Fixed-Polynomial Size Circuit Bounds

Lance Fortnow
Northwestern University
Evanston, IL, USA
fortnow@eecs.northwestern.edu

Rahul Santhanam
University of Edinburgh
Edinburgh, Scotland, UK
rsanthan@inf.ed.ac.uk

Ryan Williams
Institute for Advanced Study
Princeton, NJ, USA
ryanw@ias.edu

***Abstract*—In 1982, Kannan showed that Σ_2^{P} does not have n^k-sized circuits for any k. Do smaller classes also admit such circuit lower bounds? Despite several improvements of Kannan's result, we still cannot prove that P^{NP} does not have linear size circuits. Work of Aaronson and Wigderson provides strong evidence – the "algebrization" barrier – that current techniques have inherent limitations in this respect.**

We explore questions about fixed-polynomial size circuit lower bounds around and beyond the algebrization barrier. We find several connections, including

- **The following are equivalent:**
  - NP **is in** $\mathsf{SIZE}(n^k)$ **(has** $O(n^k)$**-size circuit families) for some** k
  - **For each** c, $\mathsf{P}^{\mathsf{NP}[n^c]}$ **is in** $\mathsf{SIZE}(n^k)$ **for some** k
  - $\mathsf{ONP}/1$ **is in** $\mathsf{SIZE}(n^k)$ **for some** k**, where** ONP **is the class of languages accepted *obliviously* by** NP **machines, with witnesses for "yes" instances depending only on the input length.**
- **For a large number of natural classes** $\mathcal{C}$ **and all** $k \geqslant 1$, $\mathcal{C}$ **is in** $\mathsf{SIZE}(n^k)$ **if and only if** $\mathcal{C}/1 \cap \mathsf{P/poly}$ **is in** $\mathsf{SIZE}(n^k)$**.**
- **If there is a** d **such that** $\mathsf{MATIME}(n) \subseteq \mathsf{NTIME}(n^d)$**, then** P^{NP} **does not have** $O(n^k)$ **size circuits for any** $k > 0$**.**
- **One cannot show** n^2**-size circuit lower bounds for** $\oplus\mathsf{P}$ **without new nonrelativizing techniques. In particular, the proof that** $\mathsf{PP} \not\subseteq \mathsf{SIZE}(n^k)$ **for all** k **relies on the (relativizing) result that** $\mathsf{P}^{\mathsf{PP}} \subseteq \mathsf{MA} \Longrightarrow \mathsf{PP} \not\subseteq \mathsf{SIZE}(n^k)$**, and we give an oracle relative to which** $\mathsf{P}^{\oplus\mathsf{P}} \subseteq \mathsf{MA}$ **and** $\oplus\mathsf{P} \subseteq \mathsf{SIZE}(n^2)$ **both hold.**

I. Introduction

Proving lower bounds for general nonuniform circuits remains one of the most difficult tasks in computational complexity. One has to go to the exponential-time version of Merlin-Arthur games to find a class provably not having a polynomial-size circuit family [BFT98]. Currently we do not have any techniques for proving EXP cannot have poly-size circuits, and certainly no techniques for super-polynomial lower bounds for NP.

A more modest goal is to show fixed-polynomial size lower bounds, i.e., lower bounds of the form n^k for some fixed k. Apart from being a first step towards super-polynomial bounds, this question is closely related to questions in derandomization, thanks to known tradeoffs between hardness and randomness ([BM84], [Yao82], [NW94]). Even this question remains open for NP – we do not even know of superlinear size lower bounds for NP.

However, we do have such lower bounds for some classes slightly above NP. In 1982, Kannan [Kan82] proved that Σ_2^{P} does not have n^k-sized circuits for any k. This result was progressively improved using relativizing techniques ([BCG$^+$96], [KW98], [Cai07]) culminating in the analogous circuit lower bound for $\mathsf{S}_2^{\mathsf{P}}$ [Cai07].

Recently, non-relativizing techniques from the theory of interactive proofs have been applied to this problem. In 2005, Vinodchandran [Vin05] showed that the class PP does not have n^k-sized circuits. Santhanam [San07] improved on this result by showing that the promise version of MA does not have n^k circuits for any fixed k. The proofs of Vinodchandran and Santhanam evade not only the relativization obstacle, but also the natural proofs obstacle [RR97].

A natural question is to ask whether these techniques can be pushed even further. Can similar results be obtained for other classes like NP, $\mathsf{P}_{\|}^{\mathsf{NP}}$, $\oplus\mathsf{P}$ and MA? This would have significant implications – for instance, n^k size lower bounds in NP for any k would separate NEXP from BPP, a long-standing open problem. In a recent influential paper [AW09], Aaronson and Wigderson give strong evidence that the answer is negative. They formalize a variant of relativization called "algebrization", and show that essentially all known structural complexity results at the polynomial time level algebrize, while several important lower bound problems, such as showing fixed-polynomial size lower bounds for NP cannot be resolved by algebrizing methods.

Thus, though we might have techniques that evade relativization and natural proofs, there is still a significant barrier to showing fixed-polynomial lower bounds for NP or even P^{NP}. In this paper, we explore the world of fixed-polynomial size lower bounds "beyond the barrier". We show some surprising connections and equivalences between questions about fixed-polynomial size circuits in this regime.

In our first batch of results, we consider various pairs of classes and show that a fixed-polynomial size lower bound for the larger class actually implies a lower bound for the smaller class. We begin with a simple observation: if AM does not have n^k-size circuits then neither does MA. We then show that fixed-polynomial circuit lower bounds for NP are equivalent to fixed polynomial-circuit lower bounds for the larger class $\mathsf{P}^{\mathsf{NP}[n^c]}$ given any fixed c, where the latter class is polynomial time with n^c adaptive queries to an NP oracle. In particular, this implies that showing fixed-polynomial lower bounds for polynomial time with nonadaptive access to an NP oracle is as hard as showing such lower bounds for NP itself.

