80240233: Complexity of Computation
ITCS, Tsinghua Univesity, Fall 2007

Lecture 17
4 December 2007

Instructor: Andrej Bogdanov

Notes by: Hongyi Yao

In the last lecture we showed that if E has "worst-case hard" problems then it also has "slightly average-case hard" problems. More precisely, if there is some $L \in$ E such that $L \notin$ BPP, then there is $L' \in$ E such that every randomized polynomial-time algorithm fails to solve L' on at least $O(1/n)$ fraction of inputs.

Today we show that from L' we can obtain a new problem $L'' \in$ E that is hard to solve even on a $1/2 - \epsilon$ fraction of inputs for every $\epsilon > 0$.[1] However we work with a different notion of hardness: Instead of hardness against randomized algorihtms, we will consider hardness against (polynomial-size) circuit families.

The approach we describe is based on Yao's XOR lemma (1982). Many different proofs of this result are known, and here we present a proof by Impagliazzo (1995).

1 The XOR Lemma

First, we give some definitions for average case hardness against circuits.

Average Case Hardness We say $f : \{0,1\}^n \to \{0,1\}$ is δ-hard for size S, if for any circuit of size S, $\Pr_{x\sim\{0,1\}^n}[f(x) = C(x)] < 1 - \delta$.

Suppose we have a problem f is somewhat hard, so how can we construct a very hard function from f? Intuitively, we want to ask one to solve many independent instances of f. But how do we combine these different instances to obtain a single decision problem? Here we will XOR the answers to the different instances. Intuitively, since the XOR function depends on the values of all its variables, in order to "know" the answer to the combined instance we will have to solve every individual instance separately.

Given $f : \{0,1\}^n \to \{0,1\}$, we write

$$f^k(x_1, x_2, ..., x_k) = f(x_1) + \cdots + f(x_k)$$

where '+' indicates summation mod 2.

Here we will prove the following theorem:

Theorem 1. *If $f : \{0,1\}^n \to \{0,1\}$ is δ-hard for circuits of size S, then f^k is $1/2 - \epsilon - (1-\delta)^k$ hard for circuits of size $O(S\epsilon^2\delta^2)$.*

[1] This level of hardness is still insufficient for the Nisan-Wigderson generator, but the approach we describe can be extended to that setting using additional ideas.

If we start from a decision problem $L \in \mathrm{E}$ that is $O(1/n)$-hard for polynomial-size circuits on every (sufficiently large) input length and apply the above transformation on every input length of L with $k = O(n \log \epsilon)$, we obtain a problem $L' \in \mathrm{E}$ that is $1/2 - \epsilon$ hard on every input length of the form kn. By padding we can make L' $1/2 - \epsilon$ hard on all input lengths.

Before we do the proof let's see why the theorem makes sense. Suppose C is the "best" circuit of size S for f, so we have

$$\mathrm{Pr}_{x \sim \{0,1\}^n}[f(x) = C(x)] = 1 - \delta.$$

We can try to solve $f^k(x_1, \dots, x_k)$ by running the circuit C on $x_1, \dots, x_k$ and xoring the answers together. This is not the only way to attack f^k, but it seems like a reasonable attempt.

One way to analyze this process is to think as follows: When I choose a random $x \sim \{0,1\}^n$, with probability $1 - 2\delta$ it will happen that $C(x) = f(x)$, and with probability 2δ the value $C(x)$ will be completely independent from $f(x)$. More formally, we can define $R \subseteq \{0,1\}^n$ to be some set containing all x such that $C(x) \neq f(x)$ and as many x (chosen arbitrarily) such that $C(x) = f(x)$. Then $\mathrm{Pr}_{x \sim \{0,1\}^n}[x \in R] = 2\delta$ and conditioned on $x \in R$, $\mathrm{Pr}[C(x) = f(x)] = 1/2$.

Now we have:

$$\begin{aligned}\mathrm{Pr}[C(x_1) + \cdots + C(x_k) = f^k(x_1, \dots, x_k)] &= \mathrm{Pr}[\ \star \mid \exists x_i \in R] \cdot \mathrm{Pr}[\exists x_i \in R] \\ &\quad + \mathrm{Pr}[\ \star \mid \forall x_i : x_i \notin R] \cdot \mathrm{Pr}[\forall x_i : x_i \notin R] \\ &= \frac{1}{2} \cdot [1 - (1 - 2\delta)^k] + 1 \cdot (1 - 2\delta)^k \\ &= \frac{1}{2}(1 + (1 - 2\delta)^k)\end{aligned}$$

where $\star$ is the event $C(x_1) + \cdots + C(x_k) = f^k(x_1, \dots, x_k)$. This is because conditioned on some x_i falling inside R, the answer $C(x_i)$ will be independent from $f(x_i)$, so regardless of the other x_js $C(x_i)$ will hit the correct value with probability exactly $1/2$.

This intuition suggests that f^k is indeed very hard. However to turn it into a proof we have to argue that *any* small circuit on input $x_1, \dots, x_k$ fails to compute $f^k(x_1, \dots, x_k)$ on a substantial fraction of inputs. Above we considered only circuits that treat each input x_i independently and then xor the answers.

