Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 131 (2026)

From Nondeterministic to Modular Branching Programs, Uniformly and in Logspace

Benny Applebaum*

Dedicated to Avi Wigderson's 70th birthday.

Abstract

A branching program is a labeled directed graph that, on each input x, induces an s-t connectivity instance. Nondeterministic branching programs (NBPs) accept an input if there exists an accepting path, whereas parity branching programs ($\oplus$BPs) accept if the number of accepting paths is odd. Wigderson (Structure in Complexity Theory Conference, 1994) showed that NBPs can be simulated by $\oplus$BPs with the aid of nonuniform advice, establishing

$$\mathsf{NL}/\operatorname{poly} \subseteq \oplus\mathsf{L}/\operatorname{poly}.$$

In this short note, we observe that the recent breakthrough of Chatterjee, Ghosh, Gurjar, Raj, and Thierauf, placing bipartite perfect matching in deterministic NC, yields a uniform version of Wigderson's theorem. Specifically, we show that every nondeterministic branching program can be transformed, in deterministic logspace, into an equivalent polynomial-size parity branching program. More generally, for every fixed prime p, every nondeterministic branching program admits a logspace-uniform simulation by a polynomial-size mod-p branching program. In complexity-theoretic terms, this establishes $\mathsf{NL} \subseteq \oplus\mathsf{L}$ and, more generally, $\mathsf{NL} \subseteq \mathsf{Mod}_p\mathsf{L}$ for every fixed prime p.

1 Introduction

Branching programs and main results. A nondeterministic branching program (NBP) is a finite acyclic directed multigraph with a source s, a sink t, and edges labeled by constants or literals from

$$\{0, 1, x_1, \bar{x}_1, \ldots, x_m, \bar{x}_m\}.$$

For an assignment x, let B_x be the subgraph containing the edges whose labels evaluate to 1. The program accepts x if B_x contains an s–t path. A parity branching program accepts x if the number of its enabled source–sink paths is odd. For a prime p, a mod-p branching program accepts x if the number of its enabled source–sink paths is nonzero mod p.[1] The size of a branching program is the number of vertices and edges in the graph.

In this note, we observe that the recent result of Chatterjee, Ghosh, Gurjar, Raj, and Thierauf [7] yields a deterministic logspace reduction from bipartite perfect matching to full-row-rank testing

*Tel Aviv University, Israel, benny.applebaum@gmail.com. Supported by Israel Science Foundation (ISF) grant no. 2805/21 and by the European Union (ERC-2022-ADG) under grant agreement no.101097959 NFITSC.

[1] Some papers additionally require that whenever the number of accepting paths is nonzero modulo p, it is in fact equal to 1 mod p. The two definitions are equivalent up to a multiplicative factor of $p-1$ in size: one may sequentially compose $p-1$ copies of the branching program, thereby mapping a path count k to k^{p-1} mod p. By Fermat's little theorem, this maps 0 to 0 and every nonzero residue modulo p to 1.

ISSN 1433-8092

over any finite field of sufficiently large (polynomial) size. Combined with a standard reduction from directed reachability to bipartite matching and a standard representation of extension-field linear maps over the base field, which preserves full row rank, this gives a *uniform* simulation of nondeterministic branching programs by parity branching programs. The argument is independent of the characteristic and therefore extends to counting modulo any fixed prime. In particular, the argument places bipartite perfect matching in $\mathsf{Mod}_p\mathsf{L}$ for every fixed prime p; composing this containment with the standard logspace reduction from directed reachability to bipartite perfect matching yields the claimed simulation. Formally, we prove the following theorem.

Theorem 1. *For every fixed prime p, there is a deterministic logspace transducer that maps every NBP B to a mod-p branching program C_p of size $|B|^{O(1)}$ such that, for every assignment x,*

$$B(x) = 1 \quad \Longleftrightarrow \quad |\operatorname{Paths}((C_p)_x)| \not\equiv 0 \pmod{p}.$$

Consequently, $\mathsf{NL} \subseteq \mathsf{Mod}_p\mathsf{L}$ for every fixed prime p.