We explore the class ONP, "Oblivious NP", implicitly defined by Chakaravarthy and Roy [CR06]. A language L is in ONP if for every n there is a single polynomial-size witness w_n, for every x in L with $|x| = n$. Thus ONP is a rather restrictive subclass of NP. Nevertheless, we show that ONP nearly captures the hardness of showing NP does not have small circuits: If NP does not have n^k-sized circuits then $\mathsf{ONP}/1$ does not have n^k-sized circuits. This result highlights a difference between the fixed-polynomial lower bound question and the super-polynomial lower bound question. The class $\mathsf{ONP}/1$ is solvable with polynomial size circuits, and we strongly believe that NP is not. Yet, from the perspective of fixed-polynomial size bounds, these two classes are equivalent!

A similar phenomenon holds more generally. We prove a result that holds for a wide variety of complexity classes such as NP, P^{NP}, MA, BPP and PP. For all these classes $\mathcal{C}$ and many more, if $\mathcal{C}$ does not have n^k-size circuit families then $\mathcal{C}/1 \cap \mathsf{P}/\mathsf{poly}$ does not have n^k-size circuits either.

We next consider the question of P^{NP} being in fixed-polynomial size. We use results from holographic proofs to show that fixed-polynomial size circuits for P^{NP} would imply that NP can be simulated by Merlin-Arthur games operating in a fixed polynomial time bound, which would be very surprising. We use this to show that a strong derandomization of MA would imply circuit lower bounds for P^{NP}, continuing a line of results relating derandomization to circuit lower bounds ([IKW02], [KI04], [San07]).

As mentioned earlier, Vinodchandran and Santhanam's circuit lower bounds ([Vin05], [San07]) use nonrelativizing techniques. The only nonrelativizing technique they use is based on interactive proof systems (see [BFL91]), arguably the only true nonrelativizing technique currently available in computational complexity. We exhibit a relativized world where $\oplus\mathsf{P}$ has a n^2-sized circuit family and the conclusions of the nonrelativizing techniques used by Vinodchandran and Santhanam also hold. This shows that a barrier analogous to algebrization also holds when trying to prove that $\oplus\mathsf{P}$ does not have quadratic-size circuit families. Actually, our oracle result rules out a certain class of techniques which algebrization is silent about – techniques in which a non-relatizing result is used in more than one way.

Questions about fixed polynomial size circuits for P and NP have also been explored by Lipton [Lip94], but with a different emphasis, namely to derive several unlikely consequences of small circuits for those classes and thereby give more evidence that lower bounds are likely to hold.

II. Preliminaries

A. Complexity Classes, Promise Problems and Advice

We assume a basic familiarity with complexity classes such as P, RP, BPP, NP, P^{NP}, MA, AM, Σ_2^{P}, $\oplus\mathsf{P}$, PP, EXP, and NEXP. The Complexity Zoo (http://qwiki.caltech.edu/wiki/ComplexityZoo) is an excellent resource for basic definitions and statements of results.

Given a complexity class C, coC is the class of languages L such that $\bar{L} \in \mathsf{C}$. Given a function $s : \mathbb{N} \to \mathbb{N}$, $\mathsf{SIZE}(s)$ is the class of Boolean functions $f = \{f_n\}$ such that for each n, f_n has Boolean circuits of size $O(s(n))$. For a Boolean function f, $Ckt(f)$ is the circuit complexity of f, i.e., the size of the smallest circuit computing f. Given a language L and an integer n, $L_n = L \cap \{0,1\}^n$.

We also require the notion of circuit size for other circuit models. These are typically defined by having one or more auxiliary inputs to a deterministic circuit and defining the language accepted by the circuit using some condition on acceptance of auxiliary inputs. $\oplus\mathsf{SIZE}(s)$ is the class of Boolean functions f computed by Parity-circuits of size $O(s)$, i.e., $f(x) = 1$ iff the circuit for f on x accepts on an odd number of auxiliary inputs.

In order to deal with promise classes in a general way, we take as fundamental the notion of a complexity measure. A complexity measure CTIME is a mapping which assigns to each pair (M, x), where M is a time-bounded machine (here a time function $t_M(x)$ is implicit) and x an input, one of three values "0" (accept), "1" (reject) and "?" (failure of CTIME promise). We distinguish between *syntactic* and *semantic* complexity measures. Syntactic measures have as their range $\{0,1\}$ while semantic measures may map some machine-input pairs to "?". The complexity measures DTIME and NTIME are syntactic (each halting deterministic or non-deterministic machine either accepts or rejects on each input), while complexity measures such as BPTIME and MATIME are semantic (a probabilistic machine may accept on an input with probability 1/2, thus failing the bounded-error promise). For syntactic measures, any halting machine defines a language, while for semantic measures, only a subset of halting machines define languages.

A promise problem is a pair (Y, N), where $Y, N \subseteq \{0,1\}^*$ and $Y \cap N = \varnothing$. A promise problem (Y, N) belongs to a class $\mathsf{CTIME}(t)$ if there is a machine M halting in time t on all inputs of length n such that M fulfils the CTIME promise on inputs in $Y \cup N$, accepting on inputs in Y and rejecting on inputs in N.

For a complexity class C, $\mathsf{Promise\text{-}C}$ is the class of promise problems which belong to C. Sometimes, when C is a syntactic class, we abuse notation and use C and Promise-C interchangeably.

A language L is in $\mathsf{CTIME}(t)/a$ if there is a machine M halting in time $t(\cdot)$ taking an auxiliary *advice* string of length $a(\cdot)$ such that for each n, there is some advice string $b_n, |b_n| = a(n)$ such that M fulfils the CTIME promise for each input x with advice string b_n and accepts x iff $x \in L$.

For syntactic classes, a lower bound with advice or for the promise version of the class translates to a lower bound for the class itself.

