

NL/poly \subseteq \oplus L/poly (Preliminary Version)

Avi Wigderson*
The Hebrew University
Jerusalem, Israel

Abstract

This note provides logspace analogs of the results of Valiant-Vazirani. We first show that solving STCONN for graphs with at most one st -path is essentially as hard as the general problem, via a probabilistic logspace reduction. We derive from it a nonuniform logspace reduction from NL to $\oplus L$.

1 Introduction

Valiant and Vazirani [5] gave a randomized reduction from NP to UP , and hence to $\oplus P$, using universal hash functions. It follows that $NP/poly \subseteq \oplus P/poly$.

It was an open problem if a similar result holds in the logspace world, i.e. does $NL/poly \subseteq \oplus L/poly$? One difficulty in applying the Valiant-Vazirani reduction, is that while the computations of hash functions could be easily embedded in CNF formulae (they used SAT as the NP -complete language for their reduction), it is not clear how to embed them in graph reachability or other NL -complete languages.

In this note we observe that the “right” reduction to use is the Isolation Lemma of Mulmuley-Vazirani-Vazirani [4], which can be easily embedded in graph reachability. Thus we give a randomized reduction from NL to UL , from which the $NL/poly \subseteq \oplus L/poly$ easily follows. We note that [4] showed how the Isolation Lemma can be used to rederive the Valiant-Vazirani result for the clique function. It is already clear there that the Isolation Lemma “interacts well” with graphs.

A very interesting question that remains open is whether one can remove the nonuniformity from these inclusions, both for NP and NL . The only result in this direction we are aware of is the observation in [3] that $SL \subseteq \oplus L$, where SL stands for symmetric

logspace (the class of languages logspace reducible to undirected s - t connectivity)

Our main result has an interesting application to the relationship between arithmetic and Boolean circuit depth for (arithmetic and Boolean, respectively) matrix powering. Borodin [1] observed that, given a depth $d(n)$ arithmetic circuit for computing the n th power of an $n \times n$ 0/1 matrix, it can be converted to a depth $d(n) \log d(n)$ Boolean circuit for the transitive closure of that matrix. Moreover, the size blows up only polynomially. The conversion uses arithmetic over small primes and the Chinese Remainder Theorem. The loss in depth comes from a Boolean simulation of the arithmetic operations over the small fields, and it was not clear if this loss is necessary. Our reduction immediately implies that, any depth $d(n)$ arithmetic circuit family as above yields a depth $O(d(n))$ Boolean circuit family for transitive closure. Again, this entails no more than polynomial blow up of size.

2 Preliminaries and Results

2.1 The Isolation Lemma

Let E be a finite set, and let $w : E \rightarrow \mathcal{R}$ be an arbitrary (weight) function. Extend w to subsets of E by $w(S) = \sum_{e \in S} w(e)$. Let \mathcal{F} be a family of subsets of E . For a fixed weight function w , denote by $\min(\mathcal{F}, w)$ the weight of the lightest set in \mathcal{F} under w , and $\text{MIN}(\mathcal{F}, w)$ those subsets in \mathcal{F} whose weight is minimum. Clearly $|\text{MIN}(\mathcal{F}, w)| \geq 1$ for every \mathcal{F} and w . Now the Isolation Lemma of Mulmuley, Vazirani and Vazirani states that choosing w at random appropriately, there will be a unique minimum weight subset in \mathcal{F} with high probability.

Theorem 1 [4] *Fix an integer k . Pick \mathbf{w} at random as follows: for every $e \in E$, $\mathbf{w}(e)$ is chosen uniformly from the integers in $[1, k|E|]$, independently of*

*Partially supported by the Wolfson Research Awards, administered by the Israeli Academy of Sciences and Humanities, and by an American Israeli BSF grant 92-00106

all other elements in E . Then for every \mathcal{F} we have

$$\Pr[|\text{MIN}(\mathcal{F}, \mathbf{w})| > 1] \leq 1/k$$

2.2 Graphs, Languages and Complexity

All graphs and all paths mentioned in this paper are directed. $G = G(V, E)$ will denote a graph with vertices V and directed edges E , and s, t will refer to distinct vertices in V .

We will consider three logspace classes (and their nonuniform versions): NL , $\oplus L$, UL . The first two are captured respectively by their complete (under deterministic logspace reductions) graph reachability languages $STCONN$, $ODD-STCONN$ defined below.