A cryptographic application. The class $\oplus\mathsf{L}$ plays a special role in cryptography. Applebaum, Ishai, and Kushilevitz [3] showed that every function in $\oplus\mathsf{L}$ admits a perfect randomized encoding by an NC^0 function; to the best of our knowledge, this is the largest known standard complexity class for which such a general result is available. Roughly speaking, given a function $f \in \oplus\mathsf{L}$, one can efficiently construct an NC^0 function $g(x; r)$, each of whose output bits depends on at most four input bits, such that, for every input x, the distribution of $g(x; r)$ for uniformly random r perfectly encodes the value $f(x)$. More precisely, an efficient decoder recovers $f(x)$ from $g(x; r)$, while an efficient simulator, given only $f(x)$, samples exactly from the distribution of $g(x; r)$. Here, *perfect* means that decoding is error-free, simulation is exact, and the encoding satisfies several additional regularity properties; see [3] for details.[2]

Previously known constant-degree randomized encodings for NL functions (that is, functions each of whose output bits is computable in NL) incurred a small correctness or privacy error. Indeed, Ishai and Kushilevitz [10] asked whether NL functions admit perfect constant-degree randomized encodings.[3] Combining Theorem 1 with [3, Theorem 4.18 and Lemmas 4.15 and 4.17] yields the following corollary.

Corollary 2. *Every Boolean function computable in NL admits a uniform perfect randomized encoding in NC^0 of degree at most 3 and locality at most 4.*

Remark 3 (Multi-output functions)**.** *The same argument applies more generally to polynomial-time-uniform families of multi-output nondeterministic branching programs. Let $f : \{0,1\}^* \to \{0,1\}^*$ be such that, for every n, the output length $|f(x)| = m(n)$ depends only on $n = |x|$, and let $f_n : \{0,1\}^n \to \{0,1\}^{m(n)}$ denote the restriction of f to n-bit inputs. We say that f is computable by a polynomial-time-uniform family of nondeterministic branching programs if there is a deterministic polynomial-time algorithm that, on input 1^n, outputs descriptions of branching programs $B_{n,1}, \ldots, B_{n,m(n)}$, where $B_{n,i}$ computes the ith output bit of f_n. Under this convention, Theorem 1 and [3, Theorem 4.18] imply that every such function admits a uniform perfect randomized encoding in NC^0. Following the complexity-theoretic notation of [3], the above can be stated as*

$$\textit{poly-time uniform } \mathsf{NL}/\operatorname{poly} \subseteq \mathsf{PREN}.$$

[2] These additional properties are, in fact, crucial for some applications; see, e.g., [3, Section 6] and [13] for a more recent example.

[3] We thank Yuval Ishai for pointing us to this open question.

Related work. The relationship between nondeterministic and modular logspace has a long history. Buntrock, Damm, Hertrampf, and Meinel introduced and systematically studied the classes $\mathrm{Mod}_k\,\mathsf{L}$ and established the connection with linear algebra over finite fields [6]. Wigderson subsequently proved the nonuniform containment

$$\mathsf{NL}/\operatorname{poly} \subseteq \oplus\mathsf{L}/\operatorname{poly},$$

using isolation to reduce reachability to odd reachability [14]. Wigderson's proof is field-independent and, when interpreted over $\mathbb{F}_p$, yields the analogous nonuniform containment for $\mathrm{Mod}_p\,\mathsf{L}$. This observation is made explicitly by Gál and Wigderson, who also extended the method to LogCFL and semi-unbounded fan-in circuits [9]. Beimel and Gál [5] gave another simulation of NBPs by counting branching programs with improved size parameters. These simulations rely on fixing suitable randomness as *nonuniform* advice and therefore do not yield uniform transformations. Allender, Reinhardt, and Zhou further investigated the uniform and nonuniform consequences of isolation for matching and logspace counting classes [2]. The linear-algebraic landscape was developed further by Allender, Beals, and Ogihara, who characterized the complexity of matrix rank and linear feasibility over the integers under logspace reductions [1].

2 Proof of the main theorem

Proof of Theorem 1. The proof composes four transformations; Section 2.5 verifies that their composition is computable by a deterministic logspace transducer.

