Resolution of Hartmanis' Conjecture for NL-hard sparse sets

Jin-yi Cai * D. Sivakumar †
Department of Computer Science,
State University of New York at Buffalo, Buffalo, NY 14260.
Email:{cai,sivak-d}@cs.buffalo.edu

Recently, Cai and Sivakumar [CS95] settled a conjecture of Hartmanis [Har78], showing that unless P = LOGSPACE, no sparse set can be hard for P under logspace many-one reductions. In this note, we settle another conjecture of Hartmanis about sparse hard sets for nondeterministic logspace (NL). We show that if there is a sparse set S that is hard for NL under logspace many-one reductions, then NL = DLOGSPACE.

In what follows, we abbreviate NL by NL and DLOGSPACE by L. The proof builds on the techniques of Cai and Sivakumar [CS95]. As in the case of reductions from P to sparse sets, we will describe a parallel algorithm for an NL-complete problem. The parallel algorithm can be implemented by a polynomial-size, log-depth circuit that makes polynomially many parallel calls to the reduction from NL to the sparse set S. It implies the stronger result that if NL has a sparse hard set under NC^1 many-one reductions, then $\mathrm{NL} = \mathrm{NC}^1$.

Our result improves the result of Cai, Naik and Sivakumar [CNS95], who show that if there is a sparse set S that is hard for NL under logspace many-one reductions, then NL equals Randomized LOGSPACE (where the randomized logspace machine is given two-way access to a random tape). Their proof makes use of a randomized reduction from s-t connectivity to the so-called "unique s-t connectivity" problem. Given a graph G and a pair of vertices s, t, this reduction produces a polynomial number of graphs $G_1, \ldots, G_k$ of polynomial size, together with distinguished vertex-pairs $(s_1, t_1), \ldots, (s_k, t_k)$, that satisfy the following conditions. If there is no path from s to t in G, then no G_i has a path from s_i to t_i; if there is a path from s to t in G, then with high probability, at least one of the G_i's has a *unique* path from s_i to t_i. This reduction is due to Avi Wigderson [Wig94], and it exploits the "isolation lemma" of Mulmuley, Vazirani and Vazirani [MVV87]. In contrast, the proof given below eliminates the need for the unique witness property; instead, it uses the famous Immerman–Szelepcsényi result that NL = co-NL [Imm88, Sze87].

First we note that the s-t connectivity problem for directed acyclic graphs (DAG-STCON) is complete for NL under logspace- (or even NC^1-) computable many-one reduc-

*Research supported in part by NSF grants CCR-9057486 and CCR-9319093, and by an Alfred P. Sloan Fellowship.
†Research supported in part by NSF grant CCR-9409104.

tions. Wlog., we may also assume that all instances of DAG-STCON are labeled and layered graphs, that is, graphs where all edges go from lower-numbered vertices to higher-numbered vertices.

It is known that if m is of the form $2 \cdot 3^\ell$ for some integer $\ell \geq 0$, the polynomial $x^m + x^{m/2} + 1$ is an irreducible polynomial of degree m over $GF(2)$ [vL91]. In the following,, by a finite field $GF(2^m)$, where $m = 2 \cdot 3^\ell$, we refer explicitly to the field $\mathbb{Z}_2[x]/(x^m + x^{m/2} + 1)$. We define the language B that consists of tuples of the form $\langle G, s, t, 1^m, \alpha, \beta \rangle$, where:

(1) $G = (V, E)$ is a directed, layered acyclic graph on n vertices (hence G has at most $k = \binom{n}{2}$ edges, and the adjacency matrix A of G is upper-triangular).

(2) s and t are vertices in G.

(3) m is of the form $2 \cdot 3^\ell$ for some integer $\ell \geq 0$.

(4) $\alpha, \beta \in GF(2^m)$

(5) $\sum_{i=1}^{k} \alpha^i X_{u_i v_i} = \beta$, where for $1 \leq i \leq k$, $1 \leq u_i < v_i \leq n$, (u_i, v_i) denotes the i-th edge in G, and $X_{u_i v_i}$ is a boolean variable that is 1 if and only if there is a path from u_i to t whose first step is the edge (u_i, v_i).

We begin by observing that the language B is complete for NL. DAG-STCON easily reduces to B, hence B is NL-hard. Moreover, testing whether $\langle G, s, t, 1^m, \alpha, \beta \rangle \in B$ requires computing polynomially many predicates X_{uv}. The language $Z = \{\langle G, s, t, u, v \rangle \mid X_{uv} = 1\}$ is in NL, and since NL = co-NL, its complement Z^c is also in NL. The nondeterministic logspace machines for Z and Z^c can be used to build a nondeterministic logspace machine that computes X_{uv} in the following strong sense: every computation either outputs the correct value of X_{uv} or aborts in a "DON'T KNOW" state, and at least one computation is guaranteed to output the correct value of X_{uv}. Using this, it is easy to build a nondeterministic logspace machine for the language B that computes X_{uv} for all $(u, v) \in E$, and accepts $\langle G, s, t, 1^m, \alpha, \beta \rangle$ if and only if $\sum_{i=1}^{k} \alpha^i X_{u_i v_i} = \beta$. In other words, $B \in$ NL.

