Prob. Algorithms – Home Assignment #1

December 13, 2004

1. • Need to show: $RP \subseteq NP$. We show that any language RP is also in NP.

   Let L be a language and let A be a randomized algorithm such that for every x

   – $\text{Prob}_\Omega\Big[Time_A(x) \leq p(|x|)\Big] = 1$

   – If $x \in L$ then $\text{Prob}_\Omega\Big[A(x) = 1_L(x)\Big] \geq 1 - 1/3$.

   – If $x \notin L$ then $\text{Prob}_\Omega\Big[A(x) = 1_L(x)\Big] = 1$.

   where Ω is the set of all possible coin tosses' results.

   We define the following nondeterminsitc algorithm A': A' simulates A except that for any coin flipped by A, A' makes a nondeterministic choice from the set $\{0, 1\}$. If the nondeterministic choice is 0 then the simulation of A proceeds as if a coin toss has been resulted with "tails". If the nondeterministic choice is 1 then the simultation of A proceeds as if a coin toss has been resulted with "heads". The simulation of A continues and A' accepts an input iff A accepts it.

   We have to show that all computational paths of A' are of polynomial length and that the following two statements hold:

   (a) $x \in L$ implies some computational path of A' accepts x.

   (b) $x \notin L$ implies all computational path of A' rejects x.

   Since A runs in time polynomial in the input length, with probability 1, it follows that for all possible coin tosses, A runs in time polynomial in the input's length, and this implies that all computational paths of A' (each of which corresponds to a series of coin tosses,) are of length polynomial in the input's length.

   To see that the two other statements are valid, first notice that for all coin tosses, A always reject $x \notin L$. That means that for every possible coins tosses, A rejects $x \notin L$. This in turn implies that all computational paths of A' rejects x, if $x \notin L$. Lastly, note that if $x \in L$ then there exists a series of coin tosses which makes A accept x. In fact, most (a fraction of $(1 - \epsilon)$) such coin tosses leads A to accept x. Hence, since every series of coin tosses in A is a nondeterministic path in A', we have that most computational paths of A' accepts x, and in particular, there exists a computational path for A' which accepts x, for every $x \in L$.

   • Need to show: $ZPP \subseteq RP \subseteq BPP$.

- We show: $ZPP \subseteq RP$. We show that every language $L \in ZPP$ is in RP.
  Let L be a language in ZPP. Let A be an algorithm deciding L within expected time $p(|x|)$ and with no errors. We devise an algorithm A', as follows.
  Let $t = 3$. Given an input x, the algorithm A' simulates A on the input x, except that whenever the simulation takes time more than $tp(|x|)$, A' stops the simulation and enters a rejecting state (i.e., says x is not in L.) If the simulation returns some answer before the time limit is reached, then A' reports that answer which was reported by A.
  By construction of A', A' always runs in time at most $3p(|x|)$.
  The only possibility for A' to make an error is if $x \in L$ and the simulation of A took more than $3p(|x|)$ time. By Markov's inequality, $\Pr\Big[\text{time more than } 3p(|x|)\Big] \leq 1/3$ Hence, for $x \in L$, A' returns the right answer with probability at least $1 - 1/3$.
- We show: $RP \subseteq BPP$. This follows from definition.

- Need to show: $RP_{1-1/|x|} = RP_{1/e^{|x|}}$
  Note: clearly showing the above is enough as $RP_{1/3}$ is covered by that case.
  Let A be an algorithm deciding L. Assume A has the following properties:
  (a) $\Pr\Big[Time_A(x) \leq p(|x|)\Big] = 1$.
  (b) If $x \in L$ then $\Pr\Big[A(x) = 1_L(x)\Big] \geq 1 - (1 - 1/|x|) = 1/|x|$.
  (c) If $x \notin L$ then $\Pr\Big[A(x) = 1_L(x)\Big] = 1$.
  We give an algorithm A' deciding L, with the following properties:
  (a) $\Pr\Big[Time_A(x) \leq |x|^2 p(|x|)\Big] = 1$.
  (b) If $x \in L$ then $\Pr\Big[A(x) = 1_L(x)\Big] \geq 1 - 1/e^{|x|}$.
  (c) If $x \notin L$ then $\Pr\Big[A(x) = 1_L(x)\Big] = 1$.
  The algorithm A', given an input x, simulates A on x for $|x|^2$ times. If for some simulation of A on x, A accepts x then A' accepts x. Otherwise, A' rejects x.
  We next show that indeed $\Pr\Big[A(x) = 1_L(x)\Big] \geq 1 - 1/e^{|x|}$. If $x \in L$ then any one simulation of A on x is bound to give an error answer with probability at most $1 - 1/|x|$. The probability that all $|x|^2$ simulations will give the wrong answer on an input $x \in L$ is at most
  $$\Pr\Big[A \text{ errors on } x \in L \text{ for } t \text{ times}\Big] \leq (1 - 1/|x|)^{|x|^2}.$$
  Since
  $$(1 - 1/|x|)^{|x|^2} \leq e^{-|x|},$$
  the probability that A rejects an input $x \in L$ is at most $e^{-|x|}$. In other words, if A' simulates A for $|x|^2$ times and if $x \in L$ then with probability at least $1 - 1/e^{|x|}$, one of the $t = |x|^2$ simulations of A by A' will accept the input x.