2.1 Directed reachability to bipartite perfect matching

First normalize B by deleting 0-labeled edges, edges entering s, and edges leaving t. Parallel edges may be removed by subdividing each edge $e = (u, v)$ into

$$u \longrightarrow w_e \longrightarrow v,$$

where the first edge retains the label of e and the second is labeled 1. Let $G = (V, E)$ be the resulting directed graph, with distinguished vertices $s, t \in V$; thus (G, s, t) is an instance of directed reachability. For an assignment x, let $G_x = (V, E_x)$ be the subgraph consisting of the enabled edges.

Construct a balanced bipartite graph H_x with sides

$$L = \{u_L : u \in V \setminus \{t\}\}, \qquad R = \{v_R : v \in V \setminus \{s\}\}.$$

For every internal vertex $v \in V \setminus \{s, t\}$ add the constant edge $v_L v_R$, and for every directed edge $u \to v$ of G add the bipartite edge $u_L v_R$ with the same label. This folklore reduction is recorded explicitly, for example, by Assadi and Raz [4, Section 6.2]; it is also closely related to the classical reductions used by Karp, Upfal, and Wigderson [11]. For every assignment x,

$$s \rightsquigarrow t \text{ in } G_x \quad \Longleftrightarrow \quad H_x \text{ has a perfect matching.} \tag{1}$$

Indeed, an s–t path is completed to a perfect matching using the identity edges $v_L v_R$ for vertices off the path. Conversely, a perfect matching defines, for every $u \neq t$, a unique successor v through the matched edge $u_L v_R$, and every $v \neq s$ has a unique predecessor. Starting at s, this successor relation cannot enter a directed cycle: the first vertex of such a cycle would then have both a predecessor on the cycle and a predecessor from outside it. Hence the resulting directed walk is a simple path, and since t is the only vertex without a left copy, it must terminate at t.

Let $A_H(x)$ denote the symbolic bi-adjacency matrix whose evaluation at $x \in \{0,1\}^n$ is the bi-adjacency matrix of H_x. Since parallel edges have been removed, every entry of $A_H(x)$ belongs to

$$\{0, 1, x_i, \bar{x}_i : i \in [n]\}.$$

2.2 Perfect matching to rank over an extension of $\mathbb{F}_p$

Fix a prime p. Our next goal is to reduce the problem of deciding whether the bipartite graph represented by $A_H(x)$ contains a perfect matching to the full-rank problem over $\mathbb{F}_p$.

Let $n = |L| = |R|$. Following Section 3 of [7], extend $A_H(x)$ to an $n \times n^2$ matrix

$$\widehat{A}(x) = (A_H(x) \mid A_0),$$

where A_0 is an $n \times n(n-1)$ binary matrix whose ith row is $\mathbf{0}^{i-1}\mathbf{1}\mathbf{0}^{n-i}$ where $\mathbf{0} = 0^{n-1}$ and $\mathbf{1} = 1^{n-1}$. That is, each row consists of n blocks of size $n-1$ each, where the ith block of the ith row is the all-one block, and all other blocks are zero blocks. Set

$$r = n^3 + 1, \qquad D = n(r-n), \qquad d = \left\lceil \log_p(n^2 r + 1) \right\rceil.$$

Let $K = \mathbb{F}_{p^d}$ and choose a primitive element $\gamma \in K$. Then $\mathrm{ord}(\gamma) = p^d - 1 \geq n^2 r$. Put

$$\alpha_j = \gamma^{(j-1)r}, \qquad 1 \leq j \leq n^2.$$

The elements $\alpha_j \gamma^\ell$, for $1 \leq j \leq n^2$ and $0 \leq \ell < r$, are pairwise distinct. Define the folded Vandermonde matrix

$$V(\alpha_j)_{q,\ell} = (\alpha_j \gamma^\ell)^q, \qquad 0 \leq q < D, \quad 0 \leq \ell < r,$$

and form the block matrix

$$M(x)_{i,j} = \widehat{A}_{i,j}(x) V(\alpha_j), \qquad M(x) \in K^{nD \times n^2 r}. \tag{2}$$

The matching-matrix theorem of Chatterjee, Ghosh, Gurjar, Raj, and Thierauf [7, Theorem 3.1], whose algebraic ingredient is the folded-Wronskian/subspace-design method of Guruswami and Kopparty [8], states that

$$H_x \text{ has a perfect matching} \quad \Longleftrightarrow \quad \mathrm{rank}_K\, M(x) = nD. \tag{3}$$

The dependence on x is local: every block of $M(x)$ is either zero or a fixed folded-Vandermonde matrix multiplied by a single literal or constant.

