18.405J/6.841J Advanced Complexity Theory	February 27, 1997

Lecture 7

Lecturer: Daniel A. Spielman — *Scribe: Mark Skandera*

1 Probabilistic Complexity Classes

Imagine that a turing machine can flip coins to decide whether or not a string x is in language L. Naturally, we want the machine to decide correctly most of the time, perhaps with probability at least 2/3. To describe the different conceptual behaviors of such algorithms, we define several probabilistic complexity classes. Surprisingly, we may define these classes with or without explicitly using probability.

Definition 1 *BPP is the class of all languages L for which there is a nondeterministic polynomial time TM, M, whose computation branches all have the same length, and which has the following properties.*
If $x \in L$ then at least 2/3 of the branches accept.
If $x \notin L$ then at most 1/3 of the branches accept.

Alternatively, using probability, we could say that

When $x \in L$ then $Pr[M(x) \text{ accepts }] \geq 2/3$
When $x \notin L$ then $Pr[M(x) \text{ accepts }] < 1/3$

Note that by both definitions, M is guaranteed to halt in polynomial time, with all branches halting together.

For the many probabilistic algorithms which have only a one-sided error, we define another class.

Definition 2 *RP is the class of all languages L for which there is a nondeterministic polynomial time TM, M, whose computation branches all have the same length, and which has the following properties.*
When $x \in L$ then $Pr[M(x) \text{ accepts }] \geq 2/3$
When $x \notin L$ then $Pr[M(x) \text{ accepts }] = 0$

Actually, most languages in BPP are known to be in RP or $Co-RP$. For example, COMPOSITES is in both RP and $Co-RP$. The intersection of RP and $Co-RP$ is an interesting class also, and has its own name.

Definition 3 *ZPP is the class $RP \cap Co-RP$.*

The numbers 1/3 and 2/3 in the above definitions may seem somewhat arbitrary, and in fact they are. We can reduce the error probability of an algorithm almost as much as we like by simple repitition. Note that after k (independent) repetitions of an RP algorithm, (assuming that we accept if any single repetition accepts)

If $x \notin L$, then $M(x)$ rejects on every repetition.
If $x \in L$, then $M(x)$ rejects with probability at most $(1/3)^k$.

Similarly, we may repeat a BPP algorithm, accepting if most of the repetitions accept. In both cases, the error probability drops dramatically (exponentially) with the number of repetitions.

As a matter of fact, by boosting the accuracy of probabilistic algorithms sufficiently, we can relate the probabilistic complexity classes to circuit complexity classes, and to the polynomial hierarchy.

2 BPP and P/poly

Theorem 4 *$BPP \subseteq P/poly$.*

Proof idea: After boosting the accuracy of a BPP algorithm sufficiently, we will show that some sequence $r \in \{0,1\}^{p(n)}$ of random bits works well for all inputs of size n, and therefore is a good deterministic advice string. Extending this idea to all possible input sizes, we will have a sequence of good advice strings, placing L in $P/poly$.

Rather than finding such a sequence of random vectors r to serve as a sequence of advice strings, we give a non-constructive probabilistic proof of its existence. That is, if the fraction of sequences r which are good advice strings is non-zero, (over all possible such sequences) then some such sequence exists.

Proof: Let L be a language in BPP, decided by TM M with error probability $2^{-(n+2)}$. (We can reduce the error by repetition while increasing the running time only polynomially.) That is,

$$\text{If } x \in L,\ Pr[M(x,r) \text{ accepts }] \geq 1 - 2^{-(n+2)}$$
$$\text{If } x \notin L,\ Pr[M(x,r) \text{ accepts }] \leq 2^{-(n+2)}$$

Let us call a vector r of random bits "good for x" if it causes $M(x)$ to accept if $x \in L$ and to reject otherwise. For a particular input x,

$$Pr[r \text{ is good for } x] \geq 1 - 2^{-(n+2)}$$
$$Pr[r \text{ is bad for } x] \leq 2^{-(n+2)}$$

Now, the probability that r is good for all inputs is simply

$$\begin{aligned}
Pr[r \text{ is good for all inputs }] &= 1 - Pr[r \text{ is bad for at least one input }] \\
&\geq 1 - \sum_x Pr[r \text{ is bad for } x] \\
&= 1 - (\# \text{ of possible inputs }) \cdot 2^{-(n+2)} \\
&= 1 - 2^n \cdot 2^{-(n+2)} \\
&= \frac{3}{4} \\
&\geq 0
\end{aligned}$$

This shows that for any n, there exists a vector r which is good advice for all inputs of size n. Collectively, these vectors form a sequence of advice strings for L. Thus, $BPP \subseteq P/poly$.

