80240233: Complexity of Computation **Lecture 22**
ITCS, Tsinghua Univesity, Fall 2007 22 December 2007

Instructor: Andrej Bogdanov **Notes by:** Hongyi Yao

When we discussed lower bounds for constant depth circuits in Lecture 7, one of the main motivations was to gain some understanding of the NP versus P/poly question. Although it is widely believed that NP $\not\subseteq$ P/poly, it is not even known how to prove the much weaker statement NEXP $\not\subseteq$ P/poly. Yet we know a proof that the parity function is not computable by constant depth circuits. If we work a little bit harder, shouldn't we be able to generalize this proof to a lower bound for polynomial-size cirucits?

In the mid 1990s Razborov and Rudich gave an interesting explanation as to why the methods used to prove results like PARITY $\notin$ AC^0 are unlikely to generalize to the setting of P/poly. They looked at all known proofs of circuit and formula lower bounds and noticed that at a very high level, all these proofs follow the same general pattern. Then they showed that if a circuit lower bound for P/poly follows the same pattern, something very strange happens: A certain strong form of one-way functions cannot exist. So if we believe in strong one-way functions, which most people do, then the type of proof we used to show PARITY $\notin$ AC^0 cannot prove, say, that SAT $\notin$ P/poly.

First, we formalize the notion of "strong one-wayness".

1 Strong pseudorandom objects

In the last few lectures we showed how, starting from a one-way permutation, one can derive a pseudorandom generator, which in turn yields a pseudorandom function. In doing so, we made an implicit choice of the "hardness parameters" with respect to which these objects were defined. For one-way permutations, we asked that, for an arbitrary polynomial p, the permutation is hard to invert on $1/p(n)$ fraction of inputs for every circuit family of size $p(n)$. For pseudoranom generators, we asked that no circuit family of size $p(n)$ can distinguish the output of the generator from a random string with probability over $1/p(n)$. For pseudorandom functions, we again asked that no circuit family of size $p(n)$ with oracle access to the function can tell the function apart from random with probability over $1/p(n)$.

All these proofs worked by reduction: We fix a device that breaks the object we are reducing to and derive a device that breaks the object we are reducing from. Each time, the reduction incurs some polynomial cost: For instance, if we have a circuit of size $p(n)$ that distinguishes the output of the pseudorandom generator from random with probability over $1/p(n)$, then we derive a circuit that breaks the one-way permutation used to construct the generator on $q(n) = \mathrm{poly}(n, p(n))$ fraction of inputs.

We could have carried out the same reductions if instead of polynomial, our notion of hardness was subexponential. We will call such pseudorandom objects "strong". For instance, a family of functions $f_n : \{0,1\}^n \to \{0,1\}^n$ is *strongly one-way* if on input x, $f_n(x)$ is polynomial-time computable and there exists a constant $\delta > 0$ such that for every circuit family C of size 2^{n^δ},

$$\Pr_{x\sim\{0,1\}^n}[f(C(f(x))) = f(x)] < 2^{-n^\delta}.$$

Similarly we can define strong pseudorandom permutations, strong pseudorandom generators and strong pseudorandom functions. In the case of pseudorandom functions, something interesting happens under the strong definition: A function family[1] $F_z : \{0,1\}^{k(n)} \to \{0,1\}$, where $z \in \{0,1\}^n$, is strongly one-way if on input (x, z), $F_z(x)$ is polynomial-time computable and there exists a constant $\delta > 0$ for every circuit family C of size 2^{n^δ},

$$\left|\Pr_z[C^{F_z}(1^n) = 1] - \Pr_R[C^R(1^n) = 1]\right| < 2^{-n^\delta}.$$

Let's look at the setting of parameters $k = n^\gamma$, where γ is a constant smaller than δ. Notice that the circuit C is then large enough to query the function C at all its inputs. So we can "feed" the function F_z as an input rather than as an oracle to the circuit C: Since F_z has description size $2^k = 2^{n^\gamma}$, the size of C will be *polynomial* in the length of F_z.

