Circuit Lower Bounds from Unconditional Derandomization of Restricted Merlin Arthur Protocols[*]

Lijie Chen
MIT
lijieche@mit.edu

March 21, 2022

Abstract

In this note we discuss the recent developments on proving circuit lower bounds from nontrivial derandomization (a.k.a., the algorithmic method). As the main focus, we will show how to derandomize Merlin-Arthur protocols with ACC^0 verifiers in nondeterministic quasi-polynomial time infinitely often and consequently deduce ACC^0 lower bounds.

We will first present a recent perspective on the algorithmic method that decomposes the whole proof into three conceptual ingredients and show how these three ingredients lead to new circuit lower bounds. Next, we will explain one of the ingredients in more detail: an approach to unconditionally derandomize Merlin-Arthur protocols whose verifiers have small circuits from a certain circuit class.

1 Introduction

Background. One important direction in complexity theory is to prove that certain explicit functions (usually meaning functions in NP) cannot be computed by small circuits. Indeed, if one can prove that a function in NP cannot be computed by poly-size general circuit, then one separates NP from P.

Unfortunately, our knowledge for even constant depth circuits are very limited. Strong lower bounds are known against AC^0 [Ajt83, FSS84, Yao85, Hås89] and $\mathsf{AC}^0[p]$ for a prime p [Raz87, Smo87].[1] However, progress has been slow since the 80s, and it had been difficult to prove lower bounds even against $\mathsf{AC}^0[6]$. A decade ago, Williams [Wil11] proved $\mathsf{NEXP} \not\subseteq \mathsf{AC}^0[6]$. In 2018, Murray and Williams [MW18] proved that $\mathsf{NTIME}[2^{\mathrm{polylog}(n)}]$ is not in $\mathsf{AC}^0[6]$.[2]

In this note, we will give a different presentation of the $\mathsf{AC}^0[6]$ lower bound from [MW18] as a consequence of the unconditional derandomization of $\mathsf{MA}_{\mathsf{AC}^0[6]}$ proved in [CLW20].

[*] This note is based on a talk given by the author at the Institute for Advanced Study on February 22, 2022. Feedbacks are definitely welcome and please send them to wjmzbmr@gmail.com.

[1] AC^0 denotes the class of polynomial-size constant-depth circuits with unbounded fan-in $\mathsf{AND}/\mathsf{OR}/\mathsf{NOT}$ gates. $\mathsf{AC}^0[m]$ denotes the class of polynomial-size constant depth circuits with unbounded fan-in $\mathsf{AND}/\mathsf{OR}/\mathsf{NOT}/\mathsf{MOD}_m$ gates. Here, $\mathsf{MOD}_m \colon \{0,1\}^n \to \{0,1\}$ outputs 1 if the number of 1's in the input is not dividable by m, and 0 otherwise.

[2] Both [Wil11, MW18] indeed proved lower bounds against $\mathsf{AC}^0[m]$ for every constant $m \in \mathbb{N}$ (*i.e.*, they proved lower bounds against ACC^0). To simplify the discussions, we will be focusing on $\mathsf{AC}^0[6]$ in this note. But we remark that all our results apply to $\mathsf{AC}^0[m]$ for every constant $m \in \mathbb{N}$.

Overview. The focus of this note is a very interesting bootstrapping theorem, which shows that barely-faster-than-brute-force (non-trivial) derandomization of Merlin Arthur (MA) protocols imply very fast derandomization of Merlin-Arthur protocols (see Section 2.1 for a definition of MA). In particular, we will show how to derandomize Merlin-Arthur protocols with ACC^0 verifiers (see Section 2.1 for a precise definition) in non-deterministic quasi-polynomial time, and how this derandomization implies circuit lower bounds for ACC^0; see Figure 1 for an overview of this note.

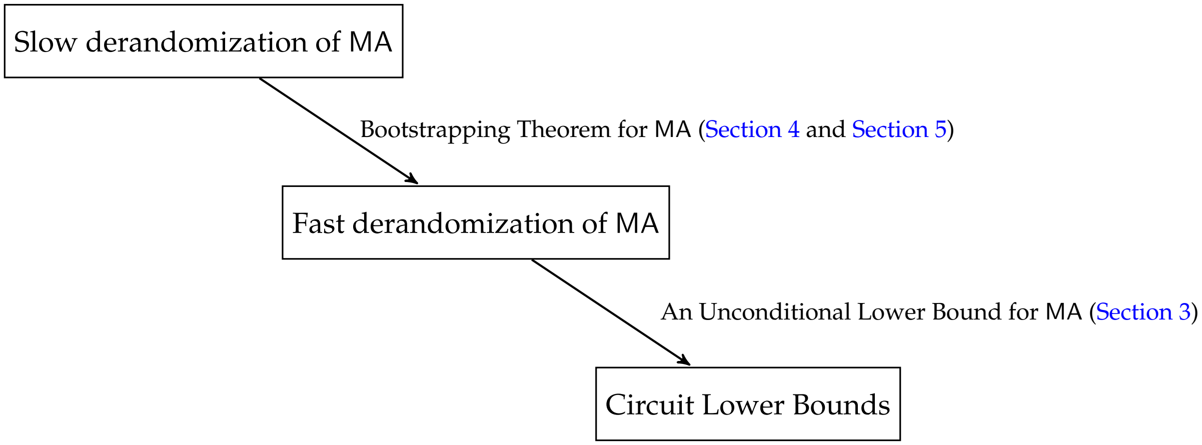
Slow derandomization of MA
Bootstrapping Theorem for MA (Section 4 and Section 5)
Fast derandomization of MA
An Unconditional Lower Bound for MA (Section 3)
Circuit Lower Bounds

Figure 1: Structure of the proof: High-level

2 Derandomization of Merlin-Arthur protocols by NPRGs

To derandomize MA, we will rely on a generalization of pseudorandom generator (PRG) called nondeterministic pseudorandom generator (NPRG). In this section we first recall some other relevant notion[3], and then motivate and introduce the concept of NPRG.

