CMSC 651, Automata Theory, Fall 2010

Sandipan Dey,
Homework Assignment - 9 and 10

December 2, 2010

Homework 9

Problem 1

Show that, if $P = NP$, then $P = PH$.

Solution

1. Since P is closed under complements, we have

$$\begin{aligned} NP = P = \bar{P} = \bar{NP} = co - NP \\ \Rightarrow NP = co - NP \end{aligned}$$

2. Since $NP_k = \Sigma_p^k$, we have

$$\begin{aligned} PH \equiv NP \subseteq NP^{NP} \subseteq NP^{NP^{NP}} \subseteq \ldots \\ \equiv \Sigma_p^1 \subseteq \Sigma_p^2 \subseteq \ldots \Sigma_p^k \subseteq \ldots \end{aligned}$$

3. Also, let's prove the following:

$$\forall k \in \aleph,\ \Sigma_p^k = \Pi_p^k \Rightarrow \Sigma_p^{k+1} = \Sigma_p^k$$

Proof:

$$\begin{aligned} & A \in \Sigma_p^{k+1} \\ & \Rightarrow A = \{x | \exists y | y \leq |x|^c \wedge R(x, y)\},\ \text{with } c \text{ a costant and } R \in \Pi_p^k \\ & = \{x | \exists y | y \leq |x|^c \wedge R(x, y)\}, R \in \Sigma_p^k \text{ since } \Sigma_p^k = \Pi_p^k \\ & \Rightarrow A \in \Sigma_p^k \\ & \Rightarrow \Sigma_p^{k+1} \subseteq \Sigma_p^k \\ & \text{Also, } \Sigma_p^k \subseteq \Sigma_p^{k+1}, \text{ by definition} \\ & \Rightarrow \Sigma_p^k = \Sigma_p^{k+1} \text{ (Proved)} \end{aligned}$$

4. Hence,

$$\begin{aligned}
&NP = co - NP \\
&\Rightarrow \Sigma_p^1 = \Pi_p^1 = \Sigma_p^2 = \Pi_p^2 = \dots \\
&\Rightarrow PH \text{ collapses to } NP\text{, but given } P = NP \\
&\Rightarrow PH \text{ collapses to } P \text{ (Proved)}
\end{aligned}$$

Problem 2

A language L has polynomial-sized circuits $\Rightarrow \exists$ a sparse set $S | L \in P^S$.

Solution

1. L has polynomial-sized circuits $\Rightarrow L \in P/poly$.

   Proof:

   If L has polynomial sized circuits, let's define $s(n)$ to be the binary encoding for that circuit at length n. Construct the Turing Machine M as follows:

   $M(\langle x, s(|x|)\rangle)$

   (a) Construct the circuit given by $s(|x|)$.

   (b) Evaluates the output of the circuit given on x.

   (c) Accept x iff the circuit given by $s(|x|)$ accepts x.

   M runs in polynomial time (since the circuit evaluates in polynomial time), hence $x \in L \Rightarrow \langle x,\ s(|x|)\rangle \in L(M)$, where $L(M) \in P \Rightarrow L \in P/poly$ (can be decided using a polynomial size advice function).

2. $L \in P/poly \Rightarrow \exists S \mid L \leq_T^P S$, for some sparse set S.

   Proof:

   Let $L \in P/poly \Rightarrow$ some polynomial time Turing machine N accepts strings $\langle x, s(|x|)\rangle$ iff $x \in L$. We want to construct a sparse set S and a machine M so that M can discover $s(|x|)$ in polynomial time using S as an oracle. If we can do this, then afterwards M can simply simulate N (since it now knows $s(|x|)$), so that $L(M^S) = L$.

   Let's consider the language $S = \{1^n \# p | p \text{ is a prefix of } s(|x|)\}$. Now, S is sparse, since there are at most linearly many strings of a given length in S.

