a regular digraph G_{reg} on the same vertex set as G'', in which every vertex has outdegree $d\cdot D_{\text{reg}}$ for $D_{\text{reg}}=(1+O(\varepsilon))\cdot D'\cdot D''$. Each edge leaving a vertex has a label $(i,j)\in[d]\times[D_{\text{reg}}]$, and the edges with $j\leq D'\cdot D''$ are identical to the edges of G''. We let $B_{\text{reg}}=[d]\times([D_{\text{reg}}]\setminus[D'\cdot D''])$ be the set of remaining edge labels.

Let π_{reg} denote the uniform distribution on the set of vertices reachable from C_s. Since G_{reg} is biregular, this is a stationary distribution for G_{reg}. We now enumerate the properties of G''.

1. The vertex set of G_{reg} is $V_{\text{reg}}=\bigcup_{v\in V} C_v$, and the outgoing edges are labelled by elements of $[d]\times[D_{\text{reg}}]$

2. G_{reg} and G'' differ in at most $O(\varepsilon d D_{\text{reg}})$ edges leaving and entering each vertex.

3. C_s and C_t are both of density at least $1/2k$.

4. There is a set $B\subseteq[d]\times[D_{\text{reg}}]$ of density $O(\varepsilon)$ such that for every vertex $\hat{u}\in C_u$ and every edge label $(i,j)\in([d]\times[D_{\text{reg}}])\setminus B$, the (i,j)'th neighbor of $\hat{u}$ in G_{reg} is in cloud C_v where v is the i'th neighbor of v in G. (Namely, take $B=B_{\text{reg}}\cup B''$.)

5. For every $\hat{s} \in C_s$, we have $\lambda_{\pi_{\text{reg}}}(G_{\text{reg}}) \leq 1 - 1/16\ell^2$.

All of these items follow from the previous discussion, except Property 5 bounding the expansion, which we proceed to do below.

Step 4: Analyze expansion of regular digraph. For this, it is useful to introduce a third Markov chain G''' on vertex set $V''' = V'' = V_{\text{reg}}$, which is more closely related to random walks on G'. From any vertex $\hat{u} \in C_u$, the Markov chain G''' chooses a random neighbor v of u in G', and goes to a uniformly selected vertex $\hat{v} \in C_v$. It can be verified that the distribution π''' that assigns each vertex $\hat{v} \in C_v$ probability mass $\pi'''(\hat{v}) = \pi(v)/N_v$ is stationary for G'''. Moreover,

$$\lambda_{\pi'''}(G''') = \lambda_{\pi'}(G') \leq 1 - \frac{1}{8\ell^2},$$

for any $\hat{s} \in C_s$.

We use this fact, and the fact that M''' is "close" to G_{reg} to bound $\lambda(G_{\text{reg}})$. Specifically, let M_{reg}, M'', and M''' denote the transition matrices for G_{reg}, G'', and G''', respectively. Let $\rho = D_{\text{reg}}/(D' \cdot D'') = 1 + O(\varepsilon)$ be the ratio between the degrees of G_{reg} and G''. We consider the two "error" matrices $\mathcal{E}_1 = \rho M_{\text{reg}} - M''$, and $\mathcal{E}_2 = M'' - M'''$. To bound $\lambda_{\pi_{\text{reg}}}(G_{\text{reg}})$, let x be any vector whose support is reachable from C_s such that $\langle x, \pi_{\text{reg}} \rangle_{\pi_{\text{reg}}} = 0$, i.e. $\sum_i x_i = 0$. We need to show that $\|M_{\text{reg}}x\|_{\pi_{\text{reg}}} \leq \lambda_{\text{reg}} \cdot \|x\|_{\pi_{\text{reg}}}$, where $\lambda_{\text{reg}} = 1 - 1/16\ell^2$. Note that since π_{reg} is uniform, $\|\cdot\|_{\pi_{\text{reg}}}$ is simply a scaling of the standard Euclidean norm. We bound $\|M_{\text{reg}}x\|_{\pi_{\text{reg}}}$ as follows.