2.3 Simulating extension-field linear maps over $\mathbb{F}_p$

Fix an $\mathbb{F}_p$-basis of K. For $a \in K$, let

$$\rho(a) \in \mathbb{F}_p^{d \times d}$$

be the matrix of the $\mathbb{F}_p$-linear map $z \mapsto az$. Applying ρ entrywise in block form to $M(x)$ gives

$$C(x) = \rho(M(x)) \in \mathbb{F}_p^{nDd \times n^2 rd}.$$

The matrices $M(x)$ and $C(x)$ represent the same linear transformation, viewed respectively as a K-linear map and as an $\mathbb{F}_p$-linear map. Therefore,

$$\mathrm{rank}_{\mathbb{F}_p}\, C(x) = d\, \mathrm{rank}_K\, M(x). \tag{4}$$

Moreover, if an entry of $M(x)$ is $a\ell(x)$, where $a \in K$ is constant and ℓ is a literal or a constant, then the corresponding $d \times d$ block over $\mathbb{F}_p$ is $\ell(x)\rho(a)$. Hence every bit in a fixed binary encoding of $C(x)$ is either a constant or a literal. In particular,

$$\mathrm{rank}_K\, M(x) = nD \quad \Longleftrightarrow \quad \mathrm{rank}_{\mathbb{F}_p}\, C(x) = nDd. \tag{5}$$

2.4 $\mathbb{F}_p$ full rank to a mod-p branching program

For every fixed prime p, testing whether a matrix over $\mathbb{F}_p$ has full row rank belongs to $\mathsf{Mod}_p\mathsf{L}$ [6, 12]; in particular, full-row-rank testing over $\mathbb{F}_2$ belongs to $\oplus\mathsf{L}$. We will now convert this $\mathsf{Mod}_p\mathsf{L}$ computation to a mod-p branching program (via a parsimonious reduction).

Fix a polynomial-time-bounded nondeterministic logspace machine R satisfying

$$\#\,\mathrm{acc}_R(C) \not\equiv 0 \pmod{p} \quad\Longleftrightarrow\quad C \text{ has full row rank over } \mathbb{F}_p. \tag{6}$$

Take the time-layered configuration graph of R. We do not write $C(x)$ explicitly. Whenever R reads a bit of the encoding of $C(x)$, that bit is recomputed from its indices. If it is a constant, the corresponding transition is retained or deleted. If it is a literal $\ell(x)$, a transition requiring the bit to be 1 is labeled by ℓ, while a transition requiring it to be 0 is labeled by $\bar{\ell}$. After padding all computations to a common polynomial length and joining all accepting configurations to a single sink, the resulting acyclic graph is a mod-p branching program P. Its enabled source–sink paths are in bijection with the accepting computations of R on $C(x)$.

Combining (1), (3), (5), and (6) establishes the equivalence asserted in Theorem 1.

2.5 Logspace uniformity

The purely graph-theoretic parts of the construction are computable by deterministic logspace transducers. The normalization of B and the reduction of Section 2.1 can be generated by enumerating the possible output vertices and edges using logarithmic counters and determining, by a logspace scan of B, whether each edge is present and how it is labeled. Likewise, the time-layered configuration graph in Section 2.4 can be generated in deterministic logspace, provided that whenever R queries a bit of the encoding of $C(x)$, that bit can be recomputed from its position using logarithmic space. It therefore remains to verify this property for the algebraic constructions of Sections 2.2 and 2.3.