We summarize the discussion in the following theorem.

Theorem 1. *Suppose that strong one-way functions exist. Then for every polynomial p there exists a constant $\gamma > 0$ and a family of functions $F_z : \{0,1\}^{n^\gamma} \to \{0,1\}$, $z \in \{0,1\}^n$ such that on input (x, z), $F_z(x)$ is polynomial-time computable and for every circuit family $C_N : \{0,1\}^N \to \{0,1\}$ of size $p(N)$ and every sufficiently large $N = 2^{n^\gamma}$,*

$$\left|\Pr_{z\sim\{0,1\}^n}[C_N(\langle F_z\rangle) = 1] - \Pr_R[C_N(\langle R\rangle) = 1]\right| < 1/p(N)$$

where $R : \{0,1\}^{n^\gamma} \to \{0,1\}$ is a random function.

In the last two lectures, we proved this theorem assuming the existence of strong one-way permutations. The assumption that strong one-way permutations exist is believed to hold.

2 Likeliness and constructivity

Now let us, revisit the proof that PARITY $\notin$ AC^0 from Lecture 7. At a very high level, the proof worked as follows: We defined a *property* of functions and argued that the parity function has this property, while no function computed by a small constant-depth circuit can have the property.

The property we looked at in the proof (due to Razborov and Smolensky) was, roughly, that "f cannot be approximated by a low-degree polynomial". More precisely, for $f : \{0,1\}^n \to \{0,1\}$, we looked at the following property:

> We say f has the HIGHDEGREE property if for every polynomial p of degree at most $\sqrt{n}$, $\Pr_{x\sim\{0,1\}^n}[p(x) = f(x)] < 0.99$.

[1] In the last lecture we looked at functions with n bits of output, but here it will be sufficient to have one bit of output; we can take the first bit and ignore the other ones.

Then, we showed that HIGHDEGREE(PARITY) $= 1$ (the parity function satisfies this property), while HIGHDEGREE(f) $= 0$ for every f computed by a circuit of depth d and size $2^{O(n^{1/4d})}$.

There is a different proof, also showing that PARITY $\notin$ AC$_0$, that we did not cover in class. Roughly speaking, this proof goes by showing that functions computed by small depth circuits are not "sensitive" to changes in their inputs – if the input to the function is flipped, the output is unlikely to change – while the parity function *is* sensitive. There are several ways to formalize this notion, and we choose one that is convenient for our discussion.

> We say $f(x_1, \ldots, x_n)$ has the SENSITIVE property if for every $S \subseteq \{1, \ldots, n\}$ of size $2\log_2(n)$ and every partial assignment of the inputs $x_i, i \notin S$, f is not a constant function over the remaining inputs $x_j, j \in S$.

It is easy to see that SENSITIVE(PARITY) $= 1$. On the other hand, Furst, Saxe, and Sipser, then Yao, and finally Håstad showed (for different parameters) that functions f computed by not too large circuits of fixed depth satisfy SENSITIVE(f) $= 0$.

Razborov and Rudich looked at these and other proofs of boolean circuit lower bounds (for sufficiently explicit functions) and saw that, at a very high level, all these proofs look like the above ones: They single out a property P of functions and show that the function to which the lower bound applies has this property, while functions computed by the circuits under consideration don't have it. Moreover, they realized that the properties used in all these proofs were special in two particular ways; they satisfy two common features which we will call "likeliness" and "constructivity". We look into these two features next, looking into the SENSITIVE property as a guiding example.

Likeliness To motivate the "likeliness" feature, let us recall a basic fact from Lecture 3: A random function is unlikely to be computable by a small circuit. In that lecture we proved explicit bounds on the size of circuits needed to compute every function, but in essence the argument shows that for any class of not too large circuits, the probability that a random function can be computed by a circuit from the class is small.