PRG. We say a function $G\colon \{0,1\}^s \to \{0,1\}^n$ is an ε-PRG for a class of n-input function $\mathcal{F}$, if for all $f \in \mathcal{F}$

$$\left| \Pr_{x \in \{0,1\}^n}[f(x) = 1] - \Pr_{r \in \{0,1\}^s}[f(G(r)) = 1] \right| \leq \varepsilon.$$

We will always assume that a PRG is computable in time $2^{O(s)}$, and the s is called the seed length of G. Note that given an $s(n)$-seed PRG for all $\mathrm{poly}(n)$-size circuits, one can estimate acceptance probability of a $\mathrm{poly}(n)$-size circuit in $2^{O(s)} \cdot \mathrm{poly}(n)$ time, which implies $\mathsf{BPP} \subseteq \mathsf{TIME}[2^{O(s(n))}]$. In particular, an $O(\log n)$-seed PRG for $\mathrm{poly}(n)$-size circuits implies $\mathsf{BPP} = \mathsf{P}$.

2.1 NTIME and MATIME

Let us also recall the definition of $\mathsf{NTIME}[T(n)]$. For a time bound function $T(n)$, a language $L \in \mathsf{NTIME}[T(n)]$ if there is an algorithm $V(x,y)$ such that $|x| = n$ and $|y| = T(n)$ and

$$x \in L \Leftrightarrow \exists y \in \{0,1\}^{T(|x|)} V(x,y) = 1.$$

We also define $\mathsf{NP} = \mathsf{NTIME}[\mathrm{poly}(n)]$, $\mathsf{NQP} = \mathsf{NTIME}[2^{\mathrm{polylog}(n)}]$.[4]

[3] We will assume basic familiarity with complexity theory (see, *e.g.*, [AB09, Gol08]).

[4] More precisely, $\mathsf{NP} = \bigcup_{k \in \mathbb{N}} \mathsf{NTIME}[n^k]$. NQP is defined similarly.

MA is the randomized version of NP. We now recall the definition of $\mathsf{MATIME}[T(n)]$. For a time bound function $T(n)$, a language $L \in \mathsf{MATIME}[T(n)]$ if there is an algorithm $V(x,y,r)$ such that $|x| = n$ and $|y| = |r| = T(n)$ and

$$x \in L \Rightarrow \exists y \in \{0,1\}^{T(|x|)} \Pr_{r \in \{0,1\}^{T(|x|)}}[V(x,y,r) = 1] \geq 2/3,$$

and

$$x \in L \Rightarrow \forall y \in \{0,1\}^{T(|x|)} \Pr_{r \in \{0,1\}^{T(|x|)}}[V(x,y,r) = 1] \leq 1/3.$$

We also define $\mathsf{MA} = \mathsf{MATIME}[\mathrm{poly}(n)]$.

$\mathsf{MA}_{\mathscr{C}}$. In this note we will pay attention to the complexity of the verifiers in MA. Let $\mathscr{C}$ be a circuit class ($\mathsf{AC}^0, \mathsf{AC}^0[6]$, etc.). We say $L \in \mathsf{MATIME}_{\mathscr{C}}[T(n)]$ if the corresponding verifier V above also satisfies the additional condition below.

- For every $x \in \{0,1\}^n$ and $y \in \{0,1\}^{T(n)}$, $V(x,y,\cdot)$ (the restriction of V to the randomness part) has a $T(n)$-size $\mathscr{C}$ circuit.

The above is weaker than insisting that V itself has a $T(n)$-size $\mathscr{C}$ circuit, so it applies to more languages and makes our results stronger. We also say $L \in \mathsf{MATIME}_{\mathscr{C}}[T(n); R(n)]$, if the corresponding verifier V only takes $R(n)$ bits of randomness.

2.2 Nondeterministic PRG

Finally we are ready to introduce NPRG, which is weaker than PRG, but still suffices to derandomize MA.

An NPRG $G = (G_{\mathsf{W}}, G_{\mathsf{P}})$ is a pair of *two* functions $G_{\mathsf{W}} \colon \{0,1\}^w \to \{0,1\}$ and $G_{\mathsf{P}} \colon \{0,1\}^w \times \{0,1\}^s \to \{0,1\}^n$; Here we call w the witness length. We always assume both G_{P} and G_{W} are computable in $2^{O(s)}$ time and $w \leq 2^{O(s)}$.

We say that G is an ε-NPRG for a class of functions $\mathcal{F}$, if the following two conditions hold.

1. For some $u \in \{0,1\}^w$, $G_{\mathsf{W}}(u) = 1$.
2. For every $u \in \{0,1\}^w$ such that $G_{\mathsf{W}}(u) = 1$, $G_{\mathsf{P}}(u;\cdot)$ is an ε-PRG for $\mathcal{F}$.

We remark that an ε-PRG can be seen as an ε-NPRG with witness length 1 and G_{W} always outputs 1.

Now, we note that an $s(n)$-seed-length $w(n)$-witness-length $1/10$-NPRG G for $T(n)$-size $\mathscr{C}$ circuits can be used to derandomize $\mathsf{MA}_{\mathscr{C}}[T(n)]$. Formally, let $L \in \mathsf{MA}_{\mathscr{C}}[T(n)]$ and $V(x,y,r)$ be the corresponding randomized verifier. We construct a new deterministic verifier V' as follows:

- V' takes both $y \in \{0,1\}^{T(n)}$ and $u \in \{0,1\}^{w(n)}$ as witness. (*i.e.*, V' takes $T(n) + w(n)$ bits as witness.)
- Accept if $G_{\mathsf{W}}(u) = 1$ and $\Pr_{r \in \{0,1\}^s}[V(x,y,G_{\mathsf{P}}(u,r)) = 1] \geq 1/2$.

It is straightforward to verify that the above verifier V' implies that $L \in \mathsf{NTIME}[2^{O(s(n))}]$. We remark that the concept of NPRG is already implicit in [IKW02]. Our definition is from the journal version of [Che19].[5]

[5] See http://www.mit.edu/~lijieche/Che19-journal-version.pdf for the draft.

Background on PRG. As it turns out, building PRG is even harder than proving circuit lower bounds. After decades of work, we now have $\mathrm{polylog}(n)$-seed-length PRGs for AC^0 [NW94, Hås89, Kel21, Lyu22] that matches the best known lower bounds against AC^0.

However, no $(1-\Omega(1))\cdot n$-seed-length PRG is known for $\mathsf{AC}^0[2]$. The state-of-the-art for $\mathsf{AC}^0[2]$ is an $n - n/\mathrm{polylog}(n)$-seed-length PRG by [FSUV13].

