Electronic Colloquium on Computational Complexity, Report No. 23 (1998)

Uniform Inclusions in Nondeterministic Logspace

Eric Allender* Shiyu Zhou†

Abstract

We show that the complexity class LogFew is contained in NL ∩ SPL. Previously, this was known only to hold in the nonuniform setting.

Key Words: Nondeterministic Logspace Computation, Nonuniform Complexity, Derandomization, ϵ-biased Sample Space.

*Department of Computer Science, Rutgers University, New Brunswick, NJ 08854, USA. E-mail: *allender@cs.rutgers.edu*. Supported in part by NSF grant CCR-9509603.

†Bell Laboratories, 700 Mountain Avenue, Murray Hill, NJ 07974, USA. E-mail: *shiyu@research.bell-labs.com*

ISSN 1433-8092

1 Introduction

In [RA97], a probabilistic construction was used to show that the complexity classes NL/poly and UL/poly coincide. That is, in the context of nonuniform complexity, nonuniform logspace is no more powerful than unambiguous logspace. It was observed in [AR98] that the equality NL=UL holds also in the uniform setting, under a plausible hypothesis concerning pseudorandom number generators. However, it remains an important open question whether NL=UL can be established without resorting to unproved assumptions.

The results and techniques of [RA97] were extended in [AR98] in a number of ways. One extension involves the class LogFew, defined in [BDHM92]. (Formal definitions appear below.) No inclusion relation was known between NL and LogFew in the uniform setting, although UL is trivially contained in LogFew, and thus NL/poly $\subseteq$ LogFew/poly. The converse inclusion LogFew/poly $\subseteq$ NL/poly was proved in [AR98], again using a probabilistic argument. In this paper, we show that this probabilistic argument can be derandomized. Thus, LogFew is contained in NL.

Another extension in [RA97] involves the complexity class SPL (also defined below). One of the main results of [AR98] is that the matching problem is in the nonuniform version of SPL. UL is trivially contained in SPL, but no inclusion relation was known between LogFew and SPL (although it follows from [AR98] that LogFew is contained in the nonuniform version of SPL). In this paper, we also show that LogFew is contained in SPL in the uniform setting. It remains unknown if NL is contained in SPL in the uniform setting.

The main point of this paper is that it shows that at least some of the nonuniform inclusions of [RA97, AR98] can be shown to hold also in the uniform setting, by making use of current derandomization techniques.

2 Preliminaries

2.1 Classes in Nondeterministic Logspace

We assume the reader is familiar with NL (Nondeterministic Logspace). The unambiguous version of NL, denoted UL, was first explicitly defined and studied in [BJLR92, AJ93]. A language A is in UL if and only if there is a nondeterministic logspace machine M accepting A such that, for every x, M has at most one accepting computation on input x.

A generalization of unambiguous machines considers machines that have only a "few" accepting computation paths. In [BDHM92], the complexity classes LogFewNL and LogFew were defined. (In [BDHM92], these classes were defined using "weakly unambiguous machines"; for our results we do not need this additional complication. Our results hold even in the stronger setting using the definitions as we present them here.) LogFewNL consists of languages accepted by NL machines having the property that the number of accepting computations is bounded by a polynomial. LogFew is also defined in terms of NL machines M such that $\#acc_M(x)$ is bounded by a polynomial; but now there is also a logspace-computable predicate R such that x is in A if and only if $R(x, \#acc_M(x))$ is true. From the definitions, it is immediate that UL $\subseteq$ LogFewNL $\subseteq$ NL, and LogFewNL $\subseteq$ LogFew. Thus it is immediate from [RA97] that in the nonuniform setting LogFewNL and NL coincide with UL. It is shown in [RA97] that, in the

nonuniform setting, these three classes also coincide with LogFew.

#L is the class of functions that count the number of accepting paths of an NL machine. GapL is the class of functions that are the difference of two #L functions. GapL is of interest because it is precisely the class of functions that are logspace-reducible to computing the determinant of an integer matrix. (See, for example [MV97].)

SPL is the set of all languages A such that the characteristic function χ_A is in GapL. In [RA97], it is shown that the nonuniform version of SPL contains such problems as computing a maximum matching and finding maximum flow in a graph with unary weights. It follows that, at least in the nonuniform setting, SPL contains LogFew. In this paper, we show that this inclusion holds also in the uniform setting.