To reformulate our assumption, we know that for every circuit C of size S, it must be the case that

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

Trying to imitate the above argument, we start by the simple observation that

> For every circuit C of size S, there exists a set $R \subseteq \{0,1\}^n$ of size $2\delta \cdot 2^n$ such that $\mathrm{Pr}_{x \sim R}[C(x) = f(x)] = 1/2$.

The *hardcore lemma* of Impagliazzo, which we state without proof, shows that we can switch the quantifiers essentially without loss of generality:

> There exists a set $H \subseteq \{0,1\}^n$ of size $\delta \cdot 2^n$ such that for every circuit of size $S \cdot \delta^2 \epsilon^2/4$, $\mathrm{Pr}_{x \sim H}[C(x) = f(x)] = 1/2 + \epsilon$.

We call such a set H an ϵ *hardcore set* for size $S' = S \cdot \delta^2\epsilon^2/4$. Using the hardcore lemma, to prove Theorem 1 it is sufficient to show that:

Lemma 2. *If $f : \{0,1\}^n \to \{0,1\}$ has an ϵ hardcore set for size S' of size $\delta 2^n$, then f^k is $1/2 - \epsilon - (1-\delta)^k$ hard for size S'.*

Proof. We will argue the contrapositive; assuming that f^k is not $1/2 - \epsilon - (1-\delta)^k$ hard for size S', we will show that for everty set H of size $\delta 2^n$, H is not ε-hardcore for size S'.

Let C be a circuit of size S' such that

$$Pr[C(x_1, x_2, ..., x_k) = f^k(x_1, x_2, ..., x_k)] \geq \frac{1}{2} + \epsilon + (1-\delta)^k.$$

and H be a candidate hard-core set of size $\delta \cdot 2^n$.

We define the sets $S_0, \ldots, S_k \subseteq \{0,1\}^{nk}$, where

$$S_t = \{\mathbf{x} = (x_1, \ldots, x_k) : \text{exactly } t \text{ of } x_1, \ldots, x_k \text{ are in } H\}.$$

Then we have:

$$\begin{aligned}
\Pr_{\mathbf{x} \sim \{0,1\}^{nk}}[C(\mathbf{x}) = f^k(\mathbf{x})] &= \Pr[C(\mathbf{x}) = f^k(\mathbf{x}) \mid \mathbf{x} \in S_0] \cdot \Pr[\mathbf{x} \in S_0] \\
&\quad + \Pr[C(\mathbf{x}) = f^k(\mathbf{x}) \mid \mathbf{x} \notin S_0] \cdot \Pr[\mathbf{x} \notin S_0] \\
&\leq (1-\delta)^k + \Pr[C(\mathbf{x}) = f^k(\mathbf{x}) \mid \mathbf{x} \notin S_0].
\end{aligned}$$

So it must be that

$$\Pr[C(\mathbf{x}) = f^k(\mathbf{x}) \mid \mathbf{x} \notin S_0] \geq \frac{1}{2} + \epsilon.$$

Since the left hand side is a weighted average over the sets $S_1, \ldots, S_k$, it must be that for some $t \neq 0$:

$$\Pr[C(\mathbf{x}) = f^k(\mathbf{x}) \mid \mathbf{x} \in S_t] \geq \frac{1}{2} + \epsilon.$$

Now we will construct a random circuit C' with "partial knowledge" of f will solve f on the set H on $1/2 + \epsilon$ fraction of instances: on input $x \in \{0,1\}^n$,

```
C': On input $x \in \{0,1\}^n$:
    Set $a_1 = x$.
    Choose $a_2, \ldots, a_t$ independently at random from $H$.
    Choose $a_{t+1}, \ldots, a_k$ independently at random from $\{0,1\}^n - H$.
    Choose a random permutation $\pi : [k] \to [k]$.
    Output $C(a_{\pi(1)}, \ldots, a_{\pi(k)}) + f(a_2) + \cdots + f(a_k)$
```

When x is randomly chosen from H, then the string $(a_{\pi(1)}, \ldots, a_{\pi(k)}) \in \{0,1\}^{nk}$ looks exactly like a random string in S_t. So it must be the case that

$$\Pr[C(a_{\pi(1)}, \ldots, a_{\pi(k)}) = f(a_1) + \cdots + f(a_k)] \geq \frac{1}{2} + \epsilon.$$

Since $a_1 = x$,

$$\Pr[C'(x) = f(x)] = \Pr[C(a_{\pi(1)}, \ldots, x, \ldots, a_{\pi(k)}) + f(a_2) + \cdots + f(a_k) = f(x)] \geq 1/2 + \epsilon.$$

To remove the randomness and the need for "partial knowlege" of f, observe that there must be a specific choice of $a_2, \ldots, a_k$ and π which maximizes the probability that $C'(x) = f(x)$. If we fix this choice, the values $f(a_2), \ldots, f(a_k)$ are fixed as well, and we can hardwire these into the circuit C'. Notice that C' has exactly the same size as C (since not gates are not counted towards the size of a circuit.) □