Recently, there are some new developments in constructing NPRGs. Specifically, [CR20] constructed i.o.- $n^{o(1)}$-seed-length NPRG for $\mathsf{AC}^0[6]$, which implies that $\mathsf{MA}_{\mathsf{AC}^0[6]} \subseteq$ i.o.-$\mathsf{NTIME}[2^{n^{o(1)}}]$.[6] Later, [CLW20] gave an i.o.- $\mathrm{polylog}(n)$-seed-length NPRG for $\mathsf{AC}^0[6]$, which implies that $\mathsf{MA}_{\mathsf{AC}^0[6]} \subseteq$ i.o.-NQP. In this note we will cover the intuitions behind [CLW20]'s NPRG construction.

3 Detailed Overview and From Derandomization to Circuit Lower Bounds

Now we are ready to give a more precise version of Figure 1.

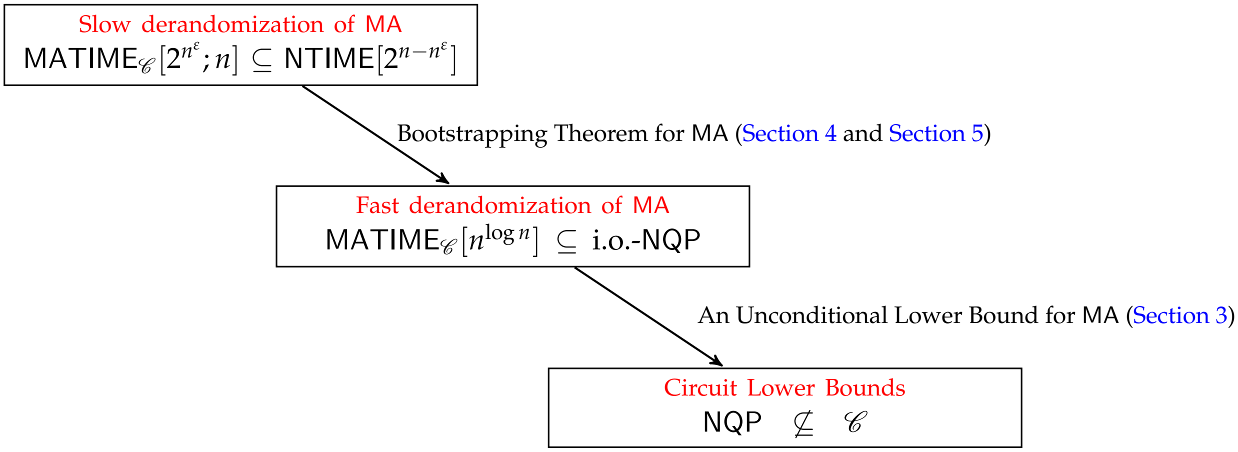
Slow derandomization of MA
$\mathsf{MATIME}_{\mathscr{C}}[2^{n^{\varepsilon}}; n] \subseteq \mathsf{NTIME}[2^{n-n^{\varepsilon}}]$
Bootstrapping Theorem for MA (Section 4 and Section 5)
Fast derandomization of MA
$\mathsf{MATIME}_{\mathscr{C}}[n^{\log n}] \subseteq$ i.o.-NQP
An Unconditional Lower Bound for MA (Section 3)
Circuit Lower Bounds
$\mathsf{NQP} \not\subseteq \mathscr{C}$

Figure 2: Structure of the proof: Detailed Version

We elaborate below.

1. (**Starting point: Non-trivial algorithm**) When $\mathscr{C} = \mathsf{AC}^0[6]$, $\mathsf{MATIME}_{\mathscr{C}}[2^{n^{\varepsilon}}; n] \subseteq \mathsf{NTIME}[2^{n-n^{\varepsilon}}]$ follows from [Wil11].[7]

2. (**First $\Rightarrow$: Bootstrapping theorem for** $\mathsf{MA}_{\mathscr{C}}$)

   The version for $\mathscr{C}$ being general circuits was proved by Williams [Wil13], which unfortunately does not work when $\mathscr{C} = \mathsf{AC}^0[6]$. The general bootstrapping theorem for all typical circuit classes $\mathscr{C}$ was proved by Chen, Lyu, and Williams [CLW20].

   Combining with the non-trivial derandomization of $\mathsf{MA}_{\mathsf{AC}^0[6]}$, we now have

$$\mathsf{MATIME}[n^{\log n}]_{\mathsf{AC}^0[6]} \subseteq \text{i.o.-}\mathsf{NQP}. \tag{1}$$

   We will first cover the bootstrapping theorem for $\mathscr{C}$ being general circuits in Section 4, and then show how to make it work for all typically circuit classes in Section 5.

[6] You can ignore the i.o.- notation for now. We will discuss this in details later.

[7] In fact, Williams gave a $2^{n-n^{\varepsilon}}$-time *deterministic* algorithm for *counting* the number of satisfying assignments to an n-input $2^{n^{\varepsilon}}$-size $\mathsf{AC}^0[6]$ circuit, which is stronger than what we need here.

3. (**Second ⇒: Circuit Lower Bounds**)

   We prove unconditionally that there is language $L^{\mathsf{hard}} \in \mathsf{MATIME}[n^{\log n}]_{\mathsf{AC}^0[6]}$ that is hard for $\mathsf{AC}^0[6]$ [Che22].[8]

   We can then derandomize L^{hard} into NQP to prove that NQP is not in $\mathsf{AC}^0[6]$.

3.1 An Important Technical Remark

The infinitely often issue. For readers that are interested in knowing more details, there is a crucial technical issue in the argument above. Earlier we wanted the readers to ignore the "i.o.-" in (1), but it now becomes crucial. Roughly speaking, it says that the derandomization of MA into NQP only works for infinitely many input lengths, not all input lengths. When we say L^{hard} is hard for $\mathsf{AC}^0[6]$ circuits, it usually only means on infinitely many input lengths L^{hard} is hard. What if on all those hard input lengths the derandomization does not work? Then we have no hardness at all!

Solution by [MW18]. This issue was addressed by Murray and Williams [MW18]. Roughly speaking, they strengthened both the derandomization and the lower bounds so that the "hardness input lengths" and the "good-for-derandomization input lengths" must intersect each other.