Note the obvious: if $x \notin L$ then A rejects x with probability 1, since so does A'. Since A' has the property that $\Pr\Big[A(x) = 1_L(x)\Big] \geq 1 - 1/e^{|x|}$, and it runs in time $|x|^2 p(|x|)$, we have showed that $RP_{1-1/|x|} \subseteq RP_{1/e^{|x|}}$.

- Need to show: $BPP_{1/2-1/|x|} = BPP_{1/2^{|x|}}$

  Let L be a language and let A be an algorithm deciding L having the following properties:

  (a) $\Pr\Big[Time_A(x) \leq p(|x|)\Big] = 1$.

  (b) $\Pr\Big[A(x) = 1_L(x)\Big] \geq 1 - 1/2 + 1/|x|$.

  We devise an algorithm A' deciding L with the following properties:

  (a) $\Pr\Big[Time_{A'}(x) \leq |x|^3 p(|x|)\Big] = 1$

  (b) $\Pr\Big[A'(x) = 1_L(x)\Big] \geq 1 - 1/2^{|x|}$.

  Given an instance x, A' simulates A on x for $|x|^3$ times. If the majority of the answers yielded by simulations of A on x resulted with accepting x, then A' accepts x. Otherwise, A' rejects x.

  We first claim that

  $$\Pr\Big[A'(x) = 1_L(x)\Big] \geq 1 - 1/2^{|x|}.$$

  We show this claim is valid. A' makes an "error" if

  - $x \notin L$ and the number of accepts by the $|x|^3$ simulations of A on x is more than $|x|^3/2$.
  - $x \in L$ and the number of rejects by the $|x|^3$ simulations of A on x is more than $|x|^3/2$.

  Given $x \in L$, the expected number of accepts by simulations of A on x is $|x|^3(1/2 + 1/|x|) = |x|^3/2 + |x|^2$. Let $E = |x|^3/2 + |x|^2$.

  Hence, the probability that A' does not accept $x \in L$ is the probability that the number of simulations of A on x accepting x is less than $|x|^3/2$. By Chernoff, this is at most

  $$\begin{aligned} \Pr\Big[\#\text{ accepts of } A \text{ at most } |x|^3/2\Big] &= \\ \Pr\Big[|\#\text{ accepts of } A - E| \geq |x|^2\Big] &\leq \\ 2^{-\frac{|x|^2}{|x|^3/2+|x|^2}\frac{|x|^3/2+|x|^2}{6}} &\approx \\ 2^{-|x|} & \end{aligned}$$

  The analyzis for $x \notin L$ is similiar. Hence, we have shown that A' accepts $x \in L$ with probability at least $1 - 2^{-|x|}$.

  It is also clear that

  $$\Pr\Big[Time_{A'}(x) \leq |x|^3 p(|x|)\Big] = 1.$$

  Hence: $BPP_{1/2-1/|x|} = BPP_{1/2^{|x|}}$.

- Need to show: $ZPP = RP \cap coRP$.

  Let L be a language in $RP \cap coRP$. Then there are two algorithms deciding L: a first one, A_1, in RP and a second, A_2, in $coRP$. The first algorithm has the property that if it accepts an instance then surely the instance is in L. The second algorithm on the other hand, has the property that if it rejects an instance then that instance is not in L with probability 1.

  Given two such algorithms, A_1 and A_2, we build an algorithm A which decides L with probability 1 and in expected polynomial (in $|x|$) time. The algorithm A, given an input x, simulates both A_1 and A_2, independantly, each for an unbounded number of times – until either A_1 accepts or A_2 rejects. With high probability (approaching exponentialy fast to 1,) since x is either in L or not, one of the simulations – either the one of A_1 or the one of A_2 will accept, or reject x after only a polynomial number of simulations. Hence, the expected time algorithm A has in polynomial.

