

# Oracles Are Subtle But Not Malicious

Scott Aaronson

April 4, 2005

## 1 Introduction

Thanks! It's great to be back home in Berkeley. In the past, most of the talks you've heard me give have been quantum, meaning most of you had to pretend to care, and I had do my stand-up routine to hold your attention. But for the past few months, I've been worrying about something that hopefully everyone here will agree is important. And that's, can we prove nontrivial lower bounds on the circuit size of explicit Boolean functions? Now I want to reassure you that I haven't *completely* lost my mind, despite having been "institutionalized" for the past year. (Or maybe I already lost it long before that.) In any case, no, I'm not gonna talk about P versus NP. It's exactly three days too late for that. But there are other questions that *seem* more reasonable: for example, is there a language in NP that does not have linear-size circuits? The language doesn't have to be SAT—it could have witnesses of size  $n^{100}$ , that take time  $n^{10000}$  to verify. I don't care.

If we attack this question, we find that it's like that rabbit in *Monty Python and The Holy Grail*. Actually there are two rabbits: one is called Razborov-Rudich, and the other is called relativization.

On the other hand, it's *not* true that we have no explicit superlinear circuit lower bounds—or rather, this depends completely on what you mean by "explicit." For there's one technique that bypasses the Razborov-Rudich rabbit: the one bequeathed to us by Gödel and Turing, diagonalization. Sometimes diagonalization can be surprisingly powerful. For example, Kannan used it in 1982 to show that if we go one level up in the hierarchy, then already  $\Sigma_2^P$  does not have linear-size circuits. In fact for every  $k$  there exists a language in  $\Sigma_2^P$  that does not have circuits of size  $n^k$ .

You might wonder, can we improve Kannan's theorem? A decade ago, Bshouty et al. (with a contribution from Köbler and Watanabe) proved something that I now think of as one of the most underrated results in complexity theory. They proved that if  $\text{NP} \subset \text{P/poly}$ , then a  $\text{ZPP}^{\text{NP}}$  machine can actually *learn* the circuits for NP. (The proof uses nontrivial ideas, mainly approximate counting.) So in particular, if  $\text{NP} \subset \text{P/poly}$  then PH collapses to  $\text{ZPP}^{\text{NP}}$ . As an immediate corollary, for every fixed  $k$  there exists a language in  $\text{ZPP}^{\text{NP}}$  that does not have circuits of size  $n^k$ .

So the strategy seems clear: we should just keep working our way down the complexity hierarchy, showing smaller and smaller classes don't have linear-size circuits, until we hit NP. But here we run into the relativization rabbit. Wilson gave oracles relative to which NP and even  $\text{P}^{\text{NP}}$  *do* have linear-size circuits.

At this point, you might be thinking one of two things: first, that these oracle results are the kind of thing "only a Fortnow could love." Everyone knows that oracle results only formalize the obvious, and in any case, you can get a relativized world where  $2 + 2 = 5$ , or where anything else you want to be true is true. At the other extreme, maybe some of you are thinking, well, if there are these oracle results, I guess we should give up. We do have nonrelativizing techniques, but none powerful enough to apply to this sort of circuit lower bound.

So, what will I show in this talk?

- An oracle where PP has linear-size circuits. This resolves a long-standing open problem of Fortnow. Previously this was only known for classes such as  $\text{P}^{\text{NP}}$  and MA.

- By nonrelativizing techniques, PP does not have circuits of size  $n^k$  for any  $k$ . This was previously shown by Vinodchandran. I'll give a more direct proof, which even shows that PP does not have quantum circuits of size  $n^k$ —not even quantum circuits with quantum advice.

This provides only the second example of a nonrelativizing separation in complexity theory—and the first example for a complexity class we really care about. The first example, due to Buhrman, Fortnow, and Thierauf, showed that  $\text{MA}_{\text{EXP}} \not\subseteq \text{P/poly}$ . Unfortunately, this doesn't scale down to MA. (Luca points out that  $\text{NP} \neq \text{MIP}$  was arguably the first nonrelativizing separation. Well, OK—but what I'm really interested in here is separations that don't follow immediately from collapses. Or in other words, separations of classes for which one doesn't need nonrelativizing techniques just to get any idea of where they fit in the hierarchy.)