By hypothesis, $B \leq_m S$. Let f denote the (logspace- or NC^1-computable) function that reduces B to S. We will show how to solve DAG-STCON using f as an oracle. Fix G, s and t. Using the same ideas as in the case of reduction from P to S, we can set up equations and solve for the variables X_{uv} for all $u, v \in V, u < v$. Recall that this scheme produces a polynomial number of sets of solutions, where at least one set of solutions is guaranteed to be correct. It remains to show how to verify the correctness of solutions. Each set of solutions for the X_{uv}'s can be assumed to be in the form of a matrix X of the same dimensions as A, the adjacency matrix of G. By its definition, X_{uv} is 1 only if $(u, v) \in E$. Therefore, for each candidate set of solutions, we will first verify that the condition $X_{uv} \leq A(u, v)$ holds for all $u, v \in V$. It is easy to see that this test can be performed simultaneously on all sets of solutions by a polynomial-size, log-depth circuit. Note that since A is a strictly upper-triangular matrix, every X that passes this test is strictly upper-triangular.

For every $u \in V$, we first compute the boolean variable X_u that is 1 if and only if $u = t$ or there exists some v such that $X_{uv} = 1$. In matrix terms, $X_u = 1$ if and only if $u = t$ or there is at least one 1 in the row corresponding to u in the matrix X. This computation can be easily done by a polynomial-size, log-depth circuit, since it only requires computing the OR of n bits. Next we perform the following local consistency check: for every u such that $X_u = 1$ and for every v such that $X_{uv} = 1$, verify that $X_v = 1$. This test ensures that if X promises a path from u to t with v as the first vertex, then indeed X also promises some path from v to t. Notice that the latter path cannot include the vertex u since G is acyclic. This is important because, otherwise, it is possible that X passes this test by setting $X_{uv} = X_{vu} = 1$ even though t is reachable from neither of u and v. It is amply clear that the local consistency check can be performed by a polynomial-size log-depth circuit.

Finally we argue that there exists a path from s to t if and only if some set of solutions X has $X_s = 1$. Clearly, if there is a path from s to t then the correct solution for X will pass all the tests and have $X_s = 1$.

Next we claim that if X passes all the tests, $X_z = 0$ for all $z > t$. In particular, if $s > t$ but $X_s = 1$ (incorrectly), the local test will catch this. The claim is vacuous if t is the last vertex. otherwise, we prove the claim by induction, starting from the last vertex. By the first test against the adjacency matrix and since the graph is layered, the base case is clear. Assume inductively that $X_z = 0$ for all $t < z_0 < z$. If $X_{z_0} = 1$, then by the definition of X_{z_0}, for some $z > z_0$, $X_{z_0 z} = 1$. However, it then fails the local test, since $X_z = 0$.

To complete the proof, we argue that whenever $X_u = 1$ for some $u \leq t$, there is a path from u to t. The base case, namely $u = t$, is trivial. Suppose $X_u = 1$ for some $u < t$. By definition, there is a vertex v such that $X_{uv} = 1$ and $X_v = 1$. The first test ensures that there is an edge (u, v). If $v = t$, it is clear that there is a path from u to t. If $v < t$, by the inductive hypothesis, there is a path from v to t, which, together with the edge (u, v), gives a path from u to t.

Theorem 1 *If there is a sparse hard set for* NL *under logspace many-one reductions, then* NL = L.

Theorem 2 *If there is a sparse hard set for* NL *under many-one reductions computable in logspace-uniform* NC^1*, then* NL *equals logspace-uniform* NC^1.

Using the techniques of Cai, Naik, and Sivakumar [CNS95], the following can be shown:

Corollary 3 *If there is a sparse hard set for* NL *under logspace-computable randomized many-one reductions with two-sided error, then* NL = RL*, where* RL *is the class of languages accepted by logspace Turing machines with two-way access to the random tape.*

Corollary 4 *If there is a sparse hard set for* NL *under randomized many-one reductions with two-sided error that are computable in logspace-uniform* NC^1*, then* NL *equals logspace-uniform* RNC^1.

Finally, exploiting an idea of Van Melkebeek [Mel95], the following can be shown; similar results hold for randomized bounded truth-table reductions:

Theorem 5 *If there is a sparse hard set for* NL *under bounded truth-table reductions computable in logspace, then* NL = L.

Theorem 6 *If there is a sparse hard set for* NL *under bounded truth-table reductions computable in logspace-uniform* NC^1*, then* NL *equals logspace-uniform* NC^1.

References

[CNS95] J. Cai, A. Naik, and D. Sivakumar. On the existence of hard sparse sets under weak reductions. UBCS–TR 95-31, Computer Science Dept., University at Buffalo, 1995.

[CS95] J. Cai and D. Sivakumar. The resolution of a Hartmanis conjecture. UBCS–TR 95-30, Computer Science Dept., University at Buffalo, 1995. To appear in FOCS 1995.

[Har78] J. Hartmanis. On log-tape isomorphisms of complete sets. *Theoretical Computer Science*, 7(3):273–286, 1978.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17:935–938, 1988.

[Mel95] D. Van Melkebeek. On reductions of P sets to sparse sets. TR 95-06, Computer Science Dept., The University of Chicago, 1995.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[Sze87] R. Szelepcsényi. The method of forcing for nondeterministic automata. *Bull. of the EATCS*, 33:96–100, 1987.

[vL91] J.H. van Lint. *Introduction to Coding Theory.* Springer-Verlag, 1991.

[Wig94] A. Wigderson. $\mathrm{NL}/\mathrm{poly} \subseteq \oplus\mathrm{L}/\mathrm{poly}$. In *Proc. 9th Annual IEEE Conference on Structure in Complexity Theory*, pages 59–62, 1994.