Electronic Colloquium on Computational Complexity, Report No. 340 (2026)

$\mathsf{LogCFL} = \mathsf{UAuxPDA}\big(O(\log n), \mathrm{poly}(n)\big)$

A. Pavan N. V. Vinodchandran

Abstract

We observe that the isolation technique introduced by Hoza in his proof of $\mathsf{NL} = \mathsf{UL}$ extends to give $\mathsf{LogCFL} = \mathsf{UAuxPDA}(O(\log n), \mathrm{poly}(n))$.

1 Introduction

We observe that Hoza's technique [1] extends to the complexity class LogCFL and show that LogCFL can be made unambiguous. We use notation from Hoza's paper [1].

2 Proof

Theorem. *Every language in* LogCFL *is accepted by an unambiguous auxiliary pushdown automaton in polynomial time with* $O(\log n)$ *work space. That is,* $\mathsf{LogCFL} = \mathsf{UAuxPDA}(O(\log n), \mathrm{poly}(n))$

Proof. By Venkateswaran's characterization of LogCFL as logspace-uniform SAC^1 [4], it suffices to evaluate a layered semi-unbounded Boolean circuit of polynomial size N and depth $D = O(\log N)$. A semi-unbounded circuit has AND gates with fan-in 2 and OR gates with unbounded fan-in. Leaves are in level zero and wires join consecutive levels. Fix the circuit and its input. A certificate tree for an accepting input chooses one true child at each OR and chooses all children (at most two) at each AND gate. Given a weight function w that assigns weights to gates, the weight of a certificate tree T is the sum $W(T; w) = \sum_{x \in V(T)} w(g_x)$, where g_x is the circuit gate at node x, and $V(T)$ is the vertex set of the tree T. We call w *min-unique* on this input if every true gate has exactly one minimum-weight certificate tree rooted at it.

By the work of Reinhardt and Allender [2, Section 3], it suffices for unambiguous circuit evaluation to construct a logspace-computable, polynomially bounded weight function that is min-unique for the given circuit and input. However, we design a family of weight functions as opposed to a single weight function. Reinhardt and Allender also give an unambiguous auxiliary pushdown automaton that checks whether a given weight function is min-unique or not. Their work uses weight on circuit wires; however, van Melkebeek and Prakriya [3] design a similar procedure to check min-uniqueness when the weights are assigned to nodes. We prove that for a given circuit and input, at least three-fourths of the weight functions are min-unique. By using the unambiguous procedure of Reinhardt and Allender and van Melkebeek and Prakriya, we can cycle through each weight function in the family till we find a min-unique weight function.

min-unique. Reinhardt and Allender also give an unambiguous auxiliary pushdown automaton that checks whether a given weight function is min-unique or not. Their work uses weight on circuit wires; however, van Melkebeek and Prakriya [3] design a similar procedure to check min-uniqueness when the weights are assigned to nodes.

ISSN 1433-8092

Weight function

Let N denote the number of gates in the circuit and let D be its depth. As in Hoza [1], let $E : [N] \to \{0,1\}^\ell$ be a logspace-computable code with distance $\delta \geq \lceil \log_2(2N^2) \rceil$, where $\ell = O(\log N)$.

For a seed $z \in \{0,1\}^\ell$ and a gate v at level h, define

$$w_z(v) = 1 + \sum_{j=1}^{\ell} 6^{\,h+\ell-j} E(v)_j z_j,$$

Since the height is $O(\log N)$, the weights are polynomial-bounded and each gate weight is computable in logspace.

Hoza uses the 2-adic valuation of the layer index to select a highest-priority layer in each path interval. In our scenario, since the circuits are log depth, the weight function is slightly simpler. In a certificate tree the root is the unique node at its highest level, and at most 2^r nodes occur r levels below it. We can therefore scale weights at level h by 6^h. Since $D = O(\log N)$, even 6^D is polynomial in N. Note that each $z \in \{0,1\}^\ell$ define a weight function. Since $\ell = O(\log N)$, we have a polynomial-size family of weight functions.

Isolation

We use Hoza's bit-flipping argument [1, Theorem 2.4]. For a true gate v, let $F_v(z) = \min_T W(T; w_z)$, where the minimum ranges over all certificate trees rooted at v. Let $(\partial_j F_v)(z) = F_v(z_{j\leftarrow 1}) - F_v(z_{j\leftarrow 0})$, keeping the other bits fixed.