Proposition 1: Let CTIME be a syntactic complexity measure. If $\mathsf{CTIME}(\mathsf{poly}(n))/O(n) \not\subseteq \mathsf{SIZE}(s(n))$, then $\mathsf{CTIME}(\mathsf{poly}(n)) \not\subseteq \mathsf{SIZE}(s(o(n)))$.

Proposition 2: Let CTIME be a syntactic complexity measure. If Promise-$\mathsf{CTIME}(\mathsf{poly}(n)) \not\subseteq \mathsf{SIZE}(s(n))$, then $\mathsf{CTIME}(\mathsf{poly}(n)) \not\subseteq \mathsf{SIZE}(s(n))$.

B. Oblivious Classes

Intuitively, if a class C is defined using "proofs of acceptance" for each input and some condition on the verifiability of proofs, the oblivious version of the class C is the class of languages for which the *same* proof can be used on any input of a certain length. Oblivious versions of symmetric alternation classes were defined by Chakaravarthy and Roy [CR06] for the purpose of obtaining tight uniform characterizations of $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$. Here, we extend the definitions to non-deterministic and Merlin-Arthur classes.

Definition 3: A language L is in $\mathsf{ONTIME}(t)$ if there is a relation $R(x,y)$ computable in deterministic time $t(|x|)$, and a sequence of witnesses $\{w_n\}, n = 1 \ldots \infty$ with $|w_n| \leqslant t(n)$ such that:

1) If $x \in L$, then $R(x, w_{|x|})$ holds.
2) If $x \notin L$, then for all y, $R(x,y)$ does not hold.

Definition 4: A language L is in $\mathsf{OMATIME}(t)$ if there is a relation $R(x,y,z)$ computable in deterministic time $t(|x|)$ and a sequence of witnesses $\{w_n\}, n = 1 \ldots \infty$ with $|w_n| \leqslant t(n)$ such that:

1) If $x \in L$, then for all z, $R(x, w_{|x|}, z)$ holds.
2) If $x \notin L$, then for any y, $\Pr_z R(x,y,z) < 1/2$.

We have that $\mathsf{ONP} \subseteq \mathsf{OMA} \subseteq \mathsf{SIZE}(\mathsf{poly})$. The first inclusion is immediate; for the second inclusion, note that we can amplify the success probability of an OMA protocol above $1 - 2^{-n}$ just as we do for an MA protocol. By the union bound, there must be some random string z that gives the correct answer for every input when we have guessed the oblivious witness y. Giving y and z as advice for each input length is sufficient to decide the language.

On the other hand, all sparse languages in NP are contained in ONP, and all sparse languages in MA are contained in OMA. Thus we do not expect either of these classes to be easy – indeed, $\mathsf{ONP} = \mathsf{P}$ implies $\mathsf{NEXP} = \mathsf{EXP}$. Nor is it likely to be easy to show $\mathsf{OMA} \subseteq \mathsf{NP}$, since that would imply $\mathsf{MA_E} = \mathsf{NE}$, and resolve long-standing derandomization questions.

Using the notions above, we can get tight uniform characterizations of $\mathsf{C} \subseteq \mathsf{SIZE}(\mathsf{poly})$ for several interesting classes C.

Proposition 5: $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ iff $\mathsf{NP} \subseteq \mathsf{ONP}$ iff $\mathsf{NP} \subseteq \mathsf{OMA}$.

Proof: From the preceding discussion, it is clear that $\mathsf{NP} \subseteq \mathsf{ONP}$ implies $\mathsf{NP} \subseteq \mathsf{OMA}$, and $\mathsf{NP} \subseteq \mathsf{OMA}$ implies $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$. Thus we just need to show that $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ implies $\mathsf{NP} = \mathsf{ONP}$. We will show that under this assumption, $SAT \in \mathsf{ONP}$, and then use the fact that ONP is closed under m-reductions to conclude $\mathsf{NP} = \mathsf{ONP}$.

Assume $SAT \in \mathsf{SIZE}(n^k)$ for some k. We define the following ONP machine M for SAT. Given a formula ϕ of size n, M guesses a circuit C of size n^k for SAT on inputs of length n. If C accepts on ϕ, M uses C to find a candidate satisfying assignment w via self-reducibility and paddability of SAT. If w is a valid assignment, M accepts, otherwise it rejects. Note that no unsatisfiable formula is ever accepted in this process, moreover if C is a correct circuit for SAT, all satisfiable formulae are accepted. Thus C is an oblivious witness for SAT on length n. ■

Proposition 6: $\mathsf{EXP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ iff $\mathsf{EXP} = \mathsf{OMA}$.

Proof: The backward direction follows since $\mathsf{OMA} \subseteq \mathsf{SIZE}(\mathsf{poly})$. For the forward direction, it follows from work on instance checkers and interactive proofs ([BFL91], [BFNW93]) that if $\mathsf{EXP} \subseteq \mathsf{SIZE}(\mathsf{poly})$, then $\mathsf{EXP} = \mathsf{MA}$. The proof of this result also gives $\mathsf{EXP} = \mathsf{OMA}$. ■

Since Impagliazzo, Kabanets and Wigderson [IKW02] showed that $\mathsf{NEXP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ implies $\mathsf{NEXP} = \mathsf{EXP}$ we have the following corollary.

Corollary 7: $\mathsf{NEXP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ iff $\mathsf{NEXP} = \mathsf{MA}$ iff $\mathsf{NEXP} = \mathsf{OMA}$.

III. Translations of Circuit Lower Bounds

The question of fixed polynomial size circuit lower bounds was first considered by Kannan, who proved lower bounds for Σ_2^{P}.

Theorem 8 (Kannan [Kan82]): For any $k > 0$, $\Sigma_2^{\mathsf{P}} \not\subseteq \mathsf{SIZE}(n^k)$.