Note that NP is not known to be contained in BPP. If it were, then the polynomial hierarchy would collapse to $\Sigma_2 P$, since this containment would imply that $NP \subseteq BPP \subseteq P/\text{poly}$. In addition, if $NP \subseteq BPP$, then $NP \subseteq RP$, implying that $NP = RP$. ($RP \subseteq NP$ because of one-sided error.) We do know, however, that BPP is contained in the second level of the polynomial hierarchy.

3 BPP and the Polynomial Hierarchy

Theorem 5 $BPP \subseteq \Sigma_2 P \cap \Pi_2 P$

Proof idea: For any language L in BPP, we will show that the statement $x \in L$ is equivalent to a quantified boolean formula $\exists y \forall z \phi(x, y, z)$, for some appropriate variables y, z and expression ϕ. Thus, we will show that $L \in \Sigma_2 P$, and since BPP is closed under complement, that $L \in \Pi_2 P$ as well.

For any language L in BPP, we know the following:

1. There is a TM M that decides L in probabilistic polynomial time with error probability 2^{-n} (by sufficient repetition).

2. M makes decisions by choosing a random vector r from the set $R = \{0,1\}^{p(n)}$ for some polynomial p.

3. For a given input x, a subset A_x of R contains vectors r that cause M to accept x (correctly or incorrectly). The size of this subset satisfies

   (a) If $x \in L$, then $\frac{|A_x|}{|R|} = 1 - 2^{-n}$

(b) If $x \notin L$ then $\frac{|A_x|}{|R|} = 2^{-n}$.

In order to create a quantified boolean formula for L, we introduce the idea of translation of vectors.

Definition 6 *Given strings r and t, $|r| = |t|$, we define the translation of r by t to be $r \oplus t$, where $\oplus$ denotes the bitwise XOR operation. Given set $S = \{r_1, ..., r_k\}$, we define the translation of S by t to be $S \oplus t = \{r_1 \oplus t, ..., r_k \oplus t\}$.*

Now, we propose the following equivalence, from which the theorem follows.

Proposition 7 *For any language L in BPP, $x \in L$ iff there exists a set $S = \{r_1, ..., r_k\}$, all translations of which contain at least one random vector r_i which causes M to accept x. i.e. iff*

$$\exists S\, \forall t\, [(S \oplus t) \cap A_x \neq \emptyset].$$

where $k = p(n)$.

We will prove both the "if" and the "only if" directions probabilistically, as we did for theorem 1. Keep in mind that if the fraction of sets S satisfying the proposition is non-zero (over all possible such sets), then some such set exists.

Proof: $x \in L \Rightarrow \exists S\, \forall t\, [(S \oplus t) \cap A_x \neq \emptyset]$
Suppose $x \in L$. We have the following string of equalities and inequalities:

$$\begin{aligned}
\Pr_S[\forall t :\ (S \oplus t) \cap A_x \neq \emptyset] &= 1 - \Pr_S[\exists t :\ (S \oplus t) \cap A_x = \emptyset] \\
&= 1 - \sum_t \Pr_S[(S \oplus t) \cap A_x = \emptyset] \\
&\geq 1 - (\#\text{choices for } t) \cdot \Pr_S[(S \oplus t) \cap A_x = \emptyset] \\
&= 1 - (\#\text{choices for } t) \cdot \prod_{i=1}^{p(n)} \Pr_{r_i}[(r_i \oplus t) \notin A_x] \\
&= 1 - 2^{p(n)} \cdot (2^{-n})^{p(n)} \\
&\gg 0.
\end{aligned}$$

We conclude that when $x \in L$, $\exists S\, \forall t\, [(S \oplus t) \cap A_x \neq \emptyset]$.

Proof: $x \notin L \Rightarrow \forall S\, \exists t\, [(S \oplus t) \cap A_x = \emptyset]$.
Suppose $x \notin L$. We have the following string of equalities and inequalities:

Given a set $S = \{r_1, ..., r_k\}$,

$$\begin{aligned}
\Pr_t[(S \oplus t) \cap A_x = \emptyset] &= 1 - \Pr_t[(S \oplus t) \cap A_x \neq \emptyset] \\
&= 1 - \sum_{r \in S} \Pr_t[r \oplus t \in A_x] \\
&\geq 1 - (|S|)\Pr_t[r \oplus t \in A_x] \\
&= 1 - k2^{-n} \\
&> 0 \quad \text{for } k < 2^n.
\end{aligned}$$

We conclude that when $x \notin L$, then for any set S, $|S| < 2^n$, $\exists t$ s.t. $[(S \oplus t) \cap A_x = \emptyset]$.

Thus we have shown that $BPP \subseteq \Sigma_2 P$, and since BPP is closed under complement, that $BPP \subseteq \Sigma_2 P \cap \Pi_2 P$.