Formally, (1) means that for any $L^{\mathsf{hard}} \in \mathsf{MATIME}_{\mathsf{AC}^0[6]}[n^{\log n}]$, there is an NQP language $\widetilde{L}$ such that for infinitely many $n \in \mathbb{N}$, L^{hard} and $\widetilde{L}$ agree with on n-bit inputs. And our issue is that L^{hard} may happen to be not hard on those n.

Murray and Williams resolved the issue above as follows:

1. We indeed can get $\widetilde{L}$ such that there are infinitely many $n \in \mathbb{N}$, for all $m \in [n, n^{\log n}]$, $L^{\mathsf{hard}}_m = \widetilde{L}_m$. In other words, $\widetilde{L}$ agrees with L^{hard} on infinitely many segments of input lengths, such that each segment has a quasi-polynomial stretch.

2. We can also prove a "robust $\mathsf{AC}^0[6]$ lower bound": for all large enough $n \in \mathbb{N}$, there exists $m \in [n, n^{\log n}]$, L^{hard}_m is hard for polynomial-size $\mathsf{AC}^0[6]$.[9]

The two statements above together imply that for infinitely many $m \in \mathbb{N}$, L^{hard}_m is hard and $L^{\mathsf{hard}}_m = \widetilde{L}_m$, hence $\widetilde{L}$ has no polynomial-size $\mathsf{AC}^0[6]$ circuits, and consequently $\mathsf{NQP} \not\subseteq \mathsf{AC}^0[6]$.

In [Che22], we indeed prove that $L^{\mathsf{hard}} \in \mathsf{MATIME}[n^{\log n}]_{\mathsf{AC}^0[6]}$ is "robustly hard" against $\mathsf{AC}^0[6]$.

Some personal thoughts. I indeed believe the infinitely often condition can be removed in (1) and we can prove that $\mathsf{MATIME}[n^{\log n}]_{\mathsf{AC}^0[6]} \subseteq \mathsf{NQP}$. If that can be done, then we don't have to prove the robust circuit lower bounds. I leave it as a fascinating open problem.[10]

4 Bootstrapping theorem: Warm up

In this section, we prove the following warm-up theorem, which is implicit in [Wil13, Section 3.2].

[8] Strictly speaking this language also needs one bit of advice. We will not discuss this technicality in this note.

[9] This is the technical centerpiece of [MW18]. See also [Che19] for a simpler proof.

[10] Indeed, [CLW20] proved that $\mathsf{MATIME}[n^{\log n}]_{\mathsf{AC}^0[6]} \subseteq \mathsf{TIME}[n^{\mathrm{polylog}(n)}]^{\mathsf{NP}}$. But it seems hard to further improve $\mathsf{TIME}[n^{\mathrm{polylog}(n)}]^{\mathsf{NP}}$ to NQP.

Theorem 1 (Bootstrapping theorem: warm up)**.** *For every constant $\varepsilon \in (0,1)$, $\mathsf{MATIME}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n-n^\varepsilon}]$ implies that $\mathsf{MA} \subseteq$ i.o.-NQP.*

In below, we will always fix ε to be a small enough constant.

4.1 Overview of the proof

We first give a high-level overview of our proof of Theorem 1.

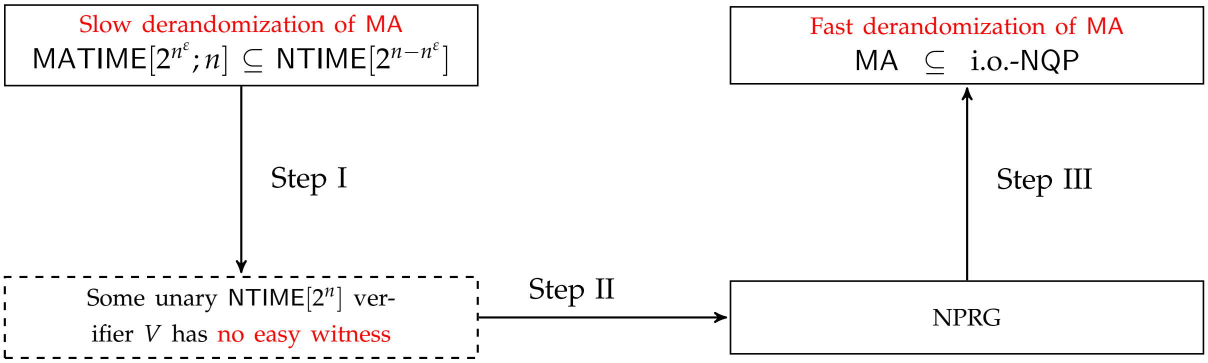
Slow derandomization of MA
$\mathsf{MATIME}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n-n^\varepsilon}]$
Fast derandomization of MA
MA ⊆ i.o.-NQP
Step I
Step III
Step II
Some unary $\mathsf{NTIME}[2^n]$ verifier V has no easy witness
NPRG

Figure 3: Structure of the proof for Theorem 1

The no easy witness condition. We first need to explain the dashed rectangle in Figure 3. By a unary $\mathsf{NTIME}[2^n]$ verifier V, we mean that V is a verifier for a unary language $L \subseteq \{1^n\}_{n\in\mathbb{N}}$ such that $L \in \mathsf{NTIME}[2^n]$. In other words,

$$1^n \in L \Leftrightarrow \exists y \in \{0,1\}^{2^n} V(1^n, y) = 1.$$

In above, $V(x, y)$ takes an input x and a witness y such that $|x| = n$ and $|y| = 2^n$, runs in roughly 2^n time.

We say "V has easy witnesses", if

- (V **has easy witnesses**) for all sufficiently large $n \in \mathbb{N}$, $1^n \in L$ implies that $V(1^n, y) = 1$ for some $y \in \{0,1\}^{2^n}$ (interpreted as an n-bit function) that is the truth-table of a 2^{n^ε}-size circuit $C\colon \{0,1\}^n \to \{0,1\}$.

By "V has no easy witness", we mean the above does not hold. Equivalently, it means

- (V **has no easy witness**) for infinitely many $n \in \mathbb{N}$, $1^n \in L$ and $V(1^n, y) = 0$ for all $y \in \{0,1\}^{2^n}$ (interpreted as an n-bit function) that is the truth-table of a 2^{n^ε}-size circuit. We call those n *hard* for V.

We also call above an **witness lower bound** for V against 2^{n^ε}-size circuits.