Theorem 8 has been strengthened progressively in a sequence of papers ([BCG$^+$96], [KW98], [Cai07]) and the smallest uniform complexity class for which we can show unconditional lower bounds is $\mathsf{S}_2^{\mathsf{P}}$ [Cai07]. Circuit lower bounds have recently been shown for the promise version of MA [San07] but showing such lower bounds for uniform

MA and smaller classes remains an important open question. Such lower bounds for NP, apart from being interesting in their own right, would also separate BPP and NEXP, which would be a major breakthrough in the area of derandomization.

One obstacle to proving lower bounds for classes smaller than S_2^P is that such results cannot relativize. There have been non-relativizing results in this area ([Vin05], [San07]) but there is a paucity of non-relativizing techniques apart from the arithmetization technique used in work on interactive proofs ([LFKN92], [Sha92]). Recently, Aaronson and Wigderson [AW09] have introduced the notion of algebrization which in fact covers all known complexity-theoretic techniques for lower bounds at the polynomial-time level; they show that fixed-polynomial size lower bounds for NP cannot be proven by algebrizing methods.

Given the insufficiency of current techniques to prove unconditional lower bounds, we focus on reductions between circuit lower bounds for various classes. We show for various pairs of classes B and C, where $\mathsf{B} \subseteq \mathsf{C}$, that a fixed polynomial lower bound for C also implies a fixed polynomial lower bound for the smaller class B. We call such results *translations* of circuit lower bounds.

Such a translation result can be interpreted in two ways. An optimist would say that we are making our lower bound task easier: in order to prove a lower bound for B we now only need to prove a lower bound for the weaker class C. A pessimist would say that this gives additional evidence that proving a lower bound for C is hard, since this would automatically result in a stronger lower bound.

One example of a translation is the result that if the polynomial hierarchy contains a language of superpolynomial circuit complexity, then so does NP. However, this result no longer holds if we consider fixed polynomial size. Indeed, if it did, we would already have a superlinear circuit lower bound for NP, by Theorem 8.

We begin by giving a simple example: translating circuit lower bounds for AM to circuit lower bounds for MA. This result seems to have been observed independently by several researchers.

Theorem 9: For any $k > 0$, $\mathsf{AM} \not\subseteq \mathsf{SIZE}(n^k)$ iff $\mathsf{MA} \not\subseteq \mathsf{SIZE}(n^k)$.

Proof: For the forward direction, if $\mathsf{MA} \subseteq \mathsf{SIZE}(n^k)$ then $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$, which is known to imply $\mathsf{AM} = \mathsf{MA}$ [AKSS95]. The backward direction follows from the fact that $\mathsf{MA} \subseteq \mathsf{AM}$ [BM88]. ■

Next, we consider fixed polynomial size lower bounds for the class $\mathsf{P}^{\mathsf{NP}[n^q]}$, which lies between NP and P^{NP}. We show that such lower bounds would in fact imply fixed polynomial lower bounds for NP. This result also shows that fixed polynomial lower bounds for $\mathsf{P}_{\|}^{\mathsf{NP}}$ yield lower bounds for NP, since $\mathsf{P}_{\|}^{\mathsf{NP}} = \mathsf{P}^{\mathsf{NP}[O(\log n)]}$ ([BH91], [Hem89]).

We note that improving our results to show that $\mathsf{NP} \subseteq \mathsf{SIZE}(n^k)$ implies $\mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{SIZE}(n^{k'})$ for some fixed k' depending only on k would require nonrelativizing techniques. This is because there is a relativized world [BFFT01] where $\mathsf{NEXP} \subseteq \mathsf{P}^{\mathsf{NP}} \cap \mathsf{SIZE}(\mathsf{poly})$. In this world, NP has fixed polynomial-size circuits, but P^{NP} does not.

Theorem 10: Fix any constant q. There is a k such that $\mathsf{NP} \subseteq \mathsf{SIZE}(n^k)$ iff there is a k' such that $\mathsf{P}^{\mathsf{NP}[n^q]} \subseteq \mathsf{SIZE}(n^{k'})$.

Proof: One direction is obvious. We prove that choosing $k' = qk^2$ is sufficient in the reverse direction. Namely, we show that $\mathsf{NP} \subseteq \mathsf{SIZE}(n^k)$ implies $\mathsf{P}^{\mathsf{NP}[n^q]} \subseteq \mathsf{SIZE}(n^{qk^2})$.

Let M be an deterministic polynomial time machine that makes n^q oracle calls to SAT. Let $b_1, \ldots, b_j$ be Boolean, where $j \leqslant n^q$. Define the *pseudo-simulation* of $M(x)$ on $b_1, \ldots, b_j$ to be the following nondeterministic computation:

Simulate $M(x)$ over its first j queries, simulating the ith oracle call as follows: (for $i = 1, \ldots, j$):

- *if $b_i = 0$ then the simulation continues, presuming that the ith query answered "no",*
- *if $b_i = 1$ then a variable assignment to the ith query is guessed, and the simulation continues if the assignment satisfies the ith query, otherwise it* rejects.

If the simulation itself accepts or rejects at any time, then accept *or* reject *accordingly. If the simulation passes all j query steps above without rejecting, then* accept. *Otherwise,* reject.

Define the language L_q to be the set of $\langle x, j, b_1, \ldots, b_{n^q} \rangle$ such that $1 \leqslant j \leqslant n^q$, $b_i \in \{0, 1\}$, and for all $i = 1, \ldots, j$, the pseudo-simulation of $M(x)$ on $b_1, \ldots, b_j$ accepts.

(Note on input $\langle x, j, b_1, \ldots, b_{n^q} \rangle$, the bits $b_{j+1}, \ldots, b_{n^q}$ are ignored. We choose this definition of L_q so that the final circuit family is easy to describe.)

Intuitively, L_q takes some candidate query answers, and checks that the "yes" query answers are correct up to some point. (We need a trick to determine that "no" query answers are correct.)

