Circuit Lower Bounds From Above

Marco L Carmosino

October 2, 2023

1 Introduction

How can we "get past" the relativization barrier to prove new lower bounds? Our current formulation of the barrier is informal, so we give heuristics. Recall the definition of a non-relativizing statement.

Definition 1 (Non-Relativizing Statement)**.** Let $\varphi(\mathcal{C}, \mathcal{D})$ be a statement about complexity classes $\mathcal{C}$ and $\mathcal{D}$. A statement φ is *Non-Relativizing* if $\exists A\ \varphi^A \wedge \exists B\ \neg\varphi^B$. We write $\varphi^{\mathcal{O}}$ as shorthand for equipping every complexity class or machine mentioned by φ with oracle $\mathcal{O}$.

Two approaches towards non-relativizing separations are:

Circuit Lower Bounds Relativizing proofs treat computation as a "black box" — using only efficient simulation and enumeration of TMs. We can try switching to a combinatorial model of computation that forces proofs to use "white box" reasoning.

Non-Relativizing Lemmas We can examine the proof of *any* non-relativizing statement about computation — not necessarily a separation — and mine it for helpful "ingredients" that avoid relativization, if used in a sufficiently essential way.

We present a simplified proof of the the first blatantly non-relativizing separation: $\mathsf{MA\text{-}EXP} \not\subset \mathsf{P/poly}$ [BFT98]. The proof combines both approaches suggested above: borrowing non-relativizing techniques from the proof of $\mathsf{IP} = \mathsf{PSPACE}$ and proving lower bounds for circuits instead of machines.

2 Definitions & Tools

2.1 Circuits

Relativizing arguments use that every TM has a constant length description. We can instead consider computation by *Boolean devices* — models composed of logic gates where a different device is used at each input length. For example, we have distinct 8-bit and 16-bit adder circuits. This makes circuit complexity a *non-uniform* model of computation.

Definition 2 (Boolean Circuits on n-Bit Inputs 6.1, 6.2 of [AB09])**.** An *n-input Boolean Circuit* is a directed acyclic graph (DAG) of in-degree at most 2 where each node is labeled with exactly one of:

- n input symbols $x_1, \ldots, x_n$
- three logical operation symbols $\{\wedge, \vee, \neg\}$
- the output symbol o

We call the nodes *gates* and the in-degree bound the *fan-in*. Because the out-degree is unbounded, we have unrestricted *fan-out*. We call the "size" of a circuit C the number of nodes and denote it $|C|$. We define the value of C on $x \in \{0,1\}^n$ as the result of substituting the ith bit of x for each gate labelled with x_i and then inductively evaluating each logic gate until the output gate has a value; denote this value by $C(x)$.

Definition 3 (Bounded-Size Circuit Families)**.** Let $T : \mathbb{N} \to \mathbb{N}$ be a gate-counting function. A $T(n)$-size circuit family is a sequence of circuits $\{C_n\}_{n\in\mathbb{N}}$ where $\forall n$

- C_n has n inputs
- $|C_n| \leq T(n)$

Fix a language $\mathcal{L} \subseteq \{0,1\}^*$. We say that $\mathcal{L}$ has $T(n)$-size circuits and denote this by

$$\mathcal{L} \in \mathsf{SIZE}[T(n)] \iff \exists \text{ a } T(n)\text{-size circuit family } \{C_n\}_{n\in\mathbb{N}} \text{ such that } \forall x \in \{0,1\}^n \;\; C_n(x) = \mathcal{L}(x)$$

Just as with machine-based models, we consider any polynomial gate-bound to be feasible.

Definition 4 ("Small" Or "Efficient" Circuits — 6.5 of [AB09])**.** $\mathsf{P/poly} = \bigcup_{c\in\mathbb{N}} \mathsf{SIZE}[n^c]$

Given that modern CPUs are built of programmable logic arrays computing a useful collection of 64-bit Boolean functions, it is no surprise that we can simulate machines by circuits.

Theorem 1 (Simulation of TMs By Circuits)**.**
$\mathsf{P} \subset \mathsf{P/poly}$

This allows a reformulation of P vs NP which may seem more tractable: is $\mathsf{NP} \subset \mathsf{P/poly}$? That circuit lower bound is *stronger* than $\mathsf{P} \neq \mathsf{NP}$. The hope was that, by thinking of the deterministic class as circuits, we could avoid relativizing proofs by using more "details" of computation than just efficient simulation lemmas. Towards this end, Kannan proved the following.

Theorem 2 (Slicewise Circuit Lower Bounds for PH — Theorem 2 of [Kan82])**.**
$\forall k \in \mathbb{N}\; \exists \mathcal{L}_k \in \mathsf{PH} \;\; \mathcal{L}_k \notin \mathsf{SIZE}[n^k]$

Lemma 1 (Circuit Lower "From Above" — Lemma 4 of [Kan82])**.**
$\mathsf{NEXP}^{\mathsf{NP}} \not\subset \mathsf{P/poly}$

Kannan's arguments mixed combinatorics with diagonalization against circuits. Even so, the above results are relativizing [Wil85]. Nearly a decade later, appropriate tools were invented to bypass the barrier.