Now suppose that we used some property P to prove a circuit lower bound for some class of circuits, like AC^0 or P/poly. We have then showed that $P(f) = 1$ for some explicit function f, like SAT or PARITY. But since we know that random functions are also hard to compute by circuits in the class, we might expect that $P(R) = 1$ with good probability for a random function R as well. This reasoning is not sound, but it turns out that all known sufficiently "explicit" properties used in circuit lower bounds have this feature. We call such properties "likely".

Definition 2. *A property P is* likely *if* $\Pr_R[P(\langle R\rangle) = 1] > 1/3$ *for sufficiently large n, where R is a random function from* $\{0,1\}^n$ *to* $\{0,1\}$.

For instance SENSITIVE is a likely property because for any one choice of $\binom{n}{2\log_2 n}$ variables to be fixed and any one of $2^{n-2\log_2 n}$ fixings of these variables, the probability that a random function is constant on the other $2\log_2 n$ variables is $2 \cdot 2^{-2^{2\log_2(n)}}$, so by a union bound:

$$\Pr[\text{SENSITIVE}(R) = 0] \leq \binom{n}{2\log_2 n} \cdot 2^{n-2\log_2 n} \cdot 2 \cdot 2^{-2^{2\log_2(n)}} \leq 2^{2n} \cdot 2^{-n^2} < 1/3.$$

Constructivity The "constructivity" feature says that the property of a function is efficiently computable when the truth table $\langle f \rangle$ is given as an input. Formally,

Definition 3. *A property P is* constructive *if $P(\langle f \rangle)$ is computable in time $2^{O(n)}$, where f is a function from $\{0,1\}^n$ to $\{0,1\}$.*

Notice the truth table of f is a string of length 2^n, so time $2^{O(n)}$ is time polynomial in the description length of f.

To explain constructivity, we want to think of a candidate circuit lower bound proof – or any other proof we are interested in for that matter – as an algorithm in disguise. This "duality" between proofs and algorithms can be made formal, but it is easier to look at an example.

Since we are short on explicit lower bound proofs in computer science let's look at an example from mathematics – for instance the proof that "$\sqrt{2}$ is irrational". This proof goes by trying to write $\sqrt{2}$ as a rational number a/b, from where one derives $a^2 = 2b^2$ where a and b are integers. But this is impossible because if we factor both sides, 2 appears an even number of times on the left and an odd number of times on the right.

The same argument easily generalizes to tell us that $\sqrt{n}$ is irrational whenever n is not a perfect square. Now let us think of "n is not a perfect square" as a property $P(n)$ of the number n. What is the algorithmic complexity of computing $P(n)$? We can do this say by running the square root algorithm from grade school whose running time is poly $\log n$, which is polynomial in the description length of n. Therefore, the proof that $\sqrt{2}$ is irrational relies on a *constructive* property of numbers.

Many known "explicit impossibility proofs" in mathematics and computer science rely on constructive properties, although there are possible exceptions. (It is impossible to know for sure that a proof "relies on a nonconstructive property", since identifying the property in question is somewhat of an art, and sometimes there are several candidates. We will see an example of this shortly.) In particular, Razborov and Rudich went through the circuit lower bound proofs known at the time and showed that they were all essentially constructive.

To give an example of a nonconstructive property, let us look at the proof that there exist functions not computable by polynomial size circuits. This proof can also be viewed as an algorithm for obtaining such a function: Choose one at random. This algorithm is not efficient, as the proof does not give an efficient way of choosing this function; but the function to which the lower bound applies (a random function) is not explicit either in any reasonable sense.