   Using S as an oracle, let's compute $s(|x|)$ one bit at a time: first, ask to the oracle if the strings $1n\#0, 1n\#1 \in S$. Let $1n\#b$ be the string out of these two which is in S. Then we can extend it to second bit of $s(|x|)$ by asking which of $1n\#b0$ or $1n\#b1$ is in S.

We proceed in this manner until neither extension of our string is in S. When this happens, we must have $s(|x|)$. Now, $s(|x|)$ has polynomially many bits, so this can be done in polynomial time.

3. 1. and 2. $\Rightarrow L$ has polynomial-sized circuits $\Rightarrow \exists S \mid L \leq_T^P S$, for some sparse set S.

Problem 3

Show that if there exists a sparse set S such that $coNP \subseteq NP^S$, then PH collapses to Σ_p^3.

Solution

By Karp-Lipton-Sipser, we have the following result:
If there exists a sparse set S such that $NP \subseteq P^S$, then PH collapses to Σ_p^2. Also, as Yaap has shown in his paper, a language L has small generators $\Rightarrow L \in NP(\Sigma_1/Poly)$ and $\Sigma_1/Poly = \Pi_1/Poly \Rightarrow \Sigma_{i+2} = \Pi_{i+2}$ and hence if every set in Π_1, has a small generator then $\Sigma_3 = \Pi_3$ which combined with problem 1 establishes that PH collapses to Σ_3.

Homework 10

Problem 1 Solution

Given:

- $0 < \epsilon_1 < \epsilon_2 < 1$, with ϵ_1, ϵ_2 fixed.
- M is a probabilistic polynomial time Turing machine that recognizes the language C with

$$w \in C \Rightarrow Pr[M \text{ rejects } w] \leq \epsilon_2$$
$$w \notin C \Rightarrow Pr[M \text{ accepts } w] \leq \epsilon_1 \leq \epsilon_2$$

When $\epsilon_2 \in (0, \frac{1}{2})$, it follows directly from the **amplification lemma** that $C \in BPP$. When $\epsilon_2 \in [\frac{1}{2}, 1)$, we have to show that the same result holds as well, i.e., the error probabilities on both the sides are bounded.

Let's consider the following exhaustive cases:

Case - 1) $0 < \epsilon_2 < \frac{1}{2}$ (The Amplification Lemma)

Construct a Turing machine N as follows:

$N(w)$

1. Compute k and Run M on w for k trials
   /* Compute k as in the amplification lemma */
2. Accept w if majority of trials accept
   otherwise reject w.

It's easy to see that N runs in polynomial time (since M does so). Now, let's prove that N decides C in BPP.

Proof (using Chernoff Bound directly)

We can think of the outcomes of the k runs of the Turing machine M to be represented by $X_1, \ldots, X_k$, k independent Bernoulli random variables, each having probability of success $1 - \epsilon_2 \geq \frac{1}{2}$ (where $\{X_k = 1\} \Leftrightarrow M$ accepts w when $w \in C$). Then the probability of simultaneous occurrence of more than $\frac{k}{2}$ of the events $\{X_k = 1\}$ has an exact value P, where

$$P = \sum_{i=\lceil \frac{k}{2} \rceil + 1}^{k} \binom{k}{i} (1 - \epsilon_2)^i . \epsilon_2^{k-i}$$

and Chernoff bound shows that P has the following lower bound

$$P \geq 1 - e^{-2k\left(1 - \epsilon_2 - \frac{1}{2}\right)} = 1 - e^{-2k\left(\frac{1}{2} - \epsilon_2\right)}$$

Hence,

$$\begin{aligned} Pr[E] &= Pr[N \text{ rejects } w | w \in C] \\ &= \sum_{i=0}^{\lceil \frac{k}{2} \rceil} \binom{k}{i} (1 - \epsilon_2)^i (\epsilon_2)^{k-i} \\ &= 1 - P \\ &\leq e^{-2k\left(\frac{1}{2} - \epsilon_2\right)} \end{aligned}$$

$$\Rightarrow \lim_{k \to \infty} Pr[N \text{ rejects } w | w \in C] = \lim_{k \to \infty} e^{-2k\left(\frac{1}{2} - \epsilon_2\right)} = 0 \text{ (since } \epsilon_2 < \tfrac{1}{2})$$

