Last revised 4/29/2010

Today we show how to reduce the error probability of probabilistic algorithms, prove Adleman's theorem that polynomial time probabilistic algorithms can be simulated by polynomial size circuits, and we give the definition of the polynomial hierarchy

1 Adleman's Theorem

Last time we mentioned that if we start from a randomized algorithm that provides the correct answer only with probability slightly higher than half, then repeating the algorithm many times with independent randomness will make the right answer appear the majority of the times with very high probability.

More formally, we have the following theorem.

Theorem 1 (Chernoff Bound) *Suppose $X_1, \ldots, X_k$ are independent random variables with values in $\{0,1\}$ and for every i, $\mathbb{P}[X_i = 1] = p_i$. Then, for any $\epsilon > 0$:*

$$\mathbb{P}\left[\sum_{i=1}^{k} X_i > \sum_{i=1}^{k} p_i + k\epsilon\right] < e^{-2\epsilon^2 k}$$

$$\mathbb{P}\left[\sum_{i=1}^{k} X_i < \sum_{i=1}^{k} p_i - k\epsilon\right] < e^{-2\epsilon^2 k}$$

The Chernoff bounds will enable us to bound the probability that our result is far from the expected. Indeed, these bounds say that this probability is exponentially small with respect to k.

Let us now consider how the Chernoff bounds apply to the algorithm we described previously. We fix the input x and call $p = \mathbb{P}_r[A(x,r) = 1]$ over all possible random sequences. We also define the independent 0/1 random variables $X_1, \ldots, X_k$ such that $X_i = 1$ if and only if $A(x, r_i)$ outputs the correct answer.

First, suppose $x \in L$. Then the algorithm $A^{(k)}(x, r_1, \ldots, r_k)$ outputs the right answer 1, when $\sum_i X_i \geq k/2$. So, the algorithm makes a mistake when $\sum_i X_i < k/2$.

We now apply the Chernoff bounds to bound this probability.

$$\mathbb{P}[A^{(k)} \text{outputs the wrong answer on } x]$$

$$= \mathbb{P}[\sum_i X_i < \tfrac{k}{2}]$$
$$\leq \mathbb{P}[\sum_i X_i - kp \leq -\tfrac{k}{6}]$$
$$\leq e^{-k/18}$$
$$= 2^{-\Omega(k)}$$

The probability is exponentially small in k. The same reasoning applies also for the case where $x \notin L$. Further, it is easy to see that by choosing k to be a polynomial in $|x|$ instead of a constant, we can change the definition of a **BPP** algorithm and instead of the bound of $\frac{1}{3}$ for the probability of a wrong answer, we could equivalently have a bound of $1/2 - 1/q(|x|)$ or $2^{-q(|x|)}$, for a fixed polynomial q.

Would it be equivalent to have a bound of $1/2 - 2^{-q(|x|)}$?

Definition 2 PP *is the set of problems that can be solved by a nondeterministic Turing machine in polynomial time where the acceptance condition is that a majority (more than half) of computation paths accept.*

Although superficially similar to **BPP**, **PP** is a very powerful class; $\mathbf{P}^{\mathbf{PP}}$ (polynomial time computations with an oracle for **PP**) includes all of **NP**, quantum polynomial time **BQP**, and the entire polynomial hierarchy $\Sigma_1 \subseteq \Sigma_2 \subseteq \ldots$ which we will define later.

Now, we are going to see how the probabilistic complexity classes relate to circuit complexity classes and specifically prove that the class **BPP** has polynomial size circuits.

Theorem 3 (Adleman) $\mathbf{BPP} \subseteq \mathbf{SIZE}(n^{O(1)})$

PROOF: Let L be in the class **BPP**. Then by definition, there is a polynomial time algorithm A and a polynomial p, such that for every input x

$$\mathbb{P}_{r \in \{0,1\}^{p(|x|)}}[A(x,r) = \text{wrong answer for} \quad x] \leq 2^{-(n+1)}$$

This follows from our previous conclusion that we can replace $\frac{1}{3}$ with $2^{-q(|x|)}$. We now fix n and try to construct a circuit C_n, that solves L on inputs of length n.