$$\|M_{\text{reg}}x\|_{\pi_{\text{reg}}} \leq \|\rho M_{\text{reg}}x\|_{\pi_{\text{reg}}} \leq \|M'''x\|_{\pi_{\text{reg}}} + \|\mathcal{E}_1 x\|_{\pi_{\text{reg}}} + \|\mathcal{E}_2 x\|_{\pi_{\text{reg}}}.$$

We bound each term separately. To bound the first, we first observe that the norms $\|\cdot\|_{\pi_{\text{reg}}}$ and $\|\cdot\|_{\pi'''}$ differ by a factor of at most $(1+\varepsilon)$, because π_{reg} and π''' almost identical. Specifically, for every vertex $\hat{v} \in C_v$, we have $\pi'''(v) = \pi'(v)/N_v$, $\pi_{\text{reg}}(v) = 1/N_{\text{reg}}$. These two quantities can be related as follows.

$$\frac{\pi'(v)}{N_v} \geq \frac{1}{(1+\varepsilon)N} \geq \frac{1}{(1+\varepsilon)N_{\text{reg}}}$$

and

$$\frac{\pi'(v)}{N_v} \leq \frac{1}{N} \leq \frac{1+\varepsilon}{N_{\text{reg}}}.$$

Thus, $\pi_{\text{reg}}(\hat{v}) \leq (1+\varepsilon) \cdot \pi'(\hat{v})$ and $\pi'(\hat{v}) \leq (1+\varepsilon) \cdot \pi_{\text{reg}}(\hat{v})$. This implies that the corresponding norms differ by a factor of at most $(1+\varepsilon)$. Therefore,

$$\begin{aligned}
\|M'''x\|_{\pi_{\text{reg}}} &\leq (1+\varepsilon) \cdot \|M'''x\|_{\pi'''} \\
&\leq (1+\varepsilon) \cdot \left(1 - \frac{1}{8\ell^2}\right) \cdot \|x\|_{\pi'''} \\
&\leq (1+\varepsilon)^2 \cdot \left(1 - \frac{1}{8\ell^2}\right) \cdot \|x\|_{\pi_{\text{reg}}}.
\end{aligned}$$

For the second term, involving $\mathcal{E}_1$, we note that $\mathcal{E}_1$ equals $1/(dD'D'')$ times the adjacency matrix A of $G_{\text{reg}} \setminus G''$. Every vertex in this graph has outdegree $dD_{\text{reg}} - dD'D'' = \Theta(\varepsilon dD'D'')$, and indegree at most $O(\varepsilon dD'D'')$ (by Property 2). This implies that $\|Ax\|_{\pi_{\text{reg}}} \leq O(\varepsilon dD'D'') \cdot \|x\|_{\pi_{\text{reg}}}$. (One way to see this is to consider the the vector y assigning each *edge* (u, v) in $G_{\text{reg}} \setminus G''$, the value x_u. The squared length of y

equals the squared length of x times the outdegree $\Theta(\varepsilon d D' D'')$. Then we obtain Ax by summing the entries of y incoming at each vertex. By Cauchy-Schwartz, this increases squared length by at most the maximum indegree $O(\varepsilon d D' D'')$.) Therefore, we have

$$\|\mathcal{E}_1 x\|_{\pi_{\mathrm{reg}}} =\leq \frac{1}{dD'D''} \cdot \|Ax\|_{\pi_{\mathrm{reg}}} = O(\varepsilon) \cdot \|x\|_{\pi_{\mathrm{reg}}}.$$

Finally, we consider the third term, involving $\mathcal{E}_2$. We argue that each entry of $\mathcal{E}_2 = M'' - M'''$ is small. For vertices $\hat{u} \in C_u$, $\hat{v} \in C_v$, the $(\hat{u}, \hat{v})$'th entry of M''' equals $(e_{uv}/(dD')) \cdot (1/N_v)$, by definition of G''', where e_{uv} is the number of edges between u and v in G'. On the other hand, by Property 3 of G'', the $(\hat{u}, \hat{v})$'th entry of M'' is in the interval $[e_{uv} \cdot \lfloor D''/N_v \rfloor/(dD'D''), e_{uv} \cdot \lceil D''/N_v \rceil/(dD'D'')]$, which is contained in the interval $[e_{uv}/(dD'N_v) - 1/D'', e_{uv}/(dD'N_v) + 1/D'']$, since $e_{uv} \leq dD'$. Thus, each entry of $\mathcal{E}_2$ has absolute value at most $1/D''$. This implies that

$$\|\mathcal{E}_2 x\|_{\pi_{\mathrm{reg}}} \leq \frac{\sqrt{N''}}{D''} \cdot \|x\|_{\pi_{\mathrm{reg}}} \leq \varepsilon \cdot \|x\|_{\pi_{\mathrm{reg}}},$$

where the last inequality comes by recalling that $N'' \leq (1+\varepsilon) \cdot N$ and $D'' \geq N/\varepsilon$.