Similarly, $\lim_{k \to \infty} Pr[N \text{ accepts } w | w \notin C] = 0$

$$\Rightarrow C \in BPP \text{ when } \epsilon_2 < \frac{1}{2}$$

Proof that Majority works

Let's construct the Turing machine N as follows instead,

$N(w)$

1. Compute k and Run M on w for k trials
   /* Compute k as in the amplification lemma */
2. Compute the fraction f, $0 < f < 1$, as follows:
   $0 < \epsilon_2 < \frac{1}{2} \Rightarrow f > \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)}$
3. Accept w if more than f fraction of the outcomes
   ($> f.k$ trials out of k trials) accept w
   otherwise reject w.

It's easy to see that N runs in polynomial time (since M does so). Now, let's prove that N decides C in BPP.

When $w \in C$, the error probability $Pr[E]$ (probability that the Turing machine N rejects w) is upper bounded by the probability that at most f fraction of the outcomes (i.e., $\leq fk$ out of k outcomes) are correct, which is upper-bounded as follows:

$$\begin{aligned}
Pr[E] &= Pr[N \text{ rejects } w | w \in C] \\
&= \sum_{i=0}^{fk} \binom{k}{i} (1-\epsilon_2)^i (\epsilon_2)^{k-i} \\
&= \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \frac{1}{\delta^i} \\
&\text{where } \delta = \frac{\epsilon_2}{1-\epsilon_2}, \ \left(\epsilon_2 < \frac{1}{2} \Leftrightarrow \delta < 1\right), \ fk = f.k < k
\end{aligned}$$

Also, note that $w \in C \Rightarrow Pr[N \text{ accepts } w | w \in C] = 1 - P[E]$

$$\begin{aligned}
\text{When } \epsilon_2 < \frac{1}{2},\ Pr[E] &= \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \left(\frac{1}{\delta}\right)^i \\
&\leq \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \left(\frac{1}{\delta}\right)^{fk}, \text{ since } \frac{1}{\delta} > 1 \text{ and } i \leq fk \\
&\leq \epsilon_2^k \left(\frac{1}{\delta}\right)^{fk} \sum_{i=0}^{k} \binom{k}{i} = \left(\frac{2\epsilon_2}{\delta^f}\right)^k
\end{aligned}$$

Hence, we have:

$$0 < \epsilon_2 < \frac{1}{2} \Rightarrow Pr[E] \leq \delta_1^k, \text{ where } \delta_1 = \frac{2\epsilon_2}{\delta^f}$$

Hence, as shown above, in order to show $C \in BPP$, f should be pre-computed from ϵ_2 in such a manner that the upper bound (on error probability) on the right hand side can be made arbitrarily small by choosing larger and larger k, i.e.,

$$\lim_{k \to \infty} P[E] = 0 \Rightarrow \lim_{k \to \infty} \delta_1^k = 0 \Rightarrow 0 < \delta_1 < 1$$

Hence, $C \in BPP$ iff we choose the fraction f in such a manner that $P[E]$ can be made arbitrarily small, i.e.,

$$0 < \epsilon_2 < \frac{1}{2} \Rightarrow 0 < \delta_1 < 1 \Rightarrow f > \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)}$$

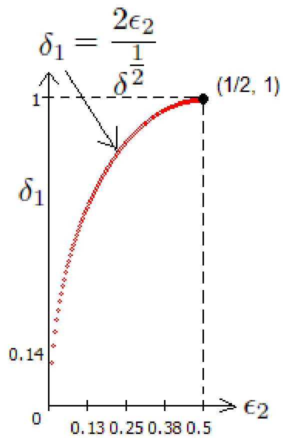
e.g.,

$$\lim_{\epsilon_2 \to \frac{1}{2}} \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)} = \left(\frac{\infty}{\infty}\right) = \frac{\left(\frac{1}{2\epsilon_2}\right).2}{\left(\frac{1-\epsilon_2}{\epsilon_2}\right).\frac{1}{(1-\epsilon_2)^2}} = \frac{1}{2}$$

The above proves the amplification lemma (see figure 1, 2), since for $\epsilon_2 < \frac{1}{2}$, it says that N can pick majority of the outcomes (if more than $f = \frac{1}{2}$, half the trials with M accept, N also accepts w).