2.2 ϵ-Biased Distributions

We associate to each $m \times n$ matrix M a probability distribution D_M defined as follows: for each n-vector v, $D_M(v) = k/m$ if there are exactly k different rows in M that are equal to v. Such a matrix M is said to be *over* a vector space $\mathcal{V}$ (of dimension n) if each row of M is a vector in $\mathcal{V}$. In what follows we use q to denote a prime power unless specified otherwise, and use F_q to denote the finite field with q elements.

ϵ-biased distributions have been studied extensively under the name of ϵ-biased sample space for about ten years (see e.g. [Vaz86, NN90, AGHP92, ABNNR92]). They have become one of the main tools for derandomization. The formal definition is as follows:

Definition 2.1 *A probability distribution D on $(F_q)^n$ is said to be* ϵ-biased *if for any non-zero $v \in (F_q)^n$ and any $c \in F_q$,*

$$\mid Pr[< u, v >= c] - 1/q \mid \leq \epsilon/q,$$

where $u \in (F_q)^n$ is randomly chosen according to distribution D, and $< u, v >$ is the inner product of u and v.

We extend this definition to matrices M over $(F_q)^n$ as follows: a matrix M over $(F_q)^n$ is said to be ϵ-*biased* if its associated distribution D_M on $(F_q)^n$ is ϵ-biased. In other words, if we take the product of an ϵ-biased matrix M over $(F_q)^n$ and *any* non-zero vector v in $(F_q)^n$ then, in the resulting vector Mv, the fraction of *any* element in F_q is between $(1-\epsilon)/q$ and $(1+\epsilon)/q$. By definition, a usual ϵ-biased sample space in the literature is an ϵ-biased matrix M over $(F_2)^n$.

2.3 Weight Assignments of a Graph

Let G be a directed graph on n vertices with vertex set V and edge set E.

For a prime power q, a q-*weight assignment* of G is a function

$$A : V \times V \rightarrow F_q.$$

Given such a weight assignment, the *weight of an edge* $(u, v) \in E$ is defined to be $A(u, v)$; and for a set S of edges in G (such as the set of edges on a path in G), the *weight of* S is defined to be the sum of the weights of the edges in S, i.e., $A(S) = \sum_{(u,v)\in S} A(u, v)$, where the summation is taken as field addition.

We will view a set S of edges in G as a vector, denoted also by S for convenience, in $(F_2)^{n^2}$ with coordinates indexed by $V \times V$ such that the (u,v)-th coordinate of S is 1 if and only if $(u,v) \in E$. We will view a q-weight assignment A of G as a vector, denoted also by A, in $(F_q)^{n^2}$ with coordinates indexed by $V \times V$ such that the (u,v)-th coordinate of A is $A(u,v)$. Now the weight of S given by A is by definition $A(S) =< A, S >$. For a given matrix M over $(F_q)^{n^2}$, we may view each row of M as a q-weight assignment of G.

3 Constructing ϵ-biased Distribution in Logspace

In this section we show how to construct a matrix M over $(F_q)^n$ in logspace such that its associated distribution D_M is ϵ-biased.

Theorem 3.1 *There is a deterministic algorithm such that given as input a positive integer n, an $\epsilon \in (0,1]$ and a prime power q, computes an $m \times n$ matrix M over $(F_q)^n$ with $m = (nq\epsilon^{-1})^2$ that is ϵ-biased; moreover, each entry of the matrix can be computed in space $O(\log n + \log q + \log \epsilon^{-1})$.*

The proof of the theorem is essentially given in [AGHP92] (see also [NN90]). We sketch a proof here for completeness and emphasize the space complexity of the construction.

Proof: Let t be an integer to be determined later and let $\psi : F_{q^t} \to (F_q)^t$ be an isomorphic mapping. The rows of matrix M are indexed by $\{(x,y) \mid x,y \in F_{q^t}\}$ (thus the total number of rows in M is $m = q^{2t}$), and the i-th coordinate of the (x,y)-th row in M is defined to be $< \psi(x^i), \psi(y) >$.

As is shown in [AGHP92] for the case $q = 2$, by choosing $t = (\log nq\epsilon^{-1})/(\log q)$ (thus $m = (nq\epsilon^{-1})^2$), the distribution D_M associated to M is ϵ-biased.