The SENSITIVE property is constructive: To compute if f is sensitive, try all possible subsets S of size $2\log_2 n$, fix all possible assignments outside S, and try all assignments inside S to see if the function is constant. This algorithm runs in time

$$\binom{n}{2\log_2 n} \cdot 2^{n-2\log_2 n} \cdot 2^{2\log_2 n} = 2^{O(n)}.$$

3 Natural proofs

We call a property of boolean functions *natural* if it is both likely and constructive. Thus SENSITIVE is a natural property. The next theorem shows that natural properties are unlikely to prove any circuit lower bounds for P/poly:

Theorem 4. *If there exist a natural property P such that for all $f \in \mathrm{P/poly}$, $P(f_n) = 0$ for sufficiently large n, then strong one-way functions do not exist.*

Proof. Since P is likely, we have that $\Pr[P(\langle R\rangle) = 1] \geq 1/3$ for a random function R. Now let $F_z : \{0,1\}^{n^\gamma} \to \{0,1\}$, $z \in \{0,1\}^n$ be an arbitrary function family computable by polynomial-size circuits. Then $P(\langle F_z\rangle) = 0$ for all z and sufficiently large n. In particular, we have that

$$\Pr_R[P(\langle R\rangle) = 1] - \Pr_z[P(\langle F_z\rangle) = 1] \geq 1/3.$$

where $R, F_z : \{0,1\}^{n^\gamma} \to \{0,1\}$. Since P is constructive, $P(\langle f\rangle)$ is computable by circuits of size $2^{O(n)}$. By Theorem 1, strong one-way fucntions cannot exist. □

We showed that the SENSITIVE property is natural. But how about the HIGHDEGREE property? It is not hard to see that HIGHDEGREE is likely, but it is not clear at all that it should be constructive. However, let us dig more deeply into the proof from Lecture 7 that PARITY $\notin \mathrm{AC}^0$. To argue that HIGHDEGREE(PARITY) = 1, we argued by contradiction: Suppose that there is a set A of size $0.99 \cdot 2^n$ on which PARITY can be written as a polynomial of degree $\sqrt{n}$. Then we argued that any function $f : A \to \mathbb{R}$ can be represented as $\tilde{f} = p_0 + \widetilde{\mathrm{PARITY}} \cdot p_1$, where p_0, p_1 are polynomials of degree at most $n/2$, and used this to reach a contradiction (since there are more such polynomial than this representation allows).

So we could argue that the proof from Lecture 7 in fact relies on the following property:

> f has the HIGHDEGREE$'$ property if every $g : \mathbb{R}^n \to \mathbb{R}$ can be written in the form $g = p_0 + \tilde{f} \cdot p_1$, where $p_0, p_1 : \mathbb{R}^n \to \mathbb{R}$ are polynomials of degree at most $n/2$.

This property is now constructive, as one can compute HIGHDEGREE$'(f)$ by linear algebra: Look at the linear space of all the polynomials $p_0 + \tilde{f} \cdot p_1$, where p_0, p_1 are "indeterminate" polynomials of degree at most $n/2$, and check that the dimension of this space is 2^n. This can be done using Gaussian elimination in time $2^{O(n)}$.

However, it is not clear anymore that HIGHDEGREE$'(f)$ is likely (it might be the case, it just does not appear easy to prove). However, without changing the spirit of the proof, we could have used the following variant HIGHDEGREE$''$ of the same property:

> f has the HIGHDEGREE$''$ property if the linear space of functions of the form $p_0 + \tilde{f} \cdot p_1$, where $p_0, p_1 : \mathbb{R}^n \to \mathbb{R}$, has dimension at least $3/4 \cdot 2^n$.

Now HIGHDEGREE$''$ is constructive by the same argument as for HIGHDEGREE$'$, but it turns out that it is also likely, although we will not prove so here.

To argue that the proof from Lecture 7 relies on a natural property, we had to change the proof a little bit. If we go back to the proof, we can see that the change necessary to make the proof rely on property HIGHDEGREE$''$ instead of property HIGHDEGREE$'$ is very minor. So even though the proof by itself may not be natural, it "naturalizes" in a very simple way.