My result also provides the first nontrivial example of a *quantum* circuit lower bound. For years, people complained to me, enough with these oracle results, we want to see unrelativized limitations of quantum computers! Well, are you happy? Is this what you wanted?

- If there's time, I'll talk about the result of Bshouty et al.'s, and whether it can be improved—for example, from  $\text{ZPP}^{\text{NP}}$  to  $\text{ZPP}_{\parallel}^{\text{NP}}$ .

## 2 PP Oracle

So, an oracle where PP has linear-size circuits. Let  $M_1, M_2, \dots$  be an enumeration of  $\text{PTIME}(n^{\log n})$  machines. On input length  $n$ , we only have to ensure  $M_1, \dots, M_n$  have linear-size circuits, for then every  $M_i$  will have linear-size circuits on all but finitely many  $n$ . (Standard trick.) Also, let's assume for simplicity that given an input of length  $n$ , the machine  $M_i$  only queries the “ $n^{\text{th}}$  part” of the oracle tape.

Our oracle  $A$  will have  $2^{5n}$  rows, each labeled by  $r \in \{0, 1\}^{5n}$ , and  $n2^n$  columns, each labeled by  $\langle i, x \rangle$  where  $i \in \{1, \dots, n\}$  and  $x \in \{0, 1\}^n$ . Given  $r, i, x$  as input, the oracle returns  $A(r, i, x)$ . Let's say that initially,  $A(r, i, x) = 0$  for all  $r, i, x$ .

The obvious idea is to pick a row uniformly at random, and write down what each PP machine does on each input. That should make PP have linear-size circuits, since we just hardwire into each circuit the address of the “undisclosed location” on the oracle tape, that tells the circuit what the PP machines do.

What's the problem? Well, we're dealing with PP machines. *They're* monitoring the whole tape too. As soon as we pick a row and encode what the machines do, some of the machines might do something different as a result of our having encoded that row. Then we have to encode another row, and so on ad infinitum. If they were NP machines, at least we could placate them with accepting witnesses—but PP machines?

So we're fighting a war of attrition. This is trench warfare. Our strategy will be to keep playing this game, encoding more rows at every iteration, until the PP machines all get so exhausted that they don't even notice when we encode another row. And then we win.

Let  $M_{i,x}(A)$  be a Boolean function that equals 1 if  $M_i$  accepts  $x$  on oracle  $A$ , and 0 otherwise. Important fact about PP: for all  $i, x$ , there exists a multilinear polynomial  $p_{i,x}(A)$  such that

- (i) If  $M_{i,x}(A) = 1$  then  $p_{i,x}(A) \geq 1$ .
- (ii) If  $M_{i,x}(A) = 0$  then  $p_{i,x}(A) \leq -1$ .
- (iii)  $\deg(p_{i,x}) \leq n^{\log n}$ .
- (iv)  $|p_{i,x}(A)| \leq 2^{n^{\log n}}$ .

So we'll define a global progress measure:

$$Q(A) = \prod_{\substack{i=1 \dots n \\ x \in \{0,1\}^n \\ b \in \{0,1\} \\ k=0 \dots n^{\log n}}} \left[ 2^{2k-3} + \left( 2^k + (-1)^b p_{i,x}(A) \right)^2 \right].$$

(Like many weird-looking definitions, this one bears the scars of battle. My first idea was just to multiply all the  $p_{i,x}$ 's. This leads to a very interesting mathematical question, but one that I wasn't able to solve.)

What can we say about  $Q$ ? Well,  $\deg(Q) = 2^{n+o(n)}$ . Also,  $2^{-2^{n+o(n)}} \leq Q(A) \leq 2^{2^{n+o(n)}}$  for all  $A$ . In particular,  $Q(A) \geq 0$ .

Here's the crucial claim:

At any time, *either* there's a row we can modify that none of the  $\langle i, x \rangle$  pairs will be sensitive to, or else there's a whole *set* of rows we can modify so as to at least double  $Q(A)$ .

Why does this claim imply the theorem? Well,  $Q(A)$  can only double so many times.

