6.841 - Advanced Complexity Theory

Lecture Notes: 2/24/2000.

Today we introduce the notion of randomized computation. The main complexity class of interest is called BPP (Bounded-error Probabilistic Polynomial time). We will present two definitions for BPP.

Def. $L \in$BPP if $\exists M$, a randomized polynomial time TM such that
If $x \in L$ then $Pr[M \text{ accepts } x] > 2/3$ and
if $x \notin L$ then $Pr[M \text{ accepts } x] < 1/3$.

As we will see later, the constants 2/3 and 1/3 are arbitrary. Really, all we need for an adequate definition are any two values that are separated by a reasonable amount.

The second definition is more formal, and can be more easily adapted as an operator that can be used on complexity classes other than P.

Def. $L \in$BPP if $\exists A \in$P and a function $f(n) = n^{O(1)}$ such that
If $x \in L$ then $Pr_{r:|r|=f(|x|)}[(r, x) \in A] > 2/3$, and
if $x \notin L$ then $Pr_{r:|r|=f(|x|)}[(r, x) \in A] < 1/3$.

This introduces the notion of a random input. The way this type of machine works is that it takes two inputs, one of which is expected to be a random string of some certain length relative to the input size.

Another reasonable probabilistic computation definition is the following definition of RP (Randomized Polynomial time), which is like BPP except that it is allowed to be incorrect only when its input is in the language.

Def. $L \in$RP if $\exists M$, a randomized polynomial time TM such that
If $x \in L$ then $Pr[M \text{ accepts } x] > 2/3$ and
if $x \notin L$ then M always rejects x.

The current state of our knowledge about randomized computation is as follows. First of all, P is clearly contained in any of the classes: BPP, RP, or co-RP. RP and co-RP are both contained in BPP.

RP is contained in NP (if M is the RP machine for a language L, then we can create an NP machine M' which uses existential states to generate the random choices to be used by M, and then it simulates M. If M accepts on any set of random choices, then the input must be in L, since M never accepts inputs not in L), and therefore co-RP is contained in co-NP.

Finally, BPP $\subset$ P/poly, and BPP $\subset \Sigma_2^P \cap \Pi_2^P$. These are the two main results we will prove in this lecture.

First, however, we will at least state the amplification theorem, which formalizes the notion that the actual constants 2/3 and 1/3 in the definitions above are arbitrary. The key to analysis of probabilistic computation are the Chernoff bounds. The statement of these bounds is as follows:

Let $y_1, \ldots, y_n$ be identically distributed, independent random variables chosen over the range $[0, 1]$. Let

$$Y = \sum_{i=1}^{n} y_i$$

be the sum of the variables and let $\mu = E(Y)$ be the expected value of this sum. Chernoff bounds allow us to limit the probability of variation from the expectation:

$$Pr[Y > (1 + \epsilon)\mu] < e^{-\mu\epsilon^2/3},$$

$$Pr[Y < (1 - \epsilon)\mu] < e^{-\mu\epsilon^2/3}.$$

The amplification theorem statement is as follows:

Theorem. If L is a language such that $\exists a > 0, M$, a randomized polynomial-time TM, such that
If $x \in L$ then $Pr[M \text{ accepts } x] > c(|x|)$,
if $x \notin L$ then $Pr[M \text{ accepts } x] < s(|x|)$, and
$c(n) - s(n) > n^{-a}$. Then $\forall b\ \exists M'$ a randomized polynomial-time TM such that

If $x \in L$ then $Pr[M' \text{ accepts } x] > 1 - 2^{-|x|^b}$ and
if $x \notin L$ then $Pr[M' \text{ accepts } x] < 2^{-|x|^b}$.

The following is a sketch of the proof of this theorem.

Let M' be a randomized polynomial-time TM that simulates M on input w a total of $k = 12|w|^{3a+b}$ times, and accepts if M accepts at least

$$\left(\frac{c(|w|) + s(|w|)}{2}\right) k$$

times. Chernoff's bound verifies that this works.

Now we will take on the first of our two primary results for this lecture: BPP $\subset$ P/Poly.

Theorem. BPP $\subset$ P/Poly.

Proof. If $L \in$BPP then $\exists A \in$P and a function $f(n) = n^{O(1)}$ such that
if $x \in L$ then $Pr_{r:|r|=f(|x|)}[(r,x) \in A] > 1 - 2^{-|x|-1}$ and
if $x \notin L$ then $Pr_{r:|r|=f(|x|)}[(r,x) \in A] < 2^{-|x|-1}$.

Consider words of length n. We want to show that

$$p = Pr_{r:|r|=f(n)}[\forall_{x:|x|=n} x \in L \Leftrightarrow (r,x) \in A]$$

is greater than zero. If we can show that $p > 0$, then there must be some r such that for all x of length n, r is a successful random string for A to use in computing x. Then, we can show that L is in P/poly by using the sequence of these r for each n as the advice strings.