Putting all of the above together, we have

$$\frac{\|M_{\mathrm{reg}} x\|_{\pi_{\mathrm{reg}}}}{\|x\|_{\pi_{\mathrm{reg}}}} \leq (1+\varepsilon)^2 \cdot \left(1 - \frac{1}{8\ell^2}\right) + O(\varepsilon) + \varepsilon \leq 1 - \frac{1}{16\ell^2},$$

provided $\varepsilon \geq c \cdot \ell^2$ for a sufficiently large constant c. ■

A.9 Combinatorial Measures

Other ways in which we can measure progress rather than spectral gaps are combinatorial measures such as edge expansion or vertex expansion.

Edge expansion is roughly preserved in the replacement product, but can deteriorate quite a bit when the graph is powered.

Theorem A.12 *Let $G = (V, E)$ be a directed graph with n edges, such that every vertex has outdgree D_{out} and every indegree is at most D. Let ϵ be the edge expansion of G. Let H be a biregular directed graph with D vertices, degree d, and edge expansion δ. Then $G' := G \textcircled{r} H$ has edge expansion at least*

$$\frac{1}{4} \cdot \epsilon \cdot \frac{D_{\mathrm{out}}}{D} \cdot \min\left\{\frac{1}{d+1}, \frac{\delta d}{d+1}\right\}$$

Concretely, we would use the replacement product using an inner graph H of constant degree and constant expansion, and D_{out} would be close to D in the outer graph, so that the expansion of $G \textcircled{r} H$ would be $\Omega(\epsilon)$.

Proof: [Of Theorem A.12] Recall that, for a vertex v of G, the *cloud of* v is a set C_v of D vertices of G' that "correspond to" to v in the replacement product.

Let A be a set of less than $nD/2$ vertices of G'. We want to prove that there are at least

$$|A| \cdot d \cdot \left(\frac{1}{4} \cdot \epsilon \cdot \frac{D_{\mathrm{out}}}{D} \cdot \min\left\{\frac{1}{d}, \delta\right\}\right)$$

edges from A to $\bar{A}$.

The intuition for the analysis is similar to the intuition in the analysis of the zig-zag graph product in [RVW]: if A is a disjoint union of clouds, then the expansion follows from the expansion of G, and if each cloud contains only a few elements of A then the expansion follows from the expansion of H. For a general set A, our analysis will use the expansion of G if most elements of A are concentrated in "half full" clouds; our analysis will use the expansion of G if most elements of A belong to "half empty" clouds.

Let $B \subseteq A$ be the subset of vertices of A that belong to "half-empty" clouds. That is, a vertex $w \in A$ is in B if it belongs to a cloud C_v such that at most $D/2$ elements of C_v are in A. For an half-empty cloud C_v, define $a_v = |A \cap C_v|$.

We consider the following two cases.

1. If $B > |A|\epsilon D_{\text{out}}/4D$, then each cloud C_v, $v \in S$, contributes at least $a_v \cdot \delta \cdot d$ to the cut between A and $\bar{A}$. (Here we are using the expansion of H.) Overall, the number of edges in the cut is at least

$$\sum_{v \in S} a_v \delta d \geq |B|\delta d \geq |A|\epsilon\delta d D_{\text{out}}/4D$$

2. If $|B| \leq |A|\epsilon D_{\text{out}}/4D$, then let T be the set of vertices v of G such that the cloud C_v contains at least $D/2$ elements of A. (These are the "half-full" clouds.) Note that $|T| \geq (|A| - |B|)/D > |A|/2D$.