Since $d = O(\log n)$, the field K has polynomial size. A degree-d irreducible polynomial over $\mathbb{F}_p$ can be found deterministically in $O(d \log p)$ workspace by enumerating all monic degree-d polynomials and rejecting any candidate divisible by a nonconstant monic polynomial of degree at most $d/2$. Candidate polynomials, possible divisors, and remainders require $O(d \log p) = O(\log n)$ bits. Once a representation of K is fixed, a primitive element can be found by enumerating K^* and computing successive powers; the field has polynomial size, so this takes polynomial time and logarithmic workspace.

Given their indices, the entries of the folded Vandermonde matrices and the bits of the multiplication matrices $\rho(a)$ are therefore computable in $O(\log n)$ space. Configurations and transitions of the fixed machine R can likewise be enumerated in logarithmic space. Closure of logspace transductions under composition shows that $B \mapsto P$ is a deterministic logspace transduction, and all intermediate dimensions are polynomial in $|B|$. Finally, the standard logspace-uniform correspondence between polynomial-size NBPs and nondeterministic logspace machines yields $\mathsf{NL} \subseteq \mathsf{Mod}_p\mathsf{L}$ and, for $p = 2$, $\mathsf{NL} \subseteq \oplus\mathsf{L}$. □

Remark 4 (DLOGTIME-uniformity)**.** *The construction also preserves* DLOGTIME*-uniformity. Let $\{B_m\}_{m\geq 1}$ be a* DLOGTIME*-uniform polynomial-size family of NBPs. Rather than having the transducer evaluate the field-dependent entries of $C(x)$, we incorporate their deterministic logspace computation into the machine simulating R. Whenever this computation needs to inspect an edge or a label of B_m, it simulates the direct-connection procedure for the family. For the fixed family $\{B_m\}_{m\geq 1}$, the resulting machine $\widehat{R}$ is fixed, polynomial-time bounded, and uses $O(\log m)$ space.*

Under the standard encoding, each time-layered configuration of $\widehat{R}$ has length $O(\log m)$, and whether one configuration follows another in a single step can be decided in $O(\log m)$ time. Transitions that query the ith input bit are labeled by x_i or $\overline{x}_i$, as appropriate. Hence the resulting symbolic configuration graphs have a DLOGTIME*-decidable direct-connection relation. Consequently, the transformation maps every* DLOGTIME*-uniform family of nondeterministic branching programs to a* DLOGTIME*-uniform family of equivalent mod-p branching programs.*

Acknowledgments. We thank Yuval Ishai and Oded Goldreich for helpful comments on this manuscript.

References

[1] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8(2):99–126, 1999.

[2] E. Allender, K. Reinhardt, and S. Zhou. Isolation, matching, and counting: Uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59(2):164–181, 1999.

[3] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

[4] S. Assadi and R. Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 342–353, 2020.

[5] A. Beimel and A. Gál. On arithmetic branching programs. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity (CCC)*, pages 68–80, 1998.

[6] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-MOD classes. *Mathematical Systems Theory*, 25(3):223–237, 1992.

[7] A. Chatterjee, S. Ghosh, R. Gurjar, R. Raj, and T. Thierauf. Bipartite matching is in NC. *ECCC Report TR26-100*, revision 2, 2026.

[8] V. Guruswami and S. Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016.

[9] A. Gál and A. Wigderson. Boolean complexity classes vs. their arithmetic analogs. *Random Structures & Algorithms*, 9(1–2):99–111, 1996.

[10] Y. Ishai and E. Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Automata, Languages and Programming—ICALP 2002*, volume 2380 of *Lecture Notes in Computer Science*, pages 244–256. Springer, 2002.

[11] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.

[12] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th Annual Structure in Complexity Theory Conference*, pages 102–111, 1993.

[13] H. Ren, R. Santhanam, and Z. Wang. On the range avoidance problem for circuits. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 640–650, 2022.

[14] A. Wigderson. $\mathsf{NL}/\operatorname{poly} \subseteq \oplus\mathsf{L}/\operatorname{poly}$. In *Proceedings of the 9th Annual Structure in Complexity Theory Conference*, pages 59–62, 1994.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il