4.2 Step II and Step III

Now, we note that Step III, from (i.o.-) NPRG to the derandomization of MA is already explained in Section 2.2. Below we first discuss the more straightforward Step II, before moving to the most interesting part Step I.

Step II: Witness lower bound $\Rightarrow$ NPRG Construction. Given a V that has no easy witnesses. We can construct an NPRG $G = \{G^n = (G^n_{\mathsf{W}}, G^n_{\mathsf{P}})\}_{n \in \mathbb{N}}$ as follows:

- $G^n_{\mathsf{W}} \colon \{0,1\}^{2^n} \to \{0,1\}$: given $u \in \{0,1\}^{2^n}$, outputs $V(1^n, u)$.

For all hard n for V, we know (1) G^n_{W} accepts some truth-tables and (2) $G^n_{\mathsf{W}}(u) = 1$ means that $f_u \colon \{0,1\}^n \to \{0,1\}$ (u treated as an n-bit function) cannot be computed by 2^{n^ε}-size circuits.

PRG requires average-case hardness and worst-case-to-average-case reduction. Given $f \colon \{0,1\}^n \to \{0,1\}$ that is $1/2 + 2^{-n^\varepsilon}$ hard against 2^{n^ε}-size $\mathscr{C}$ circuits,[11] using the Nisan-Wigderson [NW94] PRG construction, we have a PRG $\mathsf{NW}^f \colon \{0,1\}^{\mathrm{poly}(n)} \to \{0,1\}^S$ that fools S-size $\mathscr{C}$ circuits, for some $S = 2^{\Omega(n^\varepsilon)}$.

The problem is that $G^n_{\mathsf{W}}(u) = 1$ only means f_u is hard in the worst-case. Lucikly, for general circuits we have worst-case-to-average-case reductions [STV01], so given f that cannot be computed by 2^{n^ε}-size circuits, we can get $\mathsf{Amp}(f) \colon \{0,1\}^{O(n)} \to \{0,1\}$ that is $1/2 + 2^{-n^\varepsilon}$ hard against 2^{n^ε}-size circuits.

We define $G^n_{\mathsf{P}}(u, r) := \mathsf{NW}^{\mathsf{Amp}(f_u)}(r)$. For all the hard n for V, we know that $G^u_{\mathsf{W}}(u) = 1$ implies that f_u is worst-case hard and hence $\mathsf{Amp}(f_u)$ is average-case hard, and therefore $\mathsf{NW}^{\mathsf{Amp}(f_u)}$ is a PRG.

Finally, look at the parameters, the seed length is $\mathrm{poly}(n) = \mathrm{polylog}(S)$, where $S = 2^{n^\varepsilon}$ is the size of the circuits being fooled. So we have a $\mathrm{polylog}(S)$ i.o.- NPRG fooling S-size circuits.

4.3 Step I: $\mathsf{MATIME}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n - n^\varepsilon}] \Rightarrow$ No easy witness for some verifier V

Finally, we are ready to establish Step I. Here we wish to find a verifier V for a unary language $L \in \mathsf{NTIME}[2^n]$ such that V has **no easy witness**.

Intuitively, the language L should be as hard as possible, this leads us to apply the following unary NTIME hierarchy theorem.

NTIME **hierarchy theorem [Žák83].** There is a unary language $L \subseteq \{1^n \colon n \in \mathbb{N}\}$ such that $L \in \mathsf{NTIME}[2^n]$ and $L \notin \mathsf{NTIME}[2^n/n]$.

In some sense, L is the **hardest unary language** in $\mathsf{NTIME}[2^n]$! And that is why it makes sense to start from it.

PCP **theorem.** We also need to pick a verifier V for L. We will apply the famous PCP theorem [ALM+98, AS98]. Roughly speaking, PCP theorem implies that one can spend only polynomial-time to check *an exponentially long proof*, by only accessing very few bits in the proof.

Specifically, we will use a very efficient version of PCP theorem (see, *e.g.*, [BV14]), which implies that there is a super efficient verifier V_{PCP} for L, such that the following holds:

1. $V_{\mathsf{PCP}}(1^n)^{\mathcal{O}}(r)$ expects an oracle $\mathcal{O} \colon \{0,1\}^n \to \{0,1\}$ and takes n bits of randomness $r \in \{0,1\}^n$, and runs in $\mathrm{poly}(n)$ time.[12]

2. $1^n \in L \Rightarrow \exists \mathcal{O} \colon \{0,1\}^n \to \{0,1\}$

$$\Pr_{r \in \{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^{\mathcal{O}}(r) = 1] = 1.$$

[11] Here we mean that for all 2^{n^ε}-size $\mathscr{C}$ circuits $C \colon \{0,1\}^n \to \{0,1\}$, we have $\Pr_{x \in \{0,1\}^n}[f(x) = C(x)] < 1/2 + 2^{-n^\varepsilon}$.

[12] Strictly speaking $\mathcal{O}$ should have $n + O(\log n)$ bits as input instead of only n bits, and also needs $n + O(\log n)$ bits of randomness. We pretend that n bits are enough to simplify the presentation.

3. $1^n \notin L \Rightarrow \forall \mathcal{O}\colon \{0,1\}^n \to \{0,1\}$
$$\Pr_{r \in \{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^{\mathcal{O}}(r) = 1] \leq 1/3.$$

We now claim that the verifier $V(1^n, u)$ that outputs 1 if and only if $\Pr_{r\in\{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^{f_u}(r) = 1] = 1$, has **no easy witness**. Note that $V(1^n, u)$ runs in roughly 2^n deterministic time, by enumerating all $r \in \{0,1\}^n$.

Let us suppose for the contradiction that $V(1^n, u)$ has easy witness. In other words, it means

- $(\mathsf{EW}$ condition): For all sufficiently large $1^n \in L$, $\exists$ 2^{n^ε}-size circuit $C\colon \{0,1\}^n \to \{0,1\}$ such that
$$\Pr_r[V_{\mathsf{PCP}}(1^n)^C(r) = 1] = 1.$$

In the following, we will show that

$$(\mathsf{EW}) \overset{\circledast}{\Longrightarrow} L \in \mathsf{MATIME}[2^{n^\varepsilon}; n] \overset{\circledcirc}{\Longrightarrow} L \in \mathsf{NTIME}[2^{n-n^\varepsilon}], \qquad (2)$$

which is a contradiction to that $L \notin \mathsf{NTIME}[2^n/n]$ by NTIME hierarchy. Note that $\circledcirc$ follows from the assumption $\mathsf{MATIME}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n-n^\varepsilon}]$, so below we only need to establish $\circledast$.