Clearly L_q is in NP. Thus it is captured by a circuit family $\{C_n^q\}$ of n^k size, by assumption. Now define a machine N that on input x and circuit C does the following:

Guess bits $b_1, \ldots, b_{n^q}$. For $j = 1, \ldots, n^q$, check if $C(x, j, b_1, \ldots, b_{j-1}, 1, b_{j+1}, \ldots, b_{n^q}) = b_j$. Accept iff all checks passed, and the pseudo-simulation of $M(x)$ on $b_1, \ldots, b_{n^q}$ accepts.

Intuitively, N tries to use the circuit C to determine that its guesses $b_1, \ldots, b_{n^q}$ are the correct query outcomes. Notice that $L(N)$ is in NP. Therefore $L(N)$ is also captured by a circuit family $\{C_n^N\}$ of n^k size. Finally, set up a circuit family $\{D_n\}$ defined as:

$$D_n(x) := C^N_{n+(n+n^q)^k+c}(x, C^q_{n+n^q+c}),$$

for an appropriate constant c. Note the circuit D_n is of size $O(n^{qk^2})$.

We now prove that for all x, $D_n(x) = 1$ iff $M(x)$ accepts. Consider the for-loop of N. We claim the following invariant holds:

For all $j = 1, \ldots, n^q$, the jth iteration of the for-loop in N is reached without failing a check, iff the first j bits of $b_1, \ldots, b_{n^q}$ are the answers to the first j queries that M makes on x.

The claim can be proved for all j by induction. When $j = 1$ and $b_j = 1$, the call to C^q checks that there is a satisfying assignment to the first query. When $j = 1$ and $b_j = 0$, the call to C^q still checks that there is a satisfying assignment to the first query (by flipping b_j to be 1), but N only continues if C^q outputs 0. That is, the pseudo-simulation on b_1 rejects, hence there is no satisfying assignment for the first query. In the jth iteration, we have (by induction) that $b_1, \ldots, b_{j-1}$ are the answers to the first $j - 1$ queries of $M(x)$. Then for $b_j = 1$, the respective check succeeds iff the jth query can be satisfied by an assignment. When $b_j = 0$, the check succeeds iff the jth query is unsatisfiable.

It follows that $D_n(x) = 1$ iff the pseudo-simulation of $M(x)$ on $b_1, \ldots, b_{n^q}$ accepts where b_i is the outcome of the ith query on $M(x)$, which is true iff $M(x)$ accepts. This completes the proof. ■

The next result shows that fixed polynomial lower bounds for NP also translate to fixed polynomial lower bounds for the oblivious version of NP (using 1 bit of advice). An interesting aspect of this result is that it illustrates that proving superpolynomial circuit lower bounds is a very different problem than proving fixed polynomial lower bounds. On the one hand, $\mathsf{ONP}/1 \subseteq \mathsf{SIZE}(\mathsf{poly})$ and we do not expect $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$, thus the two inclusions are very unlikely to be equivalent. On the other hand, the inclusions of the two classes in fixed polynomial size are equivalent.

Theorem 11: For any k, $\mathsf{NP} \not\subseteq \mathsf{SIZE}(n^k)$ iff $\mathsf{ONP}/1 \not\subseteq \mathsf{SIZE}(n^k)$.

Proof: The "if" direction is easy. If $\mathsf{ONP}/1$ does not have circuits of size n^k, then $\mathsf{NP}/1$ does not have circuits of size n^k. Since NP is a syntactic class, this implies that NP does not have circuits of size n^k.

The other direction is more involved. Assume NP does not have circuits of size n^k. We consider two cases. If $\mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$, then by Proposition 5, $\mathsf{NP} = \mathsf{ONP}$, hence ONP does not have circuits of size n^k.

If $\mathsf{NP} \not\subseteq \mathsf{SIZE}(\mathsf{poly})$, then $SAT \notin \mathsf{SIZE}(\mathsf{poly})$. We use the fact that there is a "smoothly parameterized" version of Proposition 5. If $\mathsf{NP} \subseteq \mathsf{SIZE}(s)$, then we get $\mathsf{NP} \subseteq \bigcup_{c>0} \mathsf{ONTIME}(s^c)$, for *arbitrary* circuit size s. By letting s be the circuit complexity of SAT, we get that $SAT \in \mathsf{ONTIME}(s^c)$ for some c, but SAT does not have circuits of size $s - 1$. We then scale this separation down using an advice-efficient padding argument to conclude that a padded version of SAT is in ONP but does not have circuits of size $O(n^k)$.

The above is a brief sketch. We now proceed more formally. We define the following language L:

$$\begin{aligned} L &= \{x1^r \mid x \in SAT, r \text{ is a power of } 2, \\ &\qquad r > |x|, Ckt(SAT_{|x|}) \leqslant (|x| + r)^{2k}\}. \end{aligned}$$

First we show $L \in \mathsf{ONP}/1$, and then we show $L \notin \mathsf{SIZE}(n^k)$.

We define a non-deterministic polynomial time machine M taking one bit of advice, such that when the advice bit is correct for length n, there is a polynomial-size witness w_n which works for any input of that length. Given an input y of length n, M first checks if it can be decomposed as $x1^r$ for r a power of 2, such that $r > |x|$. For any input y, there can be at most one such decomposition since $|y| \geqslant r > |y|/2$. This check can be performed in linear time, and if it succeeds, the corresponding x and r can be obtained. The bit of advice for M is assumed to be 1 if and only if $Ckt(SAT_{|x|}) \leqslant (|x| + r)^{2k}$. This is just one bit of information given n, since n uniquely determines $|x|$ and r. If the advice bit is 0, then M rejects. Otherwise, M guesses a circuit C of size n^{2k}. It simulates C on x. If C accepts on x, it uses self-reducibility and paddability of SAT to find a candidate satisfying assignment for x. If the assignment works, M accepts, otherwise M rejects.