Claim. *For every true gate v at level h, every seed z, and every $j \in \{1, \ldots, \ell\}$, let $B_j = 6^{h+\ell-j}$. Then*

$$B_j E(v)_j \;\leq\; (\partial_j F_v)(z) \;\leq\; B_j\big(E(v)_j + \tfrac{1}{2}\big). \tag{1}$$

Proof of the claim. Fix a certificate tree T rooted at v, and write $W_b(T) = W(T; w_{z_{j\leftarrow b}})$. Turning bit j on adds $B_j E(v)_j$ at the root. At distance r there are at most 2^r nodes, each gaining at most $B_j/6^r$. The descendants therefore gain between zero and $B_j \sum_{r\geq 1}(2/6)^r = B_j/2$. Hence for every T,

$$W_0(T) + B_j E(v)_j \leq W_1(T) \leq W_0(T) + B_j E(v)_j + B_j/2.$$

Taking the minimum over the same set of trees in all three terms and subtracting $\min_T W_0(T)$ gives (1), since $\min_T W_b(T) = F_v(z_{j\leftarrow b})$. □

Claim. *For distinct true gates u, v on the same level and a uniformly random seed $z \in \{0,1\}^\ell$,*

$$\Pr_z\big[F_u(z) = F_v(z)\big] \leq 2^{-\delta}. \tag{2}$$

Proof. Fix all coordinates where $E(u)_j = E(v)_j$. At least δ free coordinates remain where the codewords disagree. We show at most *one* assignment of these free coordinates can produce a tie.

Suppose z is a tie, i.e., $F_u(z) = F_v(z)$. Let z' be a string that differ from z in at aleast one of the free coordinates. Consider a process that converts z to z' by flipping the free coordinates them in increasing order. Let j be the first coordinate flipped. WLOG $E(u)_j = 1$ and $E(v)_j = 0$. Then by the interval bound:

$$\begin{aligned} \partial_j F_u \;&\geq\; B_j \cdot 1 = B_j, \\ \partial_j F_v \;&\leq\; B_j \cdot \tfrac{1}{2} = \tfrac{B_j}{2}. \end{aligned}$$

So flipping z_j moves $F_u - F_v$ by at least $B_j - B_j/2 = B_j/2$ away from zero. For all later flips $k > j$, the total correction is at most

$$\tfrac{3}{2} \sum_{k>j} B_k \le \tfrac{3}{2} \cdot B_j \sum_{s=1}^{\ell} 6^{-s} \le \tfrac{3}{2} \cdot \frac{B_j}{5} = \frac{3B_j}{10} < \frac{B_j}{2}.$$

Hence the gap cannot return to zero, so z' is not a tie. At most one assignment of the free coordinates yields a tie; since there are $2^m \ge 2^\delta$ assignments, $\Pr[\text{tie}] \le 2^{-\delta}$. □ □

Claim. *For every fixed circuit and input, at least* $3/4$ *of the seeds* $z \in \{0,1\}^\ell$ *give a min-unique weight function* w_z.

Proof of the claim. By (2), each pair of distinct true gates on the same level has equal minimum weights with probability at most $2^{-\delta}$. There are at most $\binom{N}{2}$ such unordered pairs across all levels. Since $2^{-\delta} \le 1/(2N^2)$, the union bound gives

$$\Pr_z[\text{some such pair ties}] \le \binom{N}{2} 2^{-\delta} \le \tfrac{1}{4}.$$

Fix a seed with no such tie. We prove by induction on the level that every true gate has a unique minimum-weight certificate tree. A true leaf has only the one-node certificate. At a true OR gate, all true children lie on the preceding level and have distinct minimum weights. A minimum tree must choose the child u with the smallest $F_u(z)$ and its unique minimum tree. At a true AND gate, a minimum tree must use a minimum tree for every child. Each is unique by induction, so their combination in the prescribed order is unique. Thus every seed with no tie is min-unique, proving the claim. □

In particular, the family contains the required min-unique weighting. □

AI disclosure

GPT Astra 6.2 found the extension of Hoza's isolation technique to LogCFL. The authors take full responsibility for the contents of this note.

References

[1] William M. Hoza. NL = UL. *Electronic Colloquium on Computational Complexity*, TR26-323, 2026.

[2] Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM Journal on Computing*, 29(4), 1118–1131, 2000.

[3] Dieter van Melkebeek and Gautam Prakriya. Derandomizing isolation in space-bounded settings. *SIAM Journal on Computing*, 48(3), 979–1021, 2019.

[4] H. Venkateswaran. Properties that characterize LOGCFL. *Journal of Computer and System Sciences*, 43(2), 380–404, 1991.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il