So, without further ado,

$$p \geq 1 - \sum_{x \in L, |x|=n} Pr_r[(r,x) \notin A] - \sum_{x \notin L, |x|=n} Pr_r[(r,x) \in A].$$

Now we apply the criteria mentioned above.

$$p \geq 1 - \sum_{x \in L, |x|=n} 2^{-n-1} - \sum_{x \notin L, |x|=n} 2^{-n-1} =$$

$$1 - \sum_{x:|x|=n} 2^{-n-1} =$$

$$1 - 2^n 2^{-n-1} =$$

$$1/2 > 0.$$

This completes the proof. BPP $\subset$ P/Poly.

Corollary. if NP $\subset$ BPP then PH collapses (by the Karp-Lipton theorem).

Next, we will prove that BPP $\subset \Sigma_2^P$. The proof is due originally to Sipser; this is an improvement due to Lautemann. Throughout, $n = |w|$.

Theorem. BPP $\subset \Sigma_2^P$.

Proof. Let L be a language in BPP. We know $\exists A \in$P and a function $f(n) = n^{O(1)}$ such that
If $w \in L$ then $Pr_{r:|r|=f(n)}[(r,w) \in A] > 1 - 2^{-n}$, and
if $w \notin L$ then $Pr_{r:|r|=f(n)}[(r,w) \in A] < 2^{-n}$.

Def. Define $R_w = \{r : |r| = f(n) \text{ such that } (r,w) \in A\}$. Correspondingly,
If $w \in L$ then $|R_w| > (1 - 2^{-n})2^{f(n)}$ and
if $w \notin L$ then $|R_w| < 2^{-n}2^{f(n)}$.

The idea here is to take a number of translations of R_w, and see if they cover the entire space $\{0,1\}^{f(n)}$. Each translation of R_w has the same size as R_w, and if R_w is most of the space (ie, $w \in L$) then this collection of translations would be likely to cover the space. However, if R_w is very small (ie, $w \notin L$) then this collection could never cover the space. More formally,

Def. (translation) Let $S \subset \{0,1\}^{f(n)}$. For $t \in \{0,1\}^{f(n)}$ let the translation $S \oplus t$ be defined as

$$\{x : x \oplus t \in S\}$$

where $x \oplus t$ is defined as the XOR of the two strings (or, the bitwise sum modulo 2).

Claim. (1) If $|S| > (1 - 2^{-n})2^{f(n)}$ then $\exists \tau = \{t_1, \dots t_{f(n)}\}$ such that

$$\cup_{i=1}^{f(n)} (S \oplus t_i) = \{0,1\}^{f(n)}$$

(2) If $|S| < 2^{-n}2^{f(n)}$ then $\forall \tau = \{t_1, \dots t_{f(n)}\}$,

$$\cup_{i=1}^{f(n)} (S \oplus t_i) \neq \{0,1\}^{f(n)}.$$

First, we show that the claim proves the theorem. If the claim is true, we can design a Σ_2^P machine M to solve L as follows:

1. Use $\exists$ states to generate τ.
2. Use $\forall$ states to generate $r \in \{0,1\}^{f(n)}$.
3. Check if $r \in \cup_i R_x \oplus t_i$ and accept if so, otherwise, reject.

This is polynomial time, since we can check whether $r \in R_x$ in polynomial time, and $f(n)$ is polynomial in n. By the claim, if $x \in L$ then on any correct τ we accept. If $x \notin L$ we reject, since there is no such τ. Therefore, we only have to prove the claim.

First, we prove part (2) of the claim. If $|S| < 2^{-n}2^{f(n)}$ then

$$|\cup_i (S \oplus t_i)| \leq f(n)2^{-n}2^{f(n)}.$$

Since $f(n) = n^{O(1)}$, $f(n)2^n < 1$ for sufficiently large n. Therefore, this union doesn't cover $\{0,1\}^{f(n)}$.

Note that the "sufficiently large n" clause here doesn't cause a problem. If we take this into account, we need only hard-code the correct answer for all words smaller than this bound into our Σ_2^P machine.

Next, we prove part (1). Let

$$p = Pr_\tau[\forall_{|r|=f(n)} r \in \cup_{i=1}^{f(n)} (t_i \oplus S)] =$$

$$1 - Pr_\tau[\exists_{|r|=f(n)} r \notin \cup_{i=1}^{f(n)} (t_i \oplus S)] \geq$$

$$1 - \Sigma_{r:|r|=f(n)} Pr_\tau[r \notin \cup_{i=1}^{f(n)} (t_i \oplus S)].$$

We choose the t_i's independently, so we can consider them independently. Therefore,

$$p \geq 1 - \Sigma_{r:|r|=f(n)} \Pi_{i=1}^{f(n)} Pr_{t_i}[r \notin t_i \oplus S)] =$$

$$1 - \Sigma_{r:|r|=f(n)} \Pi_{i=1}^{f(n)} 2^{-n}$$

since $r \in t_i \oplus S$ if and only if $t_i \in r \oplus S$,

$$= 1 - 2^{f(n)}(2^{-n})^{f(n)} = 1 - 2^{-f(n)(n-1)} > 0.$$

Since this probability is nonzero, there must be at least some τ for which the union of the translations determined by τ covers $\{0,1\}^{f(n)}$. This completes the proof.