Clearly, M runs in polynomial time. Also, there is a single witness of size $\mathsf{poly}(n)$, namely a correct circuit C for SAT on inputs of length $|x|$ which works for any input $x1^r \in L$, when M is given the correct bit of advice. Thus $L \in \mathsf{ONP}/1$.

Assume, for the purpose of contradiction, that $L \in \mathsf{SIZE}(n^k)$. Hence there is a sequence of circuits D_n of size $O(n^k)$ deciding L_n for each n. We show that this implies that for infinitely many m, there is a circuit C_m of size less than $Ckt(SAT_m)$ deciding SAT on inputs of length m. We define the circuits C_m as follows. Given an input x of length m, our C_m has hard-coded the least $r(m) = 2^i$ such that $r(m) \geqslant m$ and $Ckt(SAT_m) \leqslant (m + r(m))^{2k}$. Such an $r(m)$ exists for each m. Also, there must be infinitely many m such that $r(m) > 2m$, for otherwise $Ckt(SAT_m) \leqslant (3m)^{2k} = O(\mathsf{poly}(m))$ almost everywhere, which is a contradiction to our assumption that SAT does not have polynomial-size circuits.

Now, for each m such that $r(m) > 2m$, we have $Ckt(SAT_m) > (m + r(m)/2)^{2k}$, just by assumption on minimality of $r(m)$. Thus for these m, $Ckt(SAT_m) >$

$(m + r(m))^{2k}/2^{2k}$. When C_m is given x of length m, it runs $D_{m+r(m)}(x1^{r(m)})$, using the hard-coded value for $r(m)$ and a hard-coded copy of $D_{m+r(m)}$. C_m decides SAT_m correctly and has size at most $O((m + r(m))^k)$, by the assumption on size of $\{D_n\}$. For large enough m, $O((m+r(m))^k) < (m+r(m))^{2k}/2^{2k}$, which implies that for infinitely many m, SAT_m has circuits of size less than $Ckt(SAT_m)$ – a contradiction. ■

A corollary of Theorem 11 is that if NP doesn't have circuits of size $O(n^k)$, then $\mathsf{NP}/1 \cap \mathsf{SIZE}(\mathsf{poly})$ doesn't have circuits of size $O(n^k)$. This follows since $\mathsf{ONP}/1 \subseteq \mathsf{NP}/1 \cap \mathsf{SIZE}(\mathsf{poly})$. In fact, this kind of translation result, showing that a fixed polynomial circuit lower bound for a class implies a fixed polynomial circuit lower bound for a language in the class with polynomial-size circuits, holds much more generally, for *any* complexity measure satisfying a certain natural condition. This condition corresponds to "closure under deterministic transductions" as defined by van Melkebeek and Pervyshev [vMP06], but rather than state it formally, we just observe that our proof works for any reasonable complexity class for which we wish to show a circuit lower bound. The proof abstracts out the padding argument in the proof of Theorem 11.

Theorem 12: Let C be a complexity class such as NP, P^{NP}, MA, BPP or PP. If C does not have circuits of size $O(n^k)$, then $\mathsf{C}/1 \cap \mathsf{SIZE}(\mathsf{poly})$ does not have circuits of size $O(n^k)$.

Theorem 12 can be stated as an *equivalence* for the polynomial-time versions of syntactic measures.

Corollary 13: Let CTIME be a syntactic measure, and C be the polynomial-time version of that measure, such as $\mathsf{NP}, \mathsf{P}^{\mathsf{NP}}$ or PP. C does not have circuits of size n^k iff $\mathsf{C}/1 \cap \mathsf{SIZE}(\mathsf{poly})$ does not have circuits of size n^k.

The forward implication in Corollary 13 follows from Theorem 12, and the backward implication from Proposition 1.

Proof of Theorem 12. Let $L' \in \mathsf{C}$ be a language such that L' does not have circuits of size $O(n^k)$. We define a padded language L'' such that L'' in $\mathsf{C}/1 \cap \mathsf{SIZE}(\mathsf{poly})$ and L'' does not have circuits of size $O(n^k)$. L'' is defined from L' in exactly the same way as the language L is defined from SAT in the proof of Theorem 11:

$$\begin{aligned} L'' \quad = \quad & \{x1^r \mid x \in L', r \text{ is a power of } 2, \\ & r \geqslant |x|, Ckt(L'_{|x|}) \leqslant (|x| + r)^{2k}\}. \end{aligned}$$

The proof that $L'' \notin \mathsf{SIZE}(n^k)$ is exactly as in the proof of Theorem 11. For the upper bound, we define a CTIME machine M with one bit of advice accepting L''. Given an input y of length n, M first decomposes y as $x1^r$, where r is a power of 2 and $r \geqslant |x|$, if such a decomposition is possible. If not, M rejects. If such a decomposition exists, it uniquely determines $|x|$ and r. The bit of advice just specifies if $Ckt(L'_{|x|}) \leqslant (|x| + r)^{2k}$. If yes, M simulates the CTIME machine for L'' on x, accepting iff the simulated machine does. If not, M rejects.

If CTIME is able to simulate deterministic time, as is the case for all the complexity classes in the statement of the theorem, then $L'' \in \mathsf{C}/1$, since every stage of the process above, including the simulation of the machine for L', can be implemented in polynomial time. Also, just by using the optimal circuits for L' to decide L'' on appropriately padded inputs, it follows that L'' has polynomial size circuits, in fact circuits of size $O(n^{2k})$. □

IV. On Small Circuits For P^{NP}

We show that if there were circuits for $\mathsf{P}^{\mathsf{NP}[n^c]}$ having size smaller than n^c, then one can speed up nondeterministic computations by adding randomness. This is a result in the spirit of Lipton's work [Lip94] on consequences of classes having small circuits.