An algorithm A_{PCP} putting $L \in \mathsf{MATIME}[2^{n^\varepsilon}; n]$ assuming EW. To show $\circledast$, we define the following Merlin-Arthur algorithm A_{PCP} that attempts to solve L.

1. Given an input 1^n.
2. Guess a 2^{n^ε}-size circuit $C\colon \{0,1\}^n \to \{0,1\}$.
3. Draw $r \in \{0,1\}^n$ and output $V_{\mathsf{PCP}}(1^n)^C(r)$.

It is not hard to verify that (EW) indeed implies that A_{PCP} solves L, and that A_{PCP} is an $\mathsf{MATIME}[2^{n^\varepsilon}; n]$ algorithm. This completes the proof of (2), leading to a contradiction. Hence (EW) is false and we proved that V has no easy witness, thereby completing the proof of Step I and also Theorem 1.

Why the argument above does not generalize to all typical circuit classes. Next we briefly discuss why the above does not work directly for all typical circuit classes $\mathscr{C}$. Suppose we only assume $\mathsf{MATIME}_{\mathscr{C}}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n-n^\varepsilon}]$, and in A_{PCP} we guess 2^{n^ε}-size $\mathscr{C}$ circuits instead of general circuits, the above proof still works in the sense that it now gives us a verifier V that has no 2^{n^ε}-size $\mathscr{C}$ witness.

However, we *may not have worst-case-to-average-case reduction for $\mathscr{C}$ circuits* (for example $\mathsf{AC}^0[6]$ or $\mathsf{AC}^0[2]$), so we don't know how to get a strong average-case hard function against $\mathscr{C}$, which is required by PRG construction. But still, this gives us a verifier who only accepts truth-tables with high worst-case $\mathsf{AC}^0[6]$ circuit complexity. This gives us an $\mathsf{AC}^0[6]$ witness lower bound and it is proved in [Wil16].

5 Bootstrapping Theorem: for all typical circuit classes

As discussed above, we will need strong average-case lower bounds against $\mathscr{C}$ circuits to construct PRGs fooling $\mathscr{C}$ circuits. Hence, we will need some tools for proving average-case circuit lower bounds.

Linear sums. Let $\mathscr{C}$ be a circuit class. We define $\mathsf{Sum} \circ \mathscr{C}$ as a class of real-output functions H that can be written as

$$H(x) = \sum_{i=1}^{m} \alpha_i \cdot C_i(x),$$

where $\alpha_i \in [-1,1]$ and C_i is a $\mathscr{C}$ circuit. We then define the size of H as

$$\mathsf{SIZE}(H) := \sum_{i \in [m]} \mathsf{SIZE}(C_i).$$

We also require that $H(x) \in [0,1]$ for all $x \in \{0,1\}^n$. This is indeed very important and we will see why shortly.

An XOR Lemma based on linear sums [Lev87, CLW20, CL21]. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we define $f^{\oplus k}\colon (\{0,1\}^n)^k \to \{0,1\}$ as

$$f^{\oplus k}(x_1, x_2, \dots, x_k) := \bigoplus_{i \in [k]} f(x_i).$$

We need the following lemma.

Lemma 2. *Let $f\colon \{0,1\}^n \to \{0,1\}$. If*

- *for all $\mathsf{Sum} \circ \mathscr{C}$ circuits $H\colon \{0,1\}^n \to [0,1]$ with $\mathsf{SIZE}(H) \le \frac{10ns}{\varepsilon^2}$, it holds that*

$$\mathop{\mathbb{E}}_{x \in \{0,1\}^n} |H(x) - f(x)| > 0.01.$$

  (We will abbreviate the condition above as f is 0.01-hard against $\mathsf{Sum} \circ \mathscr{C}$ circuits of size $\frac{10ns}{\varepsilon^2}$.)

Then

- *$f^{\oplus k}$ is $(1/2 + \varepsilon)$-hard against s-size $\mathscr{C}$ circuits, for some $k = \Theta(\log \varepsilon^{-1})$.*

Remark. Let us remark a bit on the XOR Lemma above. This is somewhat implicit in the Levin's proof [Lev87] of the XOR Lemma, and some researchers have been aware that Levin's proof gives a linear sum reconstruction. This fact was pointed out to me first by Shuichi Hirahara and then by Ronen Shaltiel. To the best of our knowledge, Lemma 2 first explicitly formalized in [CLW20].

Later, I and Lyu [CL21] gave a completely different proof of Lemma 2, which is based on a careful application of the linear programing duality. The new proof gives a very interesting generalization of the hardcore-based proof of XOR Lemma by Impagliazzo [Imp95], and can be derandomized in the same way of the derandomized XOR Lemma by [IW97].

Now we are ready to formally state our bootstrapping theorem for all typical circuit classes.

Theorem 3. *For every constant $\varepsilon \in (0,1)$, $\mathsf{MATIME}_{\mathsf{Sum} \circ \mathscr{C}}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n - n^\varepsilon}]$ implies that $\mathsf{MA}_{\mathscr{C}} \subseteq$ i.o.-NQP.*[13]

We have to clarify what do we mean by a $\mathsf{Sum} \circ \mathscr{C}$ verifier in the notation $\mathsf{MATIME}_{\mathsf{Sum} \circ \mathscr{C}}[2^{n^\varepsilon}; n]$: when the verifier outputs a real value $\alpha \in [0,1]$, we interpret it as accepting with probability α. Note that we crucially used the promise that a $\mathsf{Sum} \circ \mathscr{C}$ always outputs values from $[0,1]$.

[13] An acute reader will notice that we "lied" in Section 3 and Figure 2 (there the condition was stated as $\mathsf{MATIME}_{\mathscr{C}}[2^{n^\varepsilon}; n] \subseteq \mathsf{NTIME}[2^{n - n^\varepsilon}]$). We made that choice since we had not define $\mathsf{Sum} \circ \mathscr{C}$ then.

5.1 Overview of the argument.

Similarly to Figure 3, the proof of Theorem 3 has the following structure.