Similar result can be shown for $w \notin C$, i.e., we can always pre-compute a proportion f as above to upper-bound the error probability $Pr[M \text{ accepts } w | w \notin C]$ and make it arbitrarily small.

Since error probabilities from both sides can be upper-bounded, N decides C in BPP (Proved).



for $f = \frac{1}{2}$, $\quad \delta_1 = \frac{2\epsilon_2}{\delta^{\frac{1}{2}}}$

$$\Rightarrow \delta_1 = \frac{2\epsilon_2}{\left(\frac{\epsilon_2}{1-\epsilon_2}\right)^{\frac{1}{2}}} = 2\,\epsilon_2^{\frac{1}{2}}(1-\epsilon_2)^{\frac{1}{2}}$$

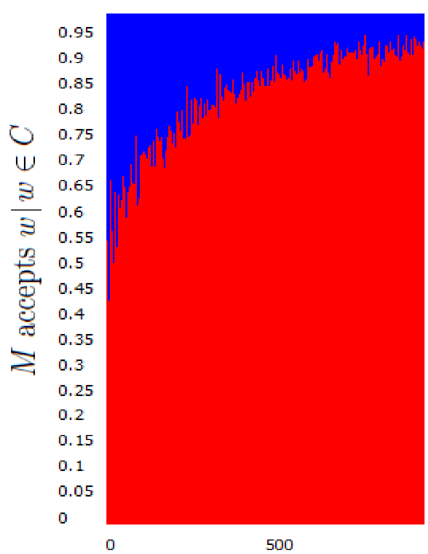
$$= 2\sqrt{\tfrac{1}{4} - (\tfrac{1}{2} - \epsilon_2)^2} \leq 2\sqrt{\tfrac{1}{4} - 0}$$

$$= 2\sqrt{\tfrac{1}{4}} = 2\cdot\tfrac{1}{2} = 1$$

$$\Rightarrow \delta_1 < 1 \;\Rightarrow \lim_{k\to\infty} P[E] \leq \lim_{k\to\infty} \delta_1^k = 0$$

$0 < \epsilon_2 < \frac{1}{2} \Rightarrow Pr[E] \leq \delta_1^k$ *amplification lemma: can use majority*

$\delta_1 = \frac{2\epsilon_2}{\delta^f}$, $\quad \delta = \frac{\epsilon_2}{1-\epsilon_2}$



ϵ_2 = 0.4 # iterations (k) M runs

Case - 2) $\frac{1}{2} \leq \epsilon_2 < 1$

Can't prove using Chernoff bounds

Here $\epsilon_2 > \frac{1}{2}$ and

$$\begin{aligned} P &= \sum_{i=\lceil \frac{k}{2} \rceil+1}^{k} \binom{k}{i} \epsilon_2^{k-i}.(1-\epsilon_2)^i \\ &= \sum_{j=0}^{\lceil \frac{k}{2} \rceil} \binom{k}{j} \epsilon_2^{j}.(1-\epsilon_2)^{k-j} \leq e^{-2k(\epsilon_2 - \frac{1}{2})} \end{aligned}$$

$$\begin{aligned} \Rightarrow Pr[E] &= Pr[N \text{ rejects } w | w \in C] \\ &= 1 - P \\ &= \sum_{j=\lceil \frac{k}{2} \rceil+1}^{k} \binom{k}{j} \epsilon_2^{j}.(1-\epsilon_2)^{k-j} \\ &\geq 1 - e^{-2k(\epsilon_2 - \frac{1}{2})}, \text{ by Chernoff bound,} \end{aligned}$$

a lower bound instead of an upper bound on the error probability!

