6.841 Advanced Complexity Theory	March 9, 2005
Lecture 11	
Lecturer: Madhu Sudan	*Scribe: Pavlo Pylyavskyy*

1 Overview

In this lecture we prove three properties of BPP. In particular, we show equivalance of weak and strong definitions of BPP; we show that any BPP algorithm can be simulated with a circuit; we show that BPP lies in the polynomial hierarchy.

2 Amplification

Recall: we say that M accepts promise problem $L = L_{yes} \cup L_{no}$ with completeness c and soundness s if

$$\forall x \in L_{yes}\ Pr_r[M(x,r) = 1] \geq c(|x|),$$
$$\forall x \in L_{no}\ Pr_r[M(x,r) = 1] \leq s(|x|).$$

We can consider two definitions of BPP. $L \in (weak)BPP$ if $\exists M$, polynomial p, functions c and s s.t. M accepts L with completeness c, soundness s and $c(n) \geq s(n) + 1/p(n)$, where $n = |x|$. $L \in (strong)BPP$ if for any polynomial q there exists M s.t. M accepts L with completeness $1 - 2^{-q(n)}$ and soundness $2^{-q(n)}$.

Theorem 1 $(weak)BPP = (strong)BPP$

Proof Let $L \in (weak)BPP$, we want to show that $L \in (strong)BPP$. Let M be $(weak)BPP$ algorithm for L, it comes with certain p, c, s. We are given a polynomial q. Design M' as follows.

Pick t large enough, as we see later $t = \Theta(p(n)^2 q(n))$ works. Pick $r_1, \ldots, r_t$ randomly and independently. Run $M(x, r_1), \ldots, M(x, r_t)$. If number of accepts is $> \frac{c(n)+s(n)}{2} t$ - accept, otherwise - reject.

Then M' places L in $(strong)BPP$. Indeed, let us see how large t needs to be for that, it will appear that polynomial size is sufficient.

The following statement is called Chernoff Bound. Let $Y_1, \ldots, Y_t \in [0,1]$ be identically distributed independent random variables with $E[Y_i] = \mu$. Then $\forall \lambda\ Pr[|\Sigma Y_i - \mu t| > \lambda\sqrt{t}] \leq exp(-\lambda^2)$, where for our purposes constant we use in exponent is not importent. In our case, let X_i be the indicator variable: $X_i = 1$ if $M(x, r_i) = 1$ and 0 otherwise. Suppose $x \in L_{yes}$, and thus $E[X_i] \geq c(n)$. Then

$$Pr[\Sigma X_i < \frac{c(n)+s(n)}{2} t] \leq Pr[|\Sigma X_i - c(n)t| \leq \frac{c(n)-s(n)}{2} t] \leq \exp(-(\frac{c(n)-s(n)}{2})^2 t) \leq \exp(-\frac{t}{4p(n)^2}).$$

So we pick $t = \Theta(p(n)^2 q(n))$ which gives us the error of the size required in definition of strong version of BPP. ■

Note, we used the following intuitive fact. if for two promise problems L_1 and L_2 we have $L_{1yes} \subset Ł_{2yes}$ and $L_{1no} \subset Ł_{2no}$, then solving L_2 is as good as solving L_1. In particular $L_2 \in P$ implies $L_1 \in P$. If C_1 and C_2 are complexity classes of promise problems, then $C_1 \subset C_2$ is equivalent to saying that for any $L_1 \in C_1$ there exists $L_2 \in C_2$ s.t. the properties above hold. This allows us to talk about promise problems in P.

3 $BPP \subset P|_{poly}$

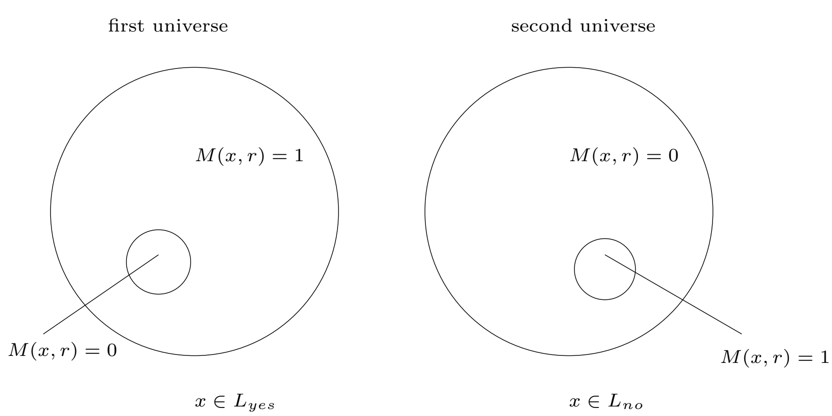
The following result is a simple corollary of the above, due to Adelman. It states that $BPP \subset P|_{poly}$. The idea is that if we pick $q(n)$ large enough than there exists a random string which is "good" for all $x \in \{0,1\}^n$. Then we use this string as advise. Here is same in more detail.