So, suppose that for every row, there's a modification of that row that some  $\langle i, x \rangle$  is sensitive to. There are  $2^{5n}$  rows, so by a counting argument, there must be a *single*  $\langle i, x \rangle$  that's sensitive to changes to at least  $2^{3n}$  rows. Fix that  $\langle i, x \rangle$ . Then given a  $2^{3n}$ -bit string  $Y = y_1 \dots y_{2^{3n}}$ , let  $A^{(Y)}$  be the oracle obtained from  $A$  by changing the rows corresponding to  $y_j = 1$ .

Suppose without loss of generality that  $M_{i,x}(A) = 1$  and hence  $p_{i,x}(A) \geq 1$ . Let  $k = \lceil \log_2 |p_{i,x}(A)| \rceil$ . Then we'll write  $Q(A)$  as the product of two nonnegative polynomials:

$$Q(A) = R(A)S(A),$$

where

$$R(A) = 2^{2k-3} + (2^k - p_{i,x}(A))^2,$$

and  $S(A)$  is the product of all other terms in  $Q(A)$ .

Notice that

$$R(A) \leq 2^{2k-3} + (2^k - 2^{k-1})^2 = \frac{3}{8} \cdot 2^{2k}.$$

On the other hand, for all  $A^{(Y)}$  obtained from  $A$  by modifying a single row,

$$R(A^{(Y)}) \geq 2^{2k-3} + 2^{2k} = \frac{9}{8} \cdot 2^{2k} \geq 3R(A).$$

So changing one of these rows at least triples  $R(A)$ .

There are now two cases. The first is that there exists an  $A^{(Y)}$ , obtained from  $A$  by modifying a single row, such that  $S(A^{(Y)}) \geq \frac{2}{3}S(A)$ . In this case

$$\begin{aligned} Q(A^{(Y)}) &= R(A^{(Y)})S(A^{(Y)}) \\ &\geq 3R(A) \cdot \frac{2}{3}S(A) \\ &= 2Q(A) \end{aligned}$$

and we're done.

The second case is that  $S(A^{(Y)}) < \frac{2}{3}S(A)$  for all  $A^{(Y)}$  obtained from  $A$  by modifying a single row. In this case we'll appeal to a lemma of Nisan and Szegedy (the same sort of thing we use in quantum lower bounds, which ultimately relies on Chebyshev polynomials and Markov's inequality).

**Lemma 1 (Nisan-Szegedy)** *Let  $P : \{0, 1\}^N \rightarrow \mathbb{R}$  be a multilinear polynomial such that*

(i)  $\deg(P) \leq \sqrt{N}/7,$

(ii)  $P(0^N) = 1,$

(iii)  $P(X) < \frac{2}{3}$  for all  $X$  with Hamming weight 1.

*Then there exists an  $X \in \{0, 1\}^N$  such that  $|P(X)| \geq 6$ .*

It follows that there exists a string  $Y$  such that  $S(A^{(Y)}) \geq 6S(A)$ . On the other hand, we have  $R(A^{(Y)}) \geq \frac{1}{3}R(A)$  for all  $Y$  (why?). Hence

$$\begin{aligned} Q(A^{(Y)}) &= R(A^{(Y)})S(A^{(Y)}) \\ &\geq \frac{1}{3}R(A) \cdot 6S(A) \\ &= 2Q(A) \end{aligned}$$

and again we're done.

Generalizing to where the oracle tape is infinitely (not just exponentially) long, and the  $M_i$ 's can query any part of it, is sort of boring and technical, so I'll skip it.

If you've seen complexity results about PP before, what I've basically done is to combine the main insight used by Beigel, Reingold, and Spielman to show PP is closed under intersection, with the main insight used by Beigel to construct an oracle relative to which  $P^{NP} \not\subseteq PP$ .

Incidentally, we can also get an oracle relative to which PE (PTIME( $2^{O(n)}$ ), not physical education!) has linear-size circuits, and hence  $PEXP \subset P/\text{poly}$  by padding. Lance pointed out that my construction also yields a relativized world where  $P^{NP} = PEXP$ , and a world where  $\oplus P = PEXP$ . Apparently these were other open problems of his.

By Toda's theorem, my construction can't be improved to  $P^{PP}$  or even  $BP \cdot PP$ .