Hence, Construct a Turing machine N' as follows:

$N'(w)$

1. Run a Bernoulli trial with probability of success p
2. If the trial outcome is sucess, then accept w
3. Else Run M on w for k trials and accept w if majority of trials accept otherwise reject w.

It's easy to see that N' runs in polynomial time (since M does so). Now, let's prove that N' decides C in BPP.

Proof

$$\begin{aligned} Pr[E] &= Pr[N' \text{ rejects } w | w \in C] \\ &= (1-p) \sum_{i=0}^{\lceil \frac{k}{2} \rceil} \binom{k}{i} (1-\epsilon_2)^i (\epsilon_2)^{k-i} \end{aligned}$$

We have to choose p arbitrarily small and accordingly choose k such that the error probability is upper bounded.

Similarly, $Pr[N \text{ accepts } w | w \notin C] = p + (1-p) \sum_{i=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{i} (1-\epsilon_2)^i (\epsilon_2)^{k-i}$.

Making these two side error probabilities arbitrarily small
$\Rightarrow C \in BPP$ when $\epsilon_2 \geq \frac{1}{2}$ as well.

Incorrect Proofs

Given:

- $0 < \epsilon_1 < \epsilon_2 < 1$, with ϵ_1, ϵ_2 fixed.
- M is a probabilistic polynomial time Turing machine that recognizes the language C with

$$w \in C \Rightarrow Pr[M \text{ accepts } w] \geq 1 - \epsilon_2$$
$$w \notin C \Rightarrow Pr[M \text{ accepts } w] \leq \epsilon_1 < \epsilon_2$$

When $\epsilon_2 \in (0, \frac{1}{2})$, it follows directly from the **amplification lemma** that $C \in BPP$. When $\epsilon_2 \in [\frac{1}{2}, 1)$, we have to show that the same result holds as well. We prove some generic result, for all $\epsilon_2 \in (0, 1)$.

Let's first construct a Turing machine N as follows:

$N(w)$

1. Compute k and Run M on w for k trials
   /* Compute k as in the amplification lemma */
2. Compute the fraction f, $0 < f < 1$, as follows:
   $$0 < \epsilon_2 < \frac{1}{2} \Rightarrow f > \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)}$$
   $$1 > \epsilon_2 \geq \frac{1}{2} \Rightarrow f > \frac{lg\left(2(1-\epsilon_2)\right)}{lg\left(\frac{1-\epsilon_2}{\epsilon_2}\right)}$$
3. Accept w if more than f fraction of the outcomes
   ($> f.k$ trials out of k trials) accept w
   otherwise reject w.

It's easy to see that N runs in polynomial time (since M does so). Now, let's prove that N decides C in BPP.

Proof

When $w \in C$, the error probability $Pr[E]$ (probability that the Turing machine N rejects w) is upper bounded by the probability that at most f fraction of the outcomes (i.e., $\leq fk$ out of k outcomes) are correct, which is upper-bounded as follows:

$$
\begin{aligned}
Pr[E] &= Pr[N \text{ rejects } w|w \in C] \\
&= \sum_{i=0}^{fk} \binom{k}{i} (1-\epsilon_2)^i (\epsilon_2)^{k-i} \\
&= \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \frac{1}{\delta^i} = (1-\epsilon_2)^k \sum_{i=0}^{fk} \binom{k}{i} \delta^{k-i} \\
&\text{where } \delta = \frac{\epsilon_2}{1-\epsilon_2},\ \left(\epsilon_2 < \frac{1}{2} \Leftrightarrow \delta < 1\right),\ fk = f.k < k
\end{aligned}
$$