- $(G, s, t) \in STCONN$ iff G contains an $s-t$ path.
- $(G, s, t) \in ODD-STCONN$ iff G contains an odd number of $s-t$ paths.

The class UL is not known to have such complete problems. Nevertheless for our purposes it is captured by any language (which we generically call $UNIQUE-STCONN$) satisfying the following two properties:

1. (G, s, t) has no $s-t$ path implies $(G, s, t) \notin UNIQUE-STCONN$.
2. (G, s, t) has a unique $s-t$ path implies $(G, s, t) \in UNIQUE-STCONN$.

We shall use two notions of reducibilities, performed respectively by functions computable in RL (probabilistic logspace machines) and $L/poly$ (nonuniform logspace). Let T, T' be two languages.

- $T \leq_{L/poly} T'$ if there is a machine $M \in L/poly$ such that for all inputs x , $x \in T \iff M(x) \in T'$.
- $T \leq_{RL}^{\epsilon} T'$ if there is a machine $M \in RL$ such that:

$$\begin{aligned} x \notin T &\Rightarrow M(x) \notin T' \\ x \in T &\Rightarrow \Pr[M(x) \in T'] \geq \epsilon \end{aligned}$$

Finally, we consider arithmetic and Boolean circuits. These are circuits of constant fan-in and fan-out, with size and depth defined as usual. The difference is that arithmetic circuits have gates from the basis $\{+, -, \times\}$ (we disallow division!) over the rational number field, while Boolean circuits have the standard Boolean basis $\{\wedge, \vee, \neg\}$. Let $d_A(p)$ (resp. $d_B(f)$) denote the smallest depth of a polynomial size

arithmetic (resp. Boolean) circuit for the polynomial p (resp. Boolean function f).

As usual, we will be interested in circuit families for families of polynomials and functions. On input an n vertex graph G , presented by a 0/1 matrix, we consider the polynomial $\#STCONN$ computing the number of $s-t$ paths in G . This is the arithmetic analog of the Boolean function $STCONN$.

2.3 Results

Our two main results are:

Theorem 2 $STCONN \leq_{RL}^{1/n^3} UNIQUE-STCONN$

Theorem 3 $STCONN \leq_{L/poly} ODD-STCONN$

A direct corollary of the second theorem is the title of the paper:

Corollary 1 $NL/poly \subseteq \oplus L/poly$

In fact, there is clearly nothing special about counting modulo 2. The same result holds for the $Mod_k L$ classes defined in [2].

Corollary 2 $NL/poly \subseteq \oplus Mod_k L/poly$

Another corollary of Theorem 3 shows that the depth of polynomial size Boolean circuits for $STCONN$ is never worse than that of arithmetic circuits for $\#STCONN$.

Corollary 3

$$d_B(STCONN) = O(d_A(\#STCONN))$$

3 Proofs

3.1 Proof of Theorem 2

We first need some simple definitions and lemmas. Let $G(V, E)$ be a directed graph, and w a weight (distance) function on E . We let $d_w(a, b)$ denote the length of the shortest path between two nodes $a, b \in V$ in this weighted graph, and let W denote the maximum weight $w(e)$ over all $e \in E$. The following lemma follows immediately from the Isolation Lemma, taking the family of subsets \mathcal{F} to be all $s-t$ paths in G .

Lemma 1 *Let G be a graph with an $s-t$ path. Let w be chosen at random with each weight independently taken uniformly from $[1, 2|E|]$. Then*

$$\Pr[\exists \text{ a unique } s-t \text{ path of distance } d_{\mathbf{w}}(s, t)] \geq 1/2$$

Comment: Observe that increasing the range of \mathbf{w} to $|V|^2|E|$ would yield a graph in which (with high probability) the shortest distance between every pair of nodes is achieved by a unique path. We see no application of this observation.

Now given a graph $G(V, E)$, a weight function w and integer l define the (unweighted, layered) graph $G_w^l(U, F)$ as follows. For every vertex $a \in V$ and every integer $0 \leq i \leq l$ put the vertex $\langle a, i \rangle$ in U (i.e. $l + 1$ copies of V , arranged in layers). For every edge $e = (a, b) \in E$ and every $0 \leq i \leq l - w(e)$ put an edge $(\langle a, i \rangle, \langle b, i + w(e) \rangle)$ in F .

Lemma 2 G_w^l can be constructed from input G, w, l in deterministic logspace.

Lemma 3 • If G has no s - t path, then for every w and l , G_w^l has no $\langle s, 0 \rangle - \langle t, l \rangle$ path.