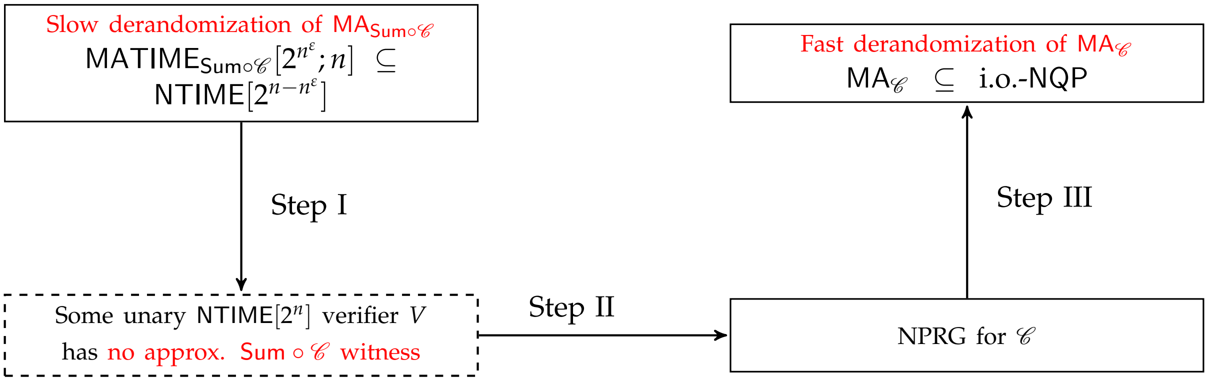
Slow derandomization of $\mathsf{MA}_{\mathsf{Sum}\circ\mathscr{C}}$
$\mathsf{MATIME}_{\mathsf{Sum}\circ\mathscr{C}}[2^{n^{\varepsilon}}; n] \subseteq \mathsf{NTIME}[2^{n-n^{\varepsilon}}]$
Step I
Some unary $\mathsf{NTIME}[2^n]$ verifier V has no approx. $\mathsf{Sum}\circ\mathscr{C}$ witness
Step II
NPRG for $\mathscr{C}$
Step III
Fast derandomization of $\mathsf{MA}_{\mathscr{C}}$
$\mathsf{MA}_{\mathscr{C}} \subseteq$ i.o.-NQP

Figure 4: Structure of the proof for Theorem 3

We remark that Williams' algorithm from [Wil14] also implies that

$$\mathsf{MATIME}_{\mathsf{Sum}\circ\mathsf{AC}^0[6]}[2^{n^{\varepsilon}}; n] \subseteq \mathsf{NTIME}[2^{n-n^{\varepsilon}}],$$

so that combining with Theorem 3, we have $\mathsf{MA}_{\mathsf{AC}^0[6]} \subseteq$ i.o.-NQP.

No approx. $\mathsf{Sum}\circ\mathscr{C}$ witness. Now we first explain what the dashed rectangle means in Figure 4. By "unary $\mathsf{NTIME}[2^n]$ verifier V has approx. $\mathsf{Sum}\circ\mathscr{C}$ witnesses", we mean that

- for all sufficiently large $n \in \mathbb{N}$, $1^n \in L \Rightarrow V(1^n, u) = 1$ for some $u \in \{0,1\}^{2^n}$ such that $\mathbb{E}_{x\in\{0,1\}^n} |f_u(u) - H(x)| \leq 0.01$ for some $2^{n^{\varepsilon}}$-size $\mathsf{Sum}\circ\mathscr{C}$ circuit $H\colon \{0,1\}^n \to [0,1]$. (*i.e.*, V has witnesses that is 0.01-close to $2^{n^{\varepsilon}}$-size $\mathsf{Sum}\circ\mathscr{C}$ circuits.)

By "unary $\mathsf{NTIME}[2^n]$ verifier V has no approx. $\mathsf{Sum}\circ\mathscr{C}$ witness", we mean the above condition does not hold. Equivalently

1. for infinitely many $n \in \mathbb{N}$, $1^n \in L$ and $V(1^n, u) = 0$ for all $u \in \{0,1\}^{2^n}$ (interpreted as an n-bit function) that is 0.01-close to a $2^{n^{\varepsilon}}$-size $\mathsf{Sum}\circ\mathscr{C}$ circuit $H\colon \{0,1\}^n \to [0,1]$. Again, we call those n hard for V.

Step II and Step III. Again, Step III follows from the definition of NPRG (see Section 2.2). For Step II, now we can define $G^n_{\mathsf{W}}(u) = V(1^n, u)$ and then set $G^n_{\mathsf{P}}(f, r) := \mathsf{NW}^{f^{\oplus\Theta(n^{\varepsilon})}}(r)$.

Now, for a hard n for V, we know that $G^n_{\mathsf{W}}(u) = 1$ for some u, and for every u satisfying $G^n_{\mathsf{W}}(u) = 1$, its corresponding n-bit function f_u is 0.01-hard against $\mathsf{Sum}\circ\mathscr{C}$ circuits of size $2^{n^{\varepsilon}}$. By Lemma 2, we then know $f_u^{\oplus\Theta(n^{\varepsilon})}$ is $1/2 + 2^{-n^{\varepsilon}}$-hard against $2^{n^{\varepsilon}}$-size $\mathscr{C}$ circuits, and hence $G^n_{\mathsf{P}}(u, \cdot)$ is a PRG fooling $\mathscr{C}$ circuits.

5.2 Step I: deriving no approx. witness

We use a similar setup as in Section 4. Recall that L is a unary language that is in $\mathsf{NTIME}[2^n] \setminus \mathsf{NTIME}[2^n/n]$, and V_{PCP} is its PCP verifier.

We similarly claim that the verifier $V(1^n, u)$ that outputs 1 if and only if $\Pr_{r \in \{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^{f_u}(r) = 1] = 1$, has no approx. $\mathsf{Sum} \circ \mathscr{C}$ witness. Again, note that $V(1^n, u)$ runs in roughly 2^n time.

Assuming V has approx. $\mathsf{Sum} \circ \mathscr{C}$ witnesses, we will show that

$$L \in \mathsf{MATIME}_{\mathsf{Sum} \circ \mathscr{C}}[2^{n^{\varepsilon}}; n] \subseteq \mathsf{NTIME}[2^{n - n^{\varepsilon}}],$$

a contradiction to our assumption that $L \notin \mathsf{NTIME}[2^n/n]$.