Also, note that $w \in C \Rightarrow Pr[N \text{ accepts } w|w \in C] = 1 - P[E]$

$$
\begin{aligned}
\text{When } \epsilon_2 < \frac{1}{2},\ Pr[E] &= \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \left(\frac{1}{\delta}\right)^i \\
&\leq \epsilon_2^k \sum_{i=0}^{fk} \binom{k}{i} \left(\frac{1}{\delta}\right)^{fk},\ \text{since } \frac{1}{\delta} > 1 \text{ and } i \leq fk \\
&\leq \epsilon_2^k \left(\frac{1}{\delta}\right)^{fk} \sum_{i=0}^{k} \binom{k}{i} = \left(\frac{2\epsilon_2}{\delta^f}\right)^k \\
\text{When } \epsilon_2 \geq \frac{1}{2},\ Pr[E] &= (1-\epsilon_2)^k \sum_{i=0}^{fk} \binom{k}{i} (\delta)^{k-i} \\
&\leq (1-\epsilon_2)^k \sum_{i=0}^{fk} \binom{k}{i} (\delta)^{fk},\ \text{since } \delta > 1 \text{ and } i \leq fk \ \textbf{Incorrect assumption!!} \\
&\leq (1-\epsilon_2)^k (\delta)^{fk} \sum_{i=0}^{k} \binom{k}{i} = \left(2(1-\epsilon_2).\delta^f\right)^k
\end{aligned}
$$

Hence, we have the following exhaustive cases:

$$
\begin{aligned}
&1.\ 0 < \epsilon_2 < \frac{1}{2} \Rightarrow Pr[E] \leq \delta_1^k,\ \text{where } \delta_1 = \frac{2\epsilon_2}{\delta^f} \\
&2.\ 1 > \epsilon_2 \geq \frac{1}{2} \Rightarrow Pr[E] \leq \delta_2^k,\ \text{where } \delta_2 = 2(1-\epsilon_2).\delta^f
\end{aligned}
$$

Hence, as shown above, in order to show $C \in BPP$, f should be pre-computed from ϵ_2 in such a manner that the upper bound (on error probability) on the right hand side can be made arbitrarily small by choosing larger and larger k, i.e.,

$$
\lim_{k\to\infty} P[E] = 0 \Rightarrow \begin{pmatrix} 0 < \epsilon_2 < \frac{1}{2} \Rightarrow \lim_{k\to\infty} \delta_1^k = 0 \Rightarrow 0 < \delta_1 < 1 \\ 1 > \epsilon_2 \geq \frac{1}{2} \Rightarrow \lim_{k\to\infty} \delta_2^k = 0 \Rightarrow 0 < \delta_2 < 1 \end{pmatrix}
$$

Hence, $C \in BPP$ iff we choose the fraction f in such a manner that $P[E]$ can be made arbitrarily small, i.e.,

$$0 < \epsilon_2 < \frac{1}{2} \Rightarrow 0 < \delta_1 < 1 \Rightarrow f > \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)}$$

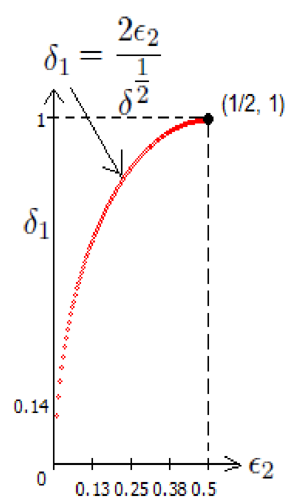
$$1 > \epsilon_2 \geq \frac{1}{2} \Rightarrow 0 < \delta_2 < 1 \Rightarrow f > \frac{lg\left(2(1-\epsilon_2)\right)}{lg\left(\frac{1-\epsilon_2}{\epsilon_2}\right)}$$

e.g.,

$$\lim_{\epsilon_2 \to \frac{1}{2}} \frac{lg(2\epsilon_2)}{lg\left(\frac{\epsilon_2}{1-\epsilon_2}\right)} = \left(\frac{\infty}{\infty}\right) = \frac{\left(\frac{1}{2\epsilon_2}\right).2}{\left(\frac{1-\epsilon_2}{\epsilon_2}\right)\cdot\frac{1}{(1-\epsilon_2)^2}} = \frac{1}{2}$$

The above proves the amplification lemma (see figure 1), since for $\epsilon_2 < \frac{1}{2}$, it says that N can pick majority of the outcomes (if more than $f = \frac{1}{2}$, half the trials with M accept, N also accepts w).