- If G has an s - t path and $l = d_w(s, t)$ then G_w^l has an $\langle s, 0 \rangle - \langle t, l \rangle$ path. Moreover the later path is unique if the shortest weighted s - t path in G is unique.

We conclude theorem 2 from the above lemmas as follows: Given G we construct $G_{\mathbf{w}}^{\mathbf{l}}$ with w picked at random as in lemma 1, and \mathbf{l} chosen uniformly from $[1, 2|V||E|]$. Note that $\Pr[\mathbf{l} = d_{\mathbf{w}}(s, t)] \geq 1/|V|^3$. Also, it is clear that this reduction can be performed in RL , if we are allowed to print the edges of the new graph in the “right” order (i.e in groups of edges related to each weight).

Comment: We caution that the fact that this reduction can be carried in RL does not mean that we can use it as a subroutine in a logspace computation. If any algorithm needs to query some edge of the output graph of this reduction more than once or in the “wrong” order, we need to remember the random bits used for the weights. This will not hurt us later, as we’ll move into nonuniform reductions.

3.2 Proof of Theorem 3

The random reductions of the previous subsection, together with a standard counting argument yield the following.

Lemma 4 Fix $m = n^{10}$. There exists m pairs $\{(w_j, l_j) | 1 \leq j \leq m\}$ satisfying the following for every graph G on n vertices.

- For every j , l_j and the weights in w_j are integers below n^3 .
- If G has no s - t path, then for every j , $G_{w_j}^{l_j}$ has no $(s, 0) - (t, l_j)$ path.

- If G has an s - t path, then there exists j such that $G_{w_j}^{l_j}$ has a unique $(s, 0) - (t, l_j)$ path.

We abbreviate by (G_j, s_j, t_j) the triple $(G_{w_j}^{l_j}, (s, 0), (t, l_j))$.

Now assume we get as an advice (of polynomial length) a set of m pairs satisfying Lemma 4. On input (G, s, t) , we will construct (G', s', t') such that G' has an odd number of s - t paths iff G had an s - t path. This uses a standard idea of adding direct s - t edges to change the parity of the number of s - t paths.

Construct the graphs (G_j, s_j, t_j) for $j \in [m]$. To each of them add the direct edge (s_j, t_j) . Identify t_j and s_{j+1} for every $j < m$. Change the name of s_1 to s' , and of t_m to t' . Finally, add the direct edge (s', t') and call the new graph G' . It is easy to verify that (G', s', t') satisfies the requirement above, and we are done.

3.3 Proof of Corollary 3

First notice that the construction of the graph G' from the input graph G in the proof of Theorem 3 can be easily carried out by an NC^1 circuit, i.e. a Boolean circuit D of polynomial size and $O(\log n)$ depth.

Now assume $d_A(\#STCONN) = d(n)$ (note that $\log n \leq d(n) \leq (\log n)^2$). Let C' be the arithmetic circuit for $\#STCONN$ for n^{10} vertex graphs. Thus C' has polynomial size and depth $d(n^{10}) = O(d(n))$ (using the upper bound on d and assuming d is a reasonably smooth function). Convert C' into a Boolean circuit C simply by thinking of the gates as performing the arithmetic operations over $GF(2)$, and then simulating them with Boolean gates. Thus C also has polynomial size and depth $O(d(n))$.

The Boolean circuit for $STCONN$ we promised is obtained simply by identifying the outputs of the circuit D with the inputs of the circuit C . By Theorem 3 it correctly computes $STCONN$ for n vertex graphs, and by the discussion above it has depth $O(d(n) + \log n) = O(d(n))$ and polynomial size.

Acknowledgements

I thank Richard Beigel, Allan Borodin and Noam Nisan for helpful discussions and comments on earlier versions of this note.

References

- [1] [Bo] A. Borodin. On relating time and space to size and depth *SIAM J. Comput.*, 6: 733–744, 1977.
- [2] [BDHM] G. Buntrock, C. Damm, H. Hertrampf, and C. Meinel. Structure and importance of the logspace-mod class. *Math. Systems Theory*, 25:223–237, 1992.
- [3] [KW] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th Annual Symposium on Structure in Complexity Theory*, 1993.
- [4] [MVV] K. Mulmuley and U. Vazirani and V. Vazirani. Matching is as Easy as Matrix Inversion In *Proc. of the 19th STOC*, pp. 345–354, 1987.
- [5] [VV] L.G. Valiant and V.V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.