Pick $q(n) = n + 1$, M - BPP algorithm for L. Call a random string r *bad* for (M, x) if $x \in L_{yes}$ and $M(x,r) = 0$ or $x \in L_{no}$ and $M(x,r) = 1$.Then $Pr_r[r \text{ bad for } (M,x)] \leq 2^{-n-1}$. Similarly, call r bad for (M, n) if $\exists x$ s.t. $|x| = n$ and r is bad for (M, x). Then $Pr[r$ is bad for $(M,n)] \leq 2^n 2^{-n-1} = 1/2$. Then there exists r which is not bad. Then if we run machine M using as an advise $r_i, i = 1, \ldots, n$ such that r_i is not bad for (M, r), we get needed $P|_{poly}$ algorithm.

4 $BPP \subset PH$

We are going to show that BPP lies in the polynomial hierarchy. In particular, we show $BPP \subset \Sigma_2^P$. In other words, there is a way for two players to exchange sertain arguments in order to convince observer in the particular result of the probabilistic algorithm.

Let us take algorithm M which makes error at most $1/m^2$, where m is the size of random string we use. Since we know we can make error exponentially small, this is a reasonable assumption. Than depending on whether $x \in L_{yes}$ or $x \in L_{no}$ we are potentially in one of the two universes shown on the picture. In the first one r-s for which $M(x,r) = 0$ constitute the small dot inside the universe, in the second case - r-s for which $M(x,r) = 1$. Call such a dot subset in either case BAD. We are interested in permutations $\pi : \{0,1\}^m \longrightarrow \{0,1\}^m$ such that $\pi(BAD) \cap BAD = \emptyset$. We know we are in the first universe iff for any string y either y or $\pi(y)$ are good, that is $M(x,r) = 1$. Indeed, if we are in the first universe, it is so by choice of π. If we are in the second universe, there would be just not enough good strings to cover the whole image with set of good strings taken twice: $1/m^2 < 1/2$.



If we were able to find such π, the following dialog would give us a solution. First player tells partner π. Second player returns a string y. The observers can now varify whether at least one of $M(x,y) = 1$ and $M(x,\pi(y)) = 1$ holds. If it is so, then second player failed to find counterexample and observers are convinced that we are in the first universe, that is $x \in L_{yes}$. If it is not so, first player loses and observers are convinced that we are in the second universe, that is $x \in L_{no}$.

However, we do not know how to find such π. Instead, we are going to find a family of permutations $\pi_1, \ldots, \pi_l$ satisfying the following property: for any y there exists i such that $\pi_i(y) \notin BAD$. The dialog would look now as follows. First player tells his partner the π_i-s. Second player returns certain string y. After that the observers varify if at least one of $\pi_i(y) = 1$ holds.

In order for the first player not to be able to cheat, we need union of images $\pi_i(BAD)$ not to cover the whole universe.This is achieved in particular if $l/m^2 < 1$ according to the our initial bound on size of BAD. In other words, if $x \in L_{no}$ and $l < m^2$, then

$$Pr[\exists i : \ M(x, \pi_i(y)) = 1] \leq l/m^2 < 1.$$

To construct such permutations, pick y_i at random in $\{0,1\}^m$. Let $\pi_i(r) = r \oplus y_i$. Then if $x \in L_{yes}$, the following holds:

$$Pr_{y_1,\ldots,y_l}[\exists y : \forall i \ M(x, y \oplus y_i) = 0] \leq 2^{-l} 2^m.$$

For $l > m$, which we can make sure to hold, this implies that there is at least one y for which not all conditions hold, and thus $M(x, \pi_i(y)) = 1$ for some i. This shows that constructed permutations indeed give us a solution.