Claim 4 *There is a random sequence $r \in \{0,1\}^{p(n)}$ such that for every $x \in \{0,1\}^n$ $A(x,r)$ is correct.*

PROOF: Informally, we can see that, for each input x of length n, the number of random sequences r that give the wrong answer is exponentially small. Therefore, even if we assume that these sequences are different for every input x, their sum is still less than the total number of random sequences. Formally, let's consider the probability over all sequences that the algorithm gives the right answer for all input. If this probability is greater than 0, then the claim is proved.

$$\mathop{\mathbb{P}}_{r}[\text{for every} \quad x, A(x,r) \quad \text{is correct}] = 1 - \mathop{\mathbb{P}}_{r}[\exists x, A(x,r) \quad \text{is wrong}]$$

the second probability is the union of 2^n possible events for each x. This is bounded by the sum of the probabilities.

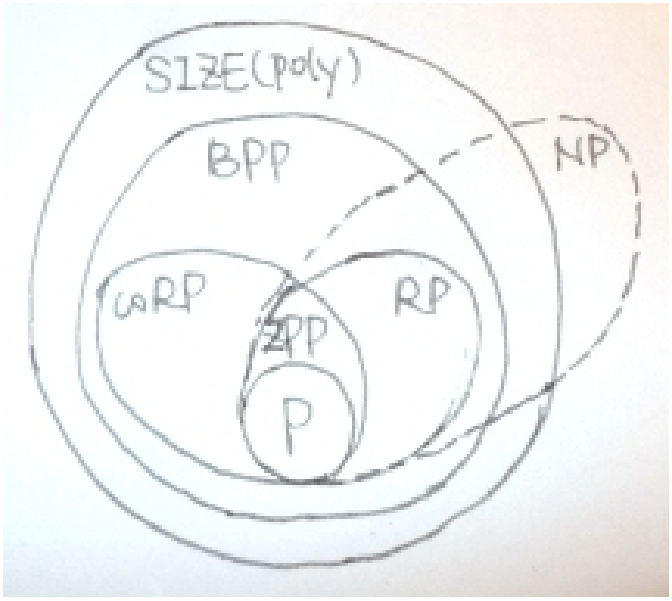
$$\begin{aligned} &\geq 1 - \sum_{x \in \{0,1\}^n} \mathop{\mathbb{P}}_{r}[A(x,r) \text{is wrong}] \\ &\geq 1 - 2^n \cdot 2^{-(n+1)} \\ &\geq \frac{1}{2} \end{aligned}$$

□

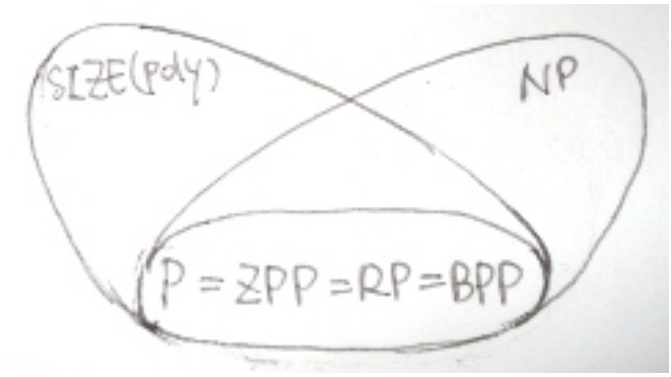
So, we proved that at least half of the random sequences are correct for all possible input x. Therefore, it is straightforward to see that we can simulate the algorithm $A(\cdot,\cdot)$, where the first input has length n and the second $p(n)$, by a circuit of size polynomial in n.

All we have to do is find a random sequence which is always correct and build it inside the circuit. Hence, our circuit will take as input only the input x and simulate A with input x and r for this fixed r. Of course, this is only an existential proof, since we don't know how to find this sequence efficiently. □

In general, the hierarchy of complexity classes looks like the following picture, if we visualize all classes that are not known to be equal as distinct.



It is, however, generally conjectured that $\mathbf{P} = \mathbf{BPP}$, in which case the complexity map greatly simplifies:



2 Complexity Classes with Advice

In this section we prove an alternative characterization of classes of functions computable by bounded-size circuits.