   Now we have to consider two sub-cases:

   (a) If $|T| \leq 3n/4$, then we claim that are at least $|A|\epsilon D_{\text{out}}/2D$ edges in G from T to $\bar{T}$. We prove the claim using the expansion of G. If $|T| \leq n/2$, then the number of edges from T to $\bar{T}$ is at least $|T|\epsilon D_{\text{out}} \geq |A|\epsilon D_{\text{out}}/2D$. If $n/2 \leq |T| \leq 3n/4$, then the number of edges from T to $\bar{T}$ is at least $|\bar{T}|\epsilon D_{\text{out}} \geq n\epsilon D_{\text{out}}/4 \geq |A|\epsilon D_{\text{out}}/2D$.

   Those edges correspond to edges in G' that go from a vertex in a half-full cloud to a vertex in a half-empty cloud. We will argue that a reasonable fraction of such edges actually go from vertices in A to vertices in $\bar{A}$.

   We first note that there are at most $|B| \leq |A|\epsilon D_{\text{out}}/4D$ edges in G' going to vertices in A that belong to half-empty clouds. Therefore, there are at least $|A|\epsilon D_{\text{out}}/4D$ edges in G' that have their first endpoint in a half-full cloud and their second endpoint in $\bar{A}$.

   Let us now look at a half-full cloud C_v in G' from which there are, say, k_v outgoing edges whose second endpoint is a vertex in $\bar{A}$ in another cloud, and call $c_v = |C_v - A|$. We note that the cloud contributes at least $(k_v - c_v) + \delta d c_v \geq k_v \min\{1, \delta d\}$ edges to the cut between A and $\bar{A}$. This is because, of the k_v edges leaving C_v and going to a vertex in $\bar{A}$, at least $k_v - c_v$ originate from a vertex in A, and because the number of edges from $A \cap C_v$ to $C_v - A$ in C_v is at least $c_v \delta d$ because of the expansion of H.

   Summing over all the clouds, we get a contribution that is at least

$$\sum_v k_v \min\{1, \delta d\} \geq |A|\epsilon \min\{1, \delta d\} D_{\text{out}}/4D$$

   (b) If $|T| \geq 3n/4$, then we have $3n/4$ or more half-full clouds, each one containing between $D/2$ and D elements of A, even though $|A| \leq nD/2$. This means that of the $|T|$ half-full clouds, at least $n/2$ must contain at most $3D/4$ elements of A. (If we let c be the number of half-full clouds with at most $3D/4$ elements of A, we get $nD/2 \geq |A| \geq c \cdot D/2 + (|T| - c) \cdot 3D/4$, which, together with $|T| \geq 3n/4$, simplifies to $c \geq n/2$.) In each such cloud, the number of edges between A and $\bar{A}$ is at least $D\delta d/4$, so that the total number of edges between A and $\bar{A}$ is at least $nDd\delta/8$, which is at least $|A|d\delta/4$.

■

For directed graphs, as can be seen by the following example, the edge expansion does not necessarily improve by powering.

Proposition A.13 *There is a directed graph G such that for every constant $t > 1$, the edge expansion of G^t is no better than that of G:*

$$\varepsilon(G^t) \leq \varepsilon(G)$$

Proof: We describe an unlabeled graph G because the labels are irrelevant in our case. Let G be the directed path on vertices $\{1, \ldots, 2n\}$ together with an additional edge from every vertex to 1. Formally, the edges of G are $(i, i+1)$ for all $i < 2n$ and also $(i, 1)$ for all i. To make the outdegree 2 everywhere duplicate the edge $(2n, 1)$. The edge expansion of this graph is obtained on the set $A = \{1, \ldots, n\}$. There is exactly one edge leaving this set in G, and since G is strongly connected the edge expansion is $\frac{E(A,\bar{A})}{2|A|} = 1/2n$.

The number edges leaving A in G^t is the number of length-t paths leaving A in G. For $t < n$, this number is equal to t. Since the out-degree of G^t is 2^t, the edge expansion of G (being the minimum over all choices of A) is bounded by $\frac{E(A,\bar{A})}{2^t|A|} = \frac{t}{2^t \cdot n} \leq 1/2n$.

Note that G can easily be made to have bounded in-degree, by 'spreading' the edges pointing to 1 to point somewhere among the first say $n/2$ vertices. ■