Similar result can be shown for $w \notin C$, i.e., we can always pre-compute a pro-



for $f = \frac{1}{2}, \quad \delta_1 = \frac{2\epsilon_2}{\delta^{\frac{1}{2}}}$

$$\Rightarrow \delta_1 = \frac{2\epsilon_2}{\left(\frac{\epsilon_2}{1-\epsilon_2}\right)^{\frac{1}{2}}} = 2\,\epsilon_2^{\frac{1}{2}}(1-\epsilon_2)^{\frac{1}{2}}$$

$$= 2\sqrt{\frac{1}{4} - (\frac{1}{2} - \epsilon_2)^2} \leq 2\sqrt{\frac{1}{4} - 0}$$

$$= 2\sqrt{\frac{1}{4}} = 2\cdot\frac{1}{2} = 1$$

$$\Rightarrow \delta_1 < 1 \Rightarrow \lim_{k\to\infty} P[E] \leq \lim_{k\to\infty} \delta_1^k = 0$$

$0 < \epsilon_2 < \frac{1}{2} \Rightarrow Pr[E] \leq \delta_1^k$ *amplification lemma: can use majority*

$$\delta_1 = \frac{2\epsilon_2}{\delta^f}, \quad \delta = \frac{\epsilon_2}{1-\epsilon_2}$$

portion f as above to upper-bound the error probability $Pr[M \text{ accepts } w | w \notin C]$ and make it arbitrarily small.

Since error probabilities from both sides can be upper-bounded, N decides N in BPP (Proved).

Yet another incorrect Proof

We know the following:

- By definition, for $0 \leq \epsilon < \frac{1}{2}$, a probabilistic polynomial time Turing machine M recognizes the language A with error probability ϵ if

$$w \notin A \Rightarrow Pr[M \text{ rejects } w] \geq 1 - \epsilon$$
$$w \in A \Rightarrow Pr[M \text{ accepts } w] \geq 1 - \epsilon$$

- If $\epsilon = \frac{1}{3}$, $A \in BPP$
- By amplification lemma, if ϵ be a fixed constant strictly between 0 and $\frac{1}{2}$, $A \in BPP$.

We are given the following:

- $0 < \epsilon_1 < \epsilon_2 < 1$, with ϵ_1, ϵ_2 fixed.
- M is a probabilistic polynomial time Turing machine that recognizes the language C with

$$w \notin C \Rightarrow Pr[M \text{ rejects } w] \geq 1 - \epsilon_1 \geq 1 - \epsilon_2$$
$$w \in C \Rightarrow Pr[M \text{ accepts } w] \geq 1 - \epsilon_2$$

Now, let's consider the following exhaustive set of cases:

1. $\epsilon_2 \in [0, \frac{1}{2})$, then it follows directly from the amplification lemma that $C \in BPP$.
2. $\epsilon_2 \in [\frac{1}{2}, 1)$, then construct another machine M' as follows:

   $M'(w)$

   - Runs M on input w repeatedly for k (constant, can be pre-computed from ϵ_2) times.
   - M' accepts if the proportion of M's acceptances is $\geq \epsilon_2$.
   - M' rejects if the proportion of M's acceptances is $< \epsilon_2$.

   We can choose the constant k depending upon ϵ_2 such that M' decides C in BPP.