Theorem 14: If $\mathsf{P}^{\mathsf{NP}[n^c]}$ has $O(n^k)$ size circuits, then $\mathsf{NTIME}[n^c] \subseteq \mathsf{MATIME}[n^k\mathsf{polylog}(n)]$.

Proof: Let M be a nondeterministic $O(n^c)$ time machine. Define a nondeterministic machine M' as follows. On an input x, first compute an equivalent SAT instance $\phi_{M,x}$ of length $O(n^c\mathsf{polylog}(n))$ using a succinct version of Cook's theorem [Coo88]. Then transform $\phi_{M,x}$ into a formula $\psi_{M,x}$ which has PCPs of length $O(n^c\mathsf{polylog}(n))$ with the property that any proposed proof can be verified in $O(\mathsf{polylog}(n))$ time. Such PCPs exist, due to work of Ben-Sasson et al. [BSGH$^+$05]. Finally, nondeterministically guess a proof, and accept iff the proof is valid.

Note the lexicographically first valid proof of $\psi_{M,x}$ can easily be computed in $\mathsf{P}^{\mathsf{NP}[n^c\mathsf{polylog}(n)]}$. Thus the hypothesis of the theorem implies that there is an $O(n^k\mathsf{polylog}(n))$ size circuit family $\{C_n\}$ with the following properties. On an input $\langle x, i\rangle$ with $|i| = \log|x|$, if $M'(x)$ accepts, then $C_{|\langle x,i\rangle|}(\langle x, i\rangle)$ outputs the ith bit of the lexicographically first valid proof of $\psi_{M,x}$. (We assume without loss of generality that the proof begins with a description of $\psi_{M,x}$.) If $M'(x)$ rejects, then the circuit outputs 0 on every input.

Our MA simulation of M on input x existentially guesses a circuit $C' = C_{|x|+\log|x|}$ of size $O(n^k\mathsf{polylog}(n))$. Then it runs the polylogarithmic time verifier for $\psi_{M,x}$. When a bit of $\psi_{M,x}$ or a bit of the proof is requested by the verifier, the bit is obtained by simulating $C'(x, i)$ with the appropriate index i, returning the output. The simulation requires only $O(n^k\mathsf{polylog}(n))$ time. ■

From Theorem 14, we derive a new example of the phenomenon that derandomization results imply circuit lower bounds ([IKW02], [KI04], [San07]).

Theorem 15: If there is a d such that $\mathsf{MATIME}(n) \subseteq \mathsf{NTIME}(n^d)$, then P^{NP} does not have $O(n^k)$ size circuits for any $k > 0$.

Proof: Suppose, on the contrary, that the assumption holds and P^{NP} does have circuits of size $O(n^k)$ for some k. We derive a contradiction. From Theorem 14, we have that $\mathsf{NP} \subseteq \mathsf{MATIME}(n^k \mathsf{polylog}(n))$. From the assumption that $\mathsf{MATIME}(n) \subseteq \mathsf{NTIME}(n^d)$, by padding, we have that $\mathsf{MATIME}(n^{k+1}) \subseteq \mathsf{NTIME}(n^{d(k+1)})$. Thus we have $\mathsf{NP} \subseteq \mathsf{NTIME}(n^{d(k+1)})$, which is a contradiction to the nondeterministic time hierarchy theorem ([Coo72], [SFM78], [Ž́83]). ■

Theorem 15 can be interpreted as a "low-end" analogue of the Impagliazzo-Kabanets-Wigderson result [IKW02] that $\mathsf{MA} \neq \mathsf{NEXP}$ implies $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$.

V. Relativized Circuit Upper Bound

Vinodchandran's circuit lower bound for PP [Vin05] raises the possibility that similar lower bounds might be provable for other counting classes. A natural candidate for such a class is $\oplus\mathsf{P}$, since Toda's landmark result [Tod91] proves that the Polynomial Hierarchy randomly reduces to $\oplus\mathsf{P}$, and we know fixed polynomial-size lower bounds for the Polynomial Hierarchy. Thus far, even the relativized status of the question of whether $\oplus\mathsf{P}$ has small circuits has remained unresolved.

We first give an oracle relative to which $\oplus\mathsf{P}$ has quadratic-size circuits. We use a previously published oracle due to Beigel, Buhrman and Fortnow [BBF98] and show using relativizing arguments that this oracle gives us what we require.

Theorem 16: There exists an oracle relative to which $\oplus\mathsf{P} \subseteq \mathsf{SIZE}(n^k)$, for some constant k.

Proof: Beigel, Buhrman and Fortnow [BBF98] created an oracle relative to which

$$\mathsf{P} = \oplus\mathsf{P} \text{ and } \mathsf{NP} = \mathsf{EXP}.$$

We will use the same oracle.

By Valiant-Vazirani [VV86], NP is in $\mathsf{BPP}^{\oplus\mathsf{P}}$. Since we have $\oplus\mathsf{P} = \mathsf{P}$ and $\mathsf{BPP} \subseteq \mathsf{SIZE}(\mathsf{poly})$ we have $\mathsf{EXP} = \mathsf{NP} \subseteq \mathsf{SIZE}(\mathsf{poly})$.

Under a standard padding argument $\oplus\mathsf{P} = \mathsf{P}$ implies $\oplus\mathsf{E} = \mathsf{E}$ and so we have

$$\oplus\mathsf{E} \subseteq \mathsf{EXP} \subseteq \mathsf{SIZE}(\mathsf{poly}).$$

Let L be a linear-time complete set for $\oplus\mathsf{E}$. There is some k such that L is in $\mathsf{SIZE}(n^k)$ and since L is linear-time complete we have

$$\oplus\mathsf{P} \subseteq \oplus\mathsf{E} \subseteq \mathsf{SIZE}(n^k).$$

■

By an analysis of the proof of Beigel, Buhrman and Fortnow [BBF98], we can show that $\oplus\mathsf{P} \subseteq \mathsf{SIZE}(n^4)$ relative to their oracle. With a more careful reworking of their proof we can show $\oplus\mathsf{P} \subseteq \mathsf{SIZE}(n^2)$ for a relativized world (proof omitted).