Let us examine the space complexity of the construction. It is well-known that we can efficiently encode the field elements of F_{q^t} (and $(F_q)^t$) so that field addition and multiplication can be done in space $O(t \log q)$: mainly what we need for this is an irreducible polynomial of degree t over F_q, which can be constructed in space $O(t \log q)$. We refer the reader to [LN86] for more background on finite fields. Given this fact, it is then easily seen that each entry of the matrix M in our construction is computable in space $O(t \log q)$, which is $O(\log n + \log q + \log \epsilon^{-1})$. □

4 Biased Distributions Give Distinct Weights

Lemma 4.1 *Let G be a directed graph on n vertices and let P be a family of r distinct sets of edges in G. Suppose q is a prime power that is at least $2\binom{r}{2}$ and suppose M is a $\frac{1}{2}$-biased matrix over $(F_q)^{n^2}$. Then, there exists a q-weight assignment of G in M (i.e., a row in M) that gives all r sets of edges in P distinct weights.*

Proof: To prove the lemma we first need the following proposition whose correctness can be easily verified:

Proposition 4.1 *Let $w \in (F_q)^{n^2}$ be a q-weight assignment of G and let S_1, S_2 be any two distinct sets of edges in G. Then w gives S_1, S_2 the same weight if and only if $< S_1 - S_2, w >= 0$.*

Let M be the matrix over $(F_q)^{n^2}$ constructed in Section 3 whose associated distribution D_M is $\frac{1}{2}$-biased. Then, by Definition 2.1 and the above proposition, we know that for any pair of distinct sets of edges S_1, S_2 in P, the probability that a random assignment w in M (i.e., a random row w in M) gives them the same weight is at most $(1+\frac{1}{2})/q$. Since in total there are r sets in P, the probability that there exists a pair of such sets such that a random assignment in M would give them the same weight is at most

$$\binom{r}{2} \times (1+\frac{1}{2})/q.$$

Now with our choice of q in the statement of the lemma, there must exist an assignment in M that gives any set of edges in P a distinct weight. This completes the proof of the lemma. □

Following Theorem 3.1 and Lemma 4.1 it is straightforward to see the next fact:

Corollary 4.1 *Let G be a directed graph on n vertices and let P be a family of r distinct paths in G. Suppose q is a prime power that is at least $2\binom{r}{2}$. Then there is a deterministic procedure that constructs in $O(\log n + \log r)$ space a matrix M over $(F_q)^{n^2}$ such that there exists a q-weight assignment of G in M that gives all r paths in P distinct weights. In particular, if r is poly(n) then such matrix can be constructed in $O(\log n)$ space.*

5 Main Results

Theorem 5.1 *LogFew $\subseteq$ NL*

Proof: Let $A \in$ LogFew. Thus, there is an NL machine N and a logspace-computable predicate B such that $x \in A$ iff $(x,i) \in B$, where $i = \#acc_N(x)$. In addition, since A is in LogFew, we know that $\#acc_N(x)$ is bounded by a polynomial in $|x|$.

First, we show that the language $\{(x,i) : \#acc_N(x) \geq i\}$ is in NL.

On input (x,i), build a graph G such that the number of s-t paths in G is equal to $\#acc_N(x)$. Now use Corollary 4.1 and see if there is a q-weight assignment in M that gives at least i distinct weights to paths in G. That is, for each q-weight assignment in M, guess a sequence of weights $w_1, w_2, \ldots, w_i$, and for each j attempt to find an s-t path in G having weight w_j under the given assignment.

It is easy to see that the above computation can be done by an NL machine. Since one of the q-weight assignments is guaranteed to give distinct weights to each of the s-t paths, the NL machine will accept if and only if $\#acc_N(x) \geq i$.

Now, since NL is closed under complementation, it follows that an NL machine can determine the value of $i = \#acc_N(x)$ exactly, and then check if $(x,i) \in B$. □

Theorem 5.2 *LogFew $\subseteq$ SPL.*

Proof: Let $f(x,i)$ be the #L function that counts the number of accepting computations of the NL machine accepting the language $\{(x,i) : \#acc_N(x) \geq i\}$ in the proof of the preceding theorem. We will now modify this function slightly.

Let $g(x,i,j)$ be the #L function that counts the number of accepting computations of the NL machine that, on input x, uses only the jth weight function of M to try to find at least i paths in the graph G. Note that *if* j is the "good" q-weight assignment for x, and if G really has exactly i paths, then $g(x,i,j) = 1$ (since there is exactly one sequence of guesses that will cause the NL machine to find the i paths and their weights). Also, if i is larger than the number of paths in G, then $g(x,i,j) = 0$.

Now consider the function $h(x,i,j)$ that is defined to be

$$g(x,i,j) \prod_{(j<j') \text{ or } (i'<i \text{ and } j'=j)} (1 - g(x,i',j')).$$

It follows from the standard closure properties of GapL that h is in GapL. (See, e.g. [AO96].)