2.2 Non-Relativizing Ingredient: Algebrization

The following non-relativizing *equivalence* was a breakthrough in complexity theory.

Theorem 3 (8.19 of [AB09], originally [Sha92; Lun+92])**.**
$\mathsf{PSPACE} = \mathsf{IP}$

The hard direction was showing $\mathsf{PSPACE} \subseteq \mathsf{IP}$. Relativization explained the difficulty involved, because $\exists \mathcal{O}\; \mathsf{PSPACE}^{\mathcal{O}} \not\subseteq \mathsf{IP}^{\mathcal{O}}$ [FS88]. Thus, to show $\mathsf{PSPACE} \subseteq \mathsf{IP}$, a non-relativizing technique was required. Roughly speaking, the idea was to transform a circuit into a polynomial, and use the fact that claims about the evaluation of polynomials can be efficiently verified using random bits. Our notes do not cover this proof, because it is an important part of the standard complexity theory class; see Section 8.3 of [AB09].

So, as of 1992, the question remained: we had a non-relativizing *inclusion* but no *separation*. How do we "hijack" this to prove separations? The conditional collapse below follows by observing that the prover in the protocol witnessing $\mathsf{PSPACE} \subseteq \mathsf{IP}$ can be implemented in PSPACE.

Theorem 4 (Karp-Lipton Style Conditional Collapse, 8.22 [AB09])**.**

$$\mathsf{PSPACE} \subset \mathsf{P/poly} \implies \mathsf{PSPACE} = \mathsf{MA}$$

3 MA-EXP Does Not Have Efficient Circuits

We combine the two ingredients introduced above to get a separation. Unconditionally, we have Kannan's circuit lower bound for $\mathsf{NEXP}^{\mathsf{NP}}$. We use the Karp-Lipton style theorem to move the hard language from Kannan's bound "down" into MA-EXP.

Theorem 5 ([BFT98])**.**

$$\mathsf{MA\text{-}EXP} \not\subset \mathsf{P/poly}$$

Proof. 1. Assume towards contradiction that $\mathsf{MA\text{-}EXP} \subset \mathsf{P/poly}$

2. Unconditionally, $\mathsf{PSPACE} \subseteq \mathsf{EXP} \subseteq \mathsf{MA\text{-}EXP}$
3. Combining the above with Theorem 4, $\mathsf{PSPACE} \subset \mathsf{P/poly} \implies \mathsf{PSPACE} = \mathsf{MA}$
4. Unconditionally, $\mathsf{NP}^{\mathsf{NP}} \subseteq \mathsf{PSPACE}$
5. Therefore, under our assumption, $\mathsf{NP}^{\mathsf{NP}} \subseteq \mathsf{MA}$
6. Padding up and recalling Lemma 1, $\mathsf{NEXP}^{\mathsf{NP}} \subseteq \mathsf{MA\text{-}EXP} \subset \mathsf{P/poly} \implies \bot$

□

Our next note will certify that $\mathsf{MA\text{-}EXP} \not\subset \mathsf{P/poly}$ is indeed non-relativizing.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN: 978-0-521-42426-4. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

[BFT98] Harry Buhrman, Lance Fortnow, and Thomas Thierauf. "Nonrelativizing Separations". In: *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, Buffalo, New York, USA, June 15-18, 1998*. IEEE Computer Society, 1998, pp. 8–12. DOI: 10.1109/CCC.1998.694585. URL: https://doi.org/10.1109/CCC.1998.694585.

[FS88] Lance Fortnow and Michael Sipser. "Are There Interactive Protocols for CO-NP Languages?" In: *Inf. Process. Lett.* 28.5 (1988), pp. 249–251. DOI: 10.1016/0020-0190(88)90199-8. URL: https://doi.org/10.1016/0020-0190(88)90199-8.

[Kan82] Ravi Kannan. "Circuit-Size Lower Bounds and Non-Reducibility to Sparse Sets". In: *Inf. Control.* 55.1-3 (1982), pp. 40–56. DOI: 10.1016/S0019-9958(82)90382-5. URL: https://doi.org/10.1016/S0019-9958(82)90382-5.

[Lun+92] Carsten Lund et al. "Algebraic Methods for Interactive Proof Systems". In: *J. ACM* 39.4 (1992), pp. 859–868. DOI: 10.1145/146585.146605. URL: https://doi.org/10.1145/146585.146605.

[Sha92] Adi Shamir. "IP = PSPACE". In: *J. ACM* 39.4 (1992), pp. 869–877. DOI: 10.1145/146585.146609. URL: https://doi.org/10.1145/146585.146609.

[Wil85] Christopher B. Wilson. "Relativized Circuit Complexity". In: *J. Comput. Syst. Sci.* 31.2 (1985), pp. 169–181. DOI: 10.1016/0022-0000(85)90040-6. URL: https://doi.org/10.1016/0022-0000(85)90040-6.