Let $a : \mathbb{N} \to \mathbb{N}$ be a function (e.g. $a(n) = 2n^2$).

Definition 5 $\mathbf{P}/a(n)$ *is the class of decision problems such there is a sequence of strings* $S_1, S_2, \ldots, S_n$ *where* $|S_n| \leq a(n)$*, and a polynomial-time algorithm* A *such that* $\forall x.A(x, S_{|x|})$ *correctly solves the problem.*

Definition 6 $\mathbf{P}/poly = \bigcup_k (\mathbf{P}/O(n^k))$

Theorem 7 $\mathbf{P}/poly = \mathbf{SIZE}(poly)$

PROOF: For any problem in $\mathbf{P}/poly$, there is an algorithm A and a sequence of strings $S_1, S_2, ..., S_n, ...$ that can solve it. For inputs of length n, we can construct $C_n = A(x, S_n)$. Such set of circuits will solve the problem.
For any problem in $\mathbf{SIZE}(poly$, there is a family of circuits $\{C_1, C_2, ..., C_n, ...\}$ that solves it. Consider constructing a circuit evaluation algorithm $A(x, C_n) = C_n(x)$.
□

3 Polynomial hierarchy

Remark 8 (Definition of NP and *co*NP) *A problem is in* **NP** *if and only if there is a polynomial time computable $F(\cdot,\cdot)$ and a polynomial time bound $p()$ such that*

$$x \text{ is a YES-instance} \Leftrightarrow \exists y.\ y \in \{0,1\}^{p(|x|)} \wedge F(x,y)$$

*co***NP** *is the class of problems whose complement (switch YES-instance to NO-instance) is in* **NP**. *Formally, a problem is in co***NP** *if and only if there is a polynomial time computable $F(\cdot,\cdot)$ and a polynomial time bound $p()$ such that*

$$x \text{ is a YES-instance} \Leftrightarrow \forall y : y \in \{0,1\}^{p(|x|)}, F(x,y)$$

The polynomial hierarchy starts with familiar classes on level one: $\Sigma_1 = \mathbf{NP}$ and $\Pi_1 = co\mathbf{NP}$. For all $i \geq 1$, it includes two classes, Σ_i and Π_i, which are defined as follows:

Definition 9 *Σ_k is the class of all problems such that there is a polynomial time computable $F(\cdot, ..., \cdot)$ and k polynomials $p_1(), ..., p_k()$ such that*

$$x \text{ is a YES-instance} \Leftrightarrow$$

$$\exists y_1 \in \{0,1\}^{p_1(|x|)}. \forall y_2 \in \{0,1\}^{p_2(|x|)}. \ \ldots$$

$$\ldots \underset{k \text{ is odd/even}}{\forall/\exists} \ y_k \in \{0,1\}^{p_k(|x|)}.\ F(x, y_1, \ldots, y_k)$$

Definition 10 *Π_k is the class of all problems such that there is a polynomial time computable $F(\cdot, ..., \cdot)$ and k polynomials $p_1(), ..., p_k()$ such that*

$$x \text{ is a YES-instance} \Leftrightarrow$$

$$\forall y_1 \in \{0,1\}^{p_1(|x|)}. \exists y_2 \in \{0,1\}^{p_2(|x|)}. \ \ldots$$

$$\ldots \underset{k \text{ is odd/even}}{\forall/\exists} \ y_k \in \{0,1\}^{p_k(|x|)}.\ F(x, y_1, \ldots, y_k)$$

One thing that is easy to see is that $\Pi_k = \text{co}\Sigma_k$. Also, note that, for all $i \leq k-1$, $\Pi_i \subseteq \Sigma_k$ and $\Sigma_i \subseteq \Sigma_k$. These subset relations hold for Π_k as well. This can be seen by noticing that the predicates F do not need to "pay attention to" all of their arguments, and so can represent classes lower on the hierarchy which have a smaller number of them.

Exercise 1 *$\forall k.\Sigma_k$ has a complete problem.*

Next time we will prove (a stronger version of) the following result:

Theorem 11 *If $\Sigma_{k+1} = \Sigma_k$, then $\forall t \geq k$, $\Sigma_t = \Sigma_k$.*