If j is the lowest-numbered "good" q-weight function for x, then for the correct value of i, $h(x,i,j)$ is equal to 1. For all other values of i and j, $h(x,i,j)$ is equal to 0.

It now follows easily that any LogFew language is in $\mathrm{L}^{\mathrm{SPL}}$. It was observed in [AR98] that $\mathrm{L}^{\mathrm{SPL}}$ is equal to SPL. □

It is perhaps worth noting that Theorem 5.2 is in some sense the logspace-analog of the inclusion Few $\subseteq$ SPP, which was proved in [KSTT92]. Their proof relies on the fact that, for any #P function f and any polynomial-time function g that is bounded by a polynomial in n, the function $\binom{f(x)}{g(x)}$ is in #P. Note that, in contrast, this closure property is not known to hold for #L or GapL functions (although it is shown in [AR98] that if f is bounded by a polynomial in n, this closure property does hold in the *nonuniform* setting – and under a plausible hypothesis holds also in the uniform setting).

6 Conclusions and Open Problems

Can some of the other probabilistic inclusions relating to NL and UL be derandomized? Can one show that LogFewNL = UL, or that LogFew = UL? Can one show that UL = coUL? It seems that some of these questions should be in reach of current methods.

References

[AJ93] C. Àlvarez and B. Jenner. A very hard log-space counting class. *Theoretical Computer Science*, 107:3–30, 1993.

[AO96] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO - Theoretical Information and Application*, 30:1–21, 1996.

[AR98] E. Allender and K. Reinhardt. Isolation, Matching, and Counting. 13th IEEE Conference on Computational Complexity, 1998 (to appear).

[ABI86] N. Alon, L. Babai and A. Itai. A fast and simple randomized parallel algorithm for the Maximal Independent Set Problem. *J. Algorithms* 7:567-583, 1986.

[ABNNR92] N. Alon, J. Bruck, J. Naor, M. Naor and R. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs.

[AGHP92] N. Alon, O, Goldreich, J. Hastad and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms* 3(3):289-303, 1992.

[BDHM92] Gerhard Buntrock, Carsten Damm, Ulrich Hertrampf, and Christoph Meinel. Structure and importance of logspace-MOD class. *Math. Systems Theory*, 25:223–237, 1992.

[BJLR92] G. Buntrock, B. Jenner, K.-J. Lange, and P. Rossmanith. Unambiguity and fewness for logarithmic space. In *Proc. 8th International Conference on Fundamentals of Computation Theory (FCT '91)*, volume 529 of *Lecture Notes in Computer Science*, pages 168–179. Springer-Verlag, 1992.

[EGLNV92] G. Even, O. Goldreich, M. Luby, N. Nisan and B. Velickovic. Approximations of general independent distributions. In *Proc. of 24th ACM Symposium on Theory of Computing*, pp. 10-16, 1992.

[KK94] D. Karger and D. Koller. (De)randomized construction of small sample spaces in NC. In *Proc. of 35th IEEE Symposium on Foundations of Computer Science*, pp. 252-263, 1994.

[KM94] H. Karloff and Y. Mansour. On construction of k-wise independent random variables. In *Proc. of the 26th Annual ACM Symposium on Theory of Computing*, 1994.

[KSTT92] J. Köbler, U. Schöning, S. Toda, and J. Torán. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences* 44(2): 272–286, 1992.

[KW84] R. Karp and A. Wigderson. A fast parallel algorithm for the maximal independent set problem. In *Proc. of the 16th Annual ACM Symposium on Theory of Computing*, 1984.

[LN86] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their applications*, Cambridge University Press, 1986.

[Lub85] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036-1053, 1986.

[MV97] Meena Mahajan and V. Vinay. Determinant: Combinatorics, Algorithms, and Complexity. *Chicago Journal of Theoretical Computer Science*, MIT Press, 1997, number 5.

[MNN89] R. Motwani, J. Naor and M. Naor. The probabilistic method yields deterministic parallel algorithms. In *Proc. of 30th IEEE Symposium on Foundations of Computer Science*, pp. 8-13, 1989.

[NN90] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838-856, 1993.

[RA97] K. Reinhardt and E. Allender. Making nondeterminism unambiguous. In *38 th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 244–253, 1997.

[Vaz86] U. Vazirani. *Randomness, Adversaries and Computation.* Ph.D. Thesis, University of California, Berkeley, 1986.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'