Easy-witness condition $\widetilde{\mathsf{EW}}$. Formally, we assume the following condition for the sake of contradiction.

- For all sufficiently large $1^n \in L$, there exists $f\colon \{0,1\}^n \to \{0,1\}$ such that $\Pr_r[V_{\mathsf{PCP}}(1^n)^f(r) = 1] = 1$, and a $2^{n^{\varepsilon}}$-size $\mathsf{Sum} \circ \mathscr{C}$ circuit $H\colon \{0,1\}^n \to [0,1]$ such that $\mathbb{E}_x |f(x) - H(x)| \le 0.01$.

Algorithm $\widetilde{A_{\mathsf{PCP}}}$. To put $L \in \mathsf{MATIME}_{\mathsf{Sum} \circ \mathscr{C}}[2^{n^{\varepsilon}}; n]$, we define the following algorithm.

1. Given an input 1^n.
2. Guess a $2^{n^{\varepsilon}}$-size $\mathsf{Sum} \circ \mathscr{C}$ circuit $H\colon \{0,1\}^n \to [0,1]$.
3. Draw $r \in \{0,1\}^n$, output $V_{\mathsf{PCP}}(1^n)^H(r) \in [0,1]$.

   How to treat H as Boolean oracle? We define a probabilistic oracle $\widetilde{H}$, when querying x, $\widetilde{H}(x)$ output 1 with probability $H(x)$, and 0 otherwise. (Without loss of generality, we can assume that V_{PCP} never queries the same position more than once.) $V_{\mathsf{PCP}}(1^n)^H(r)$ is then defined to be the probability that $V(1^n)$ accepts given $\widetilde{H}$ as oracle on input r.

Now we need to establish the following two conditions.

1. $\widetilde{A_{\mathsf{PCP}}}$ is in $\mathsf{MATIME}_{\mathsf{Sum} \circ \mathscr{C}}[2^{O(n^{\varepsilon})}; n]$.
2. $\widetilde{A_{\mathsf{PCP}}}$ solves L assuming $\widetilde{\mathsf{EW}}$.

Keeping the verifier in $\mathsf{Sum} \circ \mathscr{C}$. The first condition can be achieved by making $V_{\mathsf{PCP}}(1^n)$ so simple that $V_{\mathsf{PCP}}(1^n)^H(r)$ can still be implemented by a $\mathsf{Sum} \circ \mathscr{C}$ circuit, given that $H \in \mathsf{Sum} \circ \mathscr{C}$. We are not going to cover the details here, and the real proof is more complicated than just described because we actually do not have such a PCP. We managed to resolve this issue by a careful use of PCP of proximity in [CLW20].

Smoothness condition on V_{PCP}. To show that $\widetilde{A_{\mathsf{PCP}}}$ solves L assuming $\widetilde{\mathsf{EW}}$, we will need the PCP to satisfy the following smoothness condition:

- If $\Pr_{r \in \{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^{\mathcal{O}}(r) = 1] = 1$, then for every $H\colon \{0,1\}^n \to [0,1]$ such that $\mathbb{E}_x |f(x) - H(x)| \le 0.01$, $\mathbb{E}_{r \in \{0,1\}^n}[V_{\mathsf{PCP}}(1^n)^H(r)] \ge 2/3$.

The above condition roughly says that if a proof $\mathcal{O}$ can make V_{PCP} always accepts, then a slight corrupt version of $\mathcal{O}$ can still make V_{PCP} accept with a good probability.

We can see that the smoothness condition implies that, if $\widetilde{\mathsf{EW}}$ holds, then $\widetilde{A_{\mathsf{PCP}}}$ solves L.

An important issue. Actually there is a hidden issue with the whole argument above. When $\widetilde{A_{\mathsf{PCP}}}$ guesses a $\mathsf{Sum} \circ \mathscr{C}$ circuit H, it simply guesses a list of m coefficient/circuit pairs α_i and C_i. It is not clear how to efficiently verify the guessed $H := \sum_{i \in [m]} \alpha_i C_i$ satisfies the promise that $H(x) \in [0,1]$ for all x, since a brute-force verification takes 2^n time at least.

Fortunately, there is a way to perform some efficient check on H to make sure H is "close enough" to functions from $\{0,1\}^n \to [0,1]$, and the whole argument still works. See [CLW20] for details.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[Ajt83] M Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[BV14] Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 163–173, 2014.

[Che19] Lijie Chen. Non-deterministic quasi-polynomial time is average-case hard for ACC circuits. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1281–1304. IEEE Computer Society, 2019.

[Che22] Lijie Chen. Manuscript. 2022.

[CL21] Lijie Chen and Xin Lyu. Inverse-exponential correlation bounds and extremely rigid matrices from a new derandomized XOR lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 761–771. ACM, 2021.

[CLW20] Lijie Chen, Xin Lyu, and R. Ryan Williams. Almost-everywhere circuit lower bounds from non-trivial derandomization. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1–12. IEEE, 2020.

[CR20] Lijie Chen and Hanlin Ren. Strong average-case lower bounds from non-trivial derandomization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1327–1334. ACM, 2020.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013.

[Gol08] Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

[Hås89] Johan Håstad. Almost optimal lower bounds for small depth circuits. *Advances in Computing Research*, 5:143–170, 1989.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: exponential time vs. probabilistic polynomial time. *J. Comput. Syst. Sci.*, 65(4):672–694, 2002.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 538–545. IEEE Computer Society, 1995.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229. ACM, 1997.

[Kel21] Zander Kelley. An improved derandomization of the switching lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 272–282. ACM, 2021.

[Lev87] Leonid A. Levin. One-way functions and pseudorandom generators. 7(4):357–363, 1987.

[Lyu22] Xin Lyu. Improved pseudorandom generators for AC^0 circuits. *Electron. Colloquium Comput. Complex.*, 2022.

[MW18] Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 890–901, 2018.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[Raz87] Alexander A Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Wil11] Ryan Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 115–125. IEEE Computer Society, 2011.

[Wil13] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013.

[Wil14] Ryan Williams. Nonuniform ACC circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):2, 2014.

[Wil16] R. Ryan Williams. Natural proofs versus derandomization. *SIAM J. Comput.*, 45(2):497–529, 2016.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10, 1985.

[Žák83] Stanislav Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, 1983.