Proof:

Let's define the random variable $X = \frac{1}{k}\sum_{i=1}^{k} X_i$, where

$$X_i = \begin{pmatrix} 1, & \text{if } i^{th} \text{ run of } M \text{ accepts } w \\ 0, & \text{if } i^{th} \text{ run of } M \text{ rejects } w \end{pmatrix}$$

Hence,

$$Pr[X_i = 1|w \in C] = Pr[M \text{ accepts } w|w \in C] \geq 1 - \epsilon_2$$
$$Pr[X_i = 0|w \notin C] = Pr[M \text{ rejects } w|w \notin C] \geq 1 - \epsilon_2$$

$$\begin{aligned}
Pr\left[M' \text{ rejects } w|w \notin C\right] &= Pr\left[X \leq \epsilon_2|w \notin C\right] \\
= Pr\left[\sum_{i=1}^{k} X_i \leq k\epsilon_2|w \notin C\right] &= 1 - Pr\left[\sum_{i=1}^{k} X_i > k\epsilon_2|w \notin C\right] \\
&\geq 1 - \frac{1}{k\epsilon_2}E\left[\sum_{i=1}^{k} X_i|w \notin C\right] \text{ (by Markov inequality)} \\
&= 1 - \frac{1}{\epsilon_2}.\frac{1}{k}\sum_{i=1}^{k} E\left[X_i|w \notin C\right] \text{ (by linearity of expectation)} \\
&= 1 - \frac{1}{\epsilon_2}.E\left[\bar{X}_i|w \notin C\right] \approx 1 - \frac{\mu'}{\epsilon_2} \text{ (with } \mu'\text{, a constant)} \\
&\text{where } E\left[\bar{X}_i|w \notin C\right] \to \mu' \text{ in probability, by WLLN} \\
&\Rightarrow Pr\left[M' \text{ rejects } w|w \notin C\right] \geq 1 - \epsilon'\text{, where } \epsilon' = \frac{\mu'}{\epsilon_2} \\
\text{Similarly, } Pr\left[M' \text{ accepts } w|w \in C\right] &\geq 1 - \epsilon''\text{, where } \epsilon'' = \frac{\mu''}{\epsilon_2} \\
&\text{and } E\left[\bar{X}_i|w \in C\right] \to \mu'' \text{ in probability, by WLLN}
\end{aligned}$$

Define $\epsilon = min(\epsilon', \epsilon'')$, so that we have,

$$Pr\left[M' \text{ rejects } w|w \notin C\right] \geq 1 - \epsilon$$
$$Pr\left[M' \text{ accepts } w|w \in C\right] \geq 1 - \epsilon$$

Since μ' and μ'' represent (population) means of $0-1$ random variables, both of them must be $< 1 \Rightarrow \epsilon < 1$. Also, $\epsilon_2 \geq \frac{1}{2} \Rightarrow \epsilon' = \frac{\mu'}{\epsilon_2} < \frac{1}{2}$

Problem 3 Solution

f and g be #P functions. By the definition of #P, this means there are nondeterministic machines N_1 and N_2 such that, on each input x, $f(x)$ equals the number of accepting paths of $N_1(x)$ and $g(x)$ equals the number of accepting paths of $N_2(x)$.

Proof: #P is closed under addition.

consider the nondeterministic machine N that, on input x, makes one initial nondeterministic choice, namely, whether it will simulate N_1 or N_2. Then the machine simulates the machine it chose. Note that, in effect, the computation tree of $N(x)$ is a tree that has a root with two children, one child being the computation tree of $N_1(x)$ and the other child being the computation tree of $N_2(x)$. So it is clear that the number of accepting paths of $N(x)$ is exactly $f(x) + g(x)$

Proof: #P is closed under multiplication.

Consider a nondeterministic machine N that on input x nondeterministically guesses one computation path of $N_1(x)$ and one computation path of $N_2(x)$ and then accepts if both guessed paths are accepting paths. Clearly the number of accepting paths of $N(x)$ is exactly $f(x)\,g(x)$, thus showing that #P is closed under multiplication.