### 3 Nonrelativizing Lower Bounds

First, let's see why  $P^{PP}$  doesn't have circuits of size  $n^k$ , for any fixed  $k$ . Consider our truth table for  $f$ . Look at the first input,  $x_0$ , and consider the set of all circuits of size  $n^k$ . Do at least half of these circuits output 1 on  $x_0$ ? Then set  $f(x_0) = 0$ . Otherwise set  $f(x_0) = 1$ . Now move on to the next input,  $x_1$ . *Among the circuits that got  $x_0$  right*, do at least half output 1 on  $x_0$ ? Then set  $f(x_1) = 0$ . Otherwise set  $f(x_1) = 1$ . Keep going this way. With each input, we at least halve the number of circuits "still in the running." So we'll run out of circuits after maybe  $n^k \log n^k$  inputs. Furthermore, it's easy to see that the resulting  $f$  is computable in  $P^{PP}$  (why?).

To generalize to quantum circuits of size  $n^k$ , we just need to consider a uniform superposition over all quantum circuits of size  $n^k$  (suitably amplified). Note that we can compare the acceptance probabilities of two quantum computations in PP, even if we have to postselect on those computations having succeeded on  $x_0, \dots, x_{t-1}$ . Finally, to generalize to quantum advice, we replace the advice by the maximally mixed state on  $n^k$  qubits. (This is the same trick I used in my paper on quantum advice and one-way communication.) Again, each input at least halves our total success probability, but if there were an advice state that worked, then the final probability would need to be at least  $\sim 2^{-n^k}$  by the end.

Alright, that was  $P^{PP}$ . Now let's bootstrap down to PP. The key is a "Quantum Karp-Lipton Theorem" (where "quantum" modifies "theorem"):

If  $PP \subset BQP/\text{qpoly}$ , then the counting hierarchy (consisting of PP,  $PP^{PP}$ ,  $PP^{PP^{PP}}$ , etc.) collapses to QMA.

This theorem easily yields the desired result. Why? On the one hand, suppose  $PP \not\subset BQP/\text{qpoly}$ . Then we're done. On the other hand, suppose  $PP \subset BQP/\text{qpoly}$ . Then  $QMA = PP = P^{PP} = CH$ . But we already know  $P^{PP}$  doesn't have quantum circuits of size  $n^k$  with quantum advice. Hence PP doesn't either.

So, how do we prove the Quantum Karp-Lipton Theorem? Well, suppose  $PP \subset BQP/\text{qpoly}$ . Then we want to *learn* the polynomial-size quantum circuits for PP—since once we have them, we can use them recursively to decide any language in CH. This follows from the result of Fortnow and Rogers that  $PP^{BQP} = PP$ .

If you've seen this sort of nonrelativizing argument before, it won't surprise you. In  $QMA \subseteq PP$ , guess the quantum circuit for PP. But how do we verify that it works? Choose a bunch of matrices uniformly

at random over some finite field. Then simulate the LFKN interactive protocol for the permanent, using the quantum circuit in place of the prover. If the verifier accepts, then we know the circuit is correct on a  $1 - 1/\text{poly}(n)$  fraction of matrices. But now we can use the random self-reducibility of the permanent to bootstrap up to *all* matrices, and we're done.

Scaling up, we similarly have that  $\text{PEXP} \not\subseteq \text{BQP}/\text{qpoly}$ . Indeed,  $\text{PEXP}$  requires quantum circuits of *half-exponential* size, meaning size  $f(n)$  where  $f(f(n)) \approx 2^n$ . In the exponential case, though, we can use an idea of Buhrman, Fortnow, and Thierauf to show the stronger result that  $\text{QMA}_{\text{EXP}} \not\subseteq \text{BQP}/\text{qpoly}$  (where  $\text{QMA}_{\text{EXP}}$  is the exponential-time version of  $\text{QMA}$ ), and indeed  $\text{QMA}_{\text{EXP}}$  requires quantum circuits of half-exponential size.

## 4 Concluding Remarks

As I alluded to earlier, Bshouty et al. showed that given any Boolean function  $f$  with polynomial-size circuits, we can *learn* such circuits in  $\text{ZPP}^{\text{NP}^f}$ . From this it follows that  $\text{ZPP}^{\text{NP}}$  does not have circuits of size  $n^k$  for any fixed  $k$ . Can we improve this to  $\text{ZPP}_{\parallel}^{\text{NP}}$ ? I thought so, for a few weeks! But I couldn't *quite* make it work, until finally I found an oracle relative to which  $\text{ZPP}_{\parallel}^{\text{NP}}$  and even  $\text{BPP}_{\parallel}^{\text{NP}}$  have linear-sized circuits.