The fact that $\oplus\mathsf{P}$ has small circuits in a relativized world does not compel skepticism that a lower bound can be proved, since for instance Vinodchandran's lower bound for PP doesn't relativize ([Vin05], [Aar06]). We show something stronger: that a new non-relativizing idea is required to get circuit lower bounds for $\oplus\mathsf{P}$.

The nonrelativizing part of Vinodchandran's proof utilizes the fact that $\mathsf{P}^{\mathsf{PP}} \subseteq \mathsf{SIZE}(\mathsf{poly})$ implies $\mathsf{P}^{\mathsf{PP}} \subseteq \mathsf{MA}$ (see [BFL91]). However $\mathsf{P}^{\oplus\mathsf{P}} \subseteq \mathsf{MA}$ (and more strongly, $\mathsf{EXP} = \mathsf{BPP}$) relative to the Beigel-Buhrman-Fortnow oracle. We have the following contrasting results.

Corollary 17 (Vinodchandran): Relative to all oracles, if $\mathsf{P}^{\mathsf{PP}} \subseteq \mathsf{MA}$ then $\mathsf{PP} \not\subseteq \mathsf{SIZE}(n^k)$ for any fixed k.

Corollary 18 (Theorem 16): There is an oracle relative to which $\mathsf{P}^{\oplus\mathsf{P}} \subseteq \mathsf{MA}$ and $\oplus\mathsf{P} \subseteq \mathsf{SIZE}(n^2)$.

Thus to prove $\oplus\mathsf{P} \not\subseteq \mathsf{SIZE}(n^2)$, one would need nonrelativizing techniques beyond those used by Vinodchandran. It's an interesting open problem to show an analogue of Theorem 16 in the Aaronson-Wigderson framework of algebrization [AW09].

References

[Aar06] Scott Aaronson. Oracles are subtle but not malicious. In *Proceedings of 21st Annual IEEE Conference on Computational Complexity*, pages 340–354, 2006.

[AKSS95] Vikraman Arvind, Johannes Kobler, Uwe Schoning, and Rainer Schuler. If NP has polynomial-size circuits, then MA=AM. *Theoretical Computer Science*, 137(2):279–282, 1995.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):1–54, 2009.

[BBF98] Richard Beigel, Harry Buhrman, and Lance Fortnow. NP might not be as easy as detecting unique solutions. In *Proceedings of 30th STOC Conference*, pages 203–208. ACM, New York, 1998.

[BCG$^+$96] Nader Bshouty, Richard Cleve, Ricard Gavalda, Sampath Kannan, and Christino Tamon. Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences*, 52(2):268–286, 1996.

[BFFT01] Harry Buhrman, Steve Fenner, Lance Fortnow, and Leen Torenvliet. Two oracles that force a big crunch. *Computational Complexity*, 10(2):93–116, 2001.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[BFT98] Harry Buhrman, Lance Fortnow, and Thomas Thierauf. Nonrelativizing separations. In *Proceedings of 13th Annual IEEE Conference on Computational Complexity*, pages 8–12, 1998.

[BH91] Samuel R. Buss and Louise Hay. On truth-table reducibility to SAT. *Information and Control*, 90(2):86–102, 1991.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequence of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984.

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Computing and System Sciences*, 36(2):254–276, 1988.

[BSGH$^+$05] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Short PCPs verifiable in polylogarithmic time. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 120–134, 2005.

[Cai07] Jin-Yi Cai. $\mathrm{S}_2^{\mathrm{P}} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$. *Journal of Computer and System Sciences*, 73(1):25 – 35, 2007.

[Coo72] Stephen Cook. A hierarchy for nondeterministic time complexity. In *Proceedings of the 4th Annual ACM Symposium on Theory of Computing*, pages 187–192, Denver, Colorado, 1–3 May 1972.

[Coo88] Stephen Cook. Short propositional formulas represent nondeterministic computations. *Inf. Process. Lett.*, 26(5):269–270, 1988.

[CR06] Venkat Chakaravarthy and Sambuddha Roy. Oblivious symmetric alternation. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 230–241, 2006.

[Hem89] Lane Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

[Kan82] Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1):40–56, 1982.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KW98] Johannes Kobler and Osamu Watanabe. New collapse consequences of NP having small circuits. *SIAM Journal on Computing*, 28(1):311–324, 1998.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the Association for Computing Machinery*, 39(4):859–868, 1992.

[Lip94] Richard Lipton. Some consequences of our failure to prove non-linear lower bounds on explicit functions. In *Proceedings of 9th Annual Structure in Complexity Theory Conference*, pages 79–87, 1994.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[San07] Rahul Santhanam. Circuit lower bounds for Merlin-Arthur classes. In *Proceedings of 39th Annual Symposium on Theory of Computing*, pages 275–283, 2007.

[SFM78] Joel Seiferas, Michael Fischer, and Albert Meyer. Separating nondeterministic time complexity classes. *Journal of the ACM*, 25(1):146–167, January 1978.

[Sha92] Adi Shamir. IP = PSPACE. *Journal of the Association for Computing Machinery*, 39(4):869–877, 1992.

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[Vin05] Variyam Vinodchandran. A note on the circuit complexity of PP. *Theoretical Computer Science*, 347(1-2):415–418, 2005.

[vMP06] Dieter van Melkebeek and Konstantin Pervyshev. A generic time hierarchy for semantic models with one bit of advice. In *Proceedings of 21st Annual IEEE Conference on Computational Complexity*, pages 129–144, 2006.

[VV86] Leslie Valiant and Vijay Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[Yao82] Andrew Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[Ž́83] Stanislav Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, October 1983.