There's this asinine saying, "you're a liberal until you get mugged." You're an oracle pooh-pooher until your brilliant idea gets killed by an oracle.

In my oracle construction for  $\text{BPP}_{\parallel}^{\text{NP}}$ , the high-level idea is similar to the construction for  $\text{PP}$ —you define a global progress measure, and show that you can keep on increasing it. I won't go into the details here.

So, does this mean that in Bshouty et al.'s algorithm, the  $\text{NP}$  queries can't be parallelized? Well, we need to distinguish carefully between *relativizing* results and *black-box* results. In the former, the circuit we're trying to learn is an oracle circuit. In the latter, it's a normal circuit, but we can still learn about it only by querying the function  $f$  it computes.

Rather unexpectedly, I showed that if computation cost isn't an issue, then *the "query part" of Bshouty et al.'s algorithm can be parallelized (and derandomized)*. More precisely, if  $\text{P} = \text{NP}$ , then given any Boolean function  $f$  with polynomial-size circuits, we can learn a circuit for  $f$  in  $\text{P}_{\parallel}^{\text{NP}^f}$ . So my intuitions about query complexity were right on the money—all I was missing was one *tiny* computational assumption, namely that (ahem)  $\text{P} = \text{NP}$ . But despite the ridiculous-seeming strength of the hypothesis, the theorem is not trivial to prove! Like Bshouty et al.'s original result, it requires approximate counting, hashing, etc. The upshot is that we can't show black-box learning in  $\text{P}_{\parallel}^{\text{NP}}$  is impossible, without also showing that  $\text{P} \neq \text{NP}$ .

(Note that if  $\text{BP} \cdot \text{PP}$  equals  $\text{PP}$ , which it does under a plausible derandomization assumption, then  $\text{PH} \subseteq \text{PP}$  by Toda's theorem—so given any  $f$  with polynomial-size circuits, we can also learn a circuit for  $f$  in  $\text{PP}^f$ .)

All of this is closely related to the *circuit minimization problem*, which dates back to the 1950's, and was one of the main open problems in Garey and Johnson. Given a circuit, is there a circuit of some smaller size  $s$  that computes the same function? Certainly this problem is in  $\Sigma_2$ , and certainly it's  $\text{NP}$ -hard, but not much more is known. Still, Bshouty et al.'s result implies that we can *approximate* the minimum circuit size  $\text{ZPP}^{\text{NP}}$  (and thus in  $\text{P}^{\text{NP}}$ , under some derandomization assumption). So, what is the *true* complexity of this problem? Is it  $\text{P}^{\text{NP}}$ ?  $\text{P}_{\parallel}^{\text{NP}}$ ? I don't have any good intuition! My results at least show that non-relativizing ideas will be needed to make further progress.

In conclusion, to show that classes like  $\text{ZPP}_{\parallel}^{\text{NP}}$  and  $\text{MA}$  and  $\text{NP}$  don't have linear-size circuits, we'll need nonrelativizing results that kick in *without* the full arithmetization that we can do in counting classes like  $\text{PP}$ . Umesh likes to argue that we already have one such nonrelativizing result: namely, the PCP Theorem. But can that theorem be pumped for circuit lower bounds? I wouldn't be terribly surprised if, somewhere out there in Platonic heaven, a way existed. So, find it!

In any case, if it was ever tenable to dismiss oracle results as mere formalizations of the obvious, I don't think it is today. In fact, I conjecture that there's some absolute lower bound on the *product* of two difficulties: the difficulty of proving a theorem, and the difficulty of proving that theorem false relative to an

oracle. Basically, what the oracle results do is set up a battle line snaking through the Complexity Zoo—and our task, as nonrelativizing lowerboundsmen and lowerboundswomen, is to probe that line for weak points. I believe, I hope, I dream, that the oracles (being oracles) are trying to tell us something, and if only we could understand what it was, we could start slaying some of these rabbits. Thank you.