  Conversely, let A be an algorithm deciding a language Lin expected poylnomial time and with probability of success 1. We've already showed above that this implies L is in RP. A symmetric argument also shows that L is in $coRP$.

2. Let A be an array of length n. Denote the sorted elements of A by $a_1, a_2, \ldots, a_n$. Given an integers a, t, k, we are interested in finding an $a_{k'}$ such that $|k - k'| \leq an/\sqrt{t}$. Consider the following randomized algorithm:

   (a) Pick uniformly at random t elements $b_1, b_2, \ldots, b_t$ from the array.

   (b) Sort the elements $b_1, \ldots, b_t$. Assume w.l.o.g., $b_i < b_j \Leftrightarrow i < j$.

   (c) Return b_k (simply by indexing the sorted array of $b_1, \ldots, b_t$.)

   Claim 1 *Let $b_k = a'_k$. Then with probability larger than $1 - 2^{-\Omega(a^2)}$, $|k - k'| \leq an/\sqrt{t}$.*

   Proof: Let $\delta n = k$. We have to show that with probability larger than $2^{-\Omega(a^2)}$, we have that $b_k = a_{k'}$ and $|k' - k| \leq an/\sqrt{t}$.

   Let X_i be the indicator (0/1) random variable that the i-th element ($i \in [t]$) we choose is one of the first $\delta n - \frac{an}{\sqrt{t}}$ elements in the array. Let Y_i be the indicator (0/1) random variable that the i-th element we choose is one of the last $\delta n + \frac{an}{\sqrt{t}}$ elements in the array. Fix $X = \sum_i X_i$ and $Y = \sum_i Y_i$.

   Clearly, the algorithm fails if either $X \geq \delta n = k$, or if $Y \geq t - ....$ We show that the probability of the union of the above two events is small.

   We have
   $$P[X_i] = \frac{\delta n - an/\sqrt{t}}{n} = \delta - a/\sqrt{t}.$$
   Hence,
   $$E[X] = \delta t - a\sqrt{t}.$$

We now bound from above, using Chernoff's inequality, the probability that $X \geq \delta n$.

$$\begin{aligned}
\Pr\Big[X \geq \delta t\Big] &= \Pr\Big[|X - E[X]| \geq a\sqrt{t}\Big] \\
&= \Pr\Big[|X - E[X]| \geq \Big(\frac{a}{\delta\sqrt{t}}\Big)\delta t\Big] \\
&\leq 2^{-\frac{a^2}{\delta^2 t}\frac{\delta t}{6}} \\
&= 2^{-\Omega(a^2)}
\end{aligned}$$

The same analysis leads to a similiar upper bound on the event $Y \geq ....$ Hence, we conclude that with probability greater than

$$1 - 2^{-\Omega(a^2)},$$

$b_k = a_{k'}$ with $|k' - k| \leq an/\sqrt{t}$.

□

3. (a) Let $x = x_1, x_2, \ldots, x_n$, $y = y_1, y_2, \ldots, y_n$, where x_i, y_i are chosen uniformly and independantly of each other, in random from $\{0,1\}$. Let z_i be the random variable which equals 1 if $x_i = y_i$, and equals 0 otherwise. Clearly, since the x_i's are independant of each other (and so does the y_i's), the z_i's are independant. Let $Z = \sum_{i=1}^{n} z_i$. Clearly, $Z = d_H(x,y)$. We have that $E[Z] = n/2$, as the probability of z_i to be 1 is exactly half (as x_i, y_i are uniform.) Since the z_i's are independant 0/1 variables, we apply Chernoff's inequlity to obtain

$$\begin{aligned}
\text{Prob}_\Omega\Big[d_H(x,y) < n/4\Big] &= \text{Prob}_\Omega\Big[Z < n/4\Big] \\
&= \text{Prob}_\Omega\Big[|Z - n/2| \geq \frac{1}{2}n/2\Big] \\
&\leq 2^{-\frac{0.5^2(n/2)}{6}} \\
&= 2^{-\Omega(n)}
\end{aligned}$$

(b) Let c be the constant in the upper bound $2^{-\Omega(n)}$ we proved above. We choose uniformly at random $l = 2^{cn/2}$ strings in $\{0,1\}^n$.

From what we've proved above, we know that any two strings x, y of the l strings we chose, satisfy $d_H(x,y) < n/4$ with probability at most 2^{-cn},

Those, the expected number of strings in l which satisfy $d_H(x,y) < n/4$ is

$$2^{cn/2}2^{-cn} < 1.$$

Hence, by the pigeon hole property of the expectation, there exist a set of $l = 2^{cn/2}$ strings in $\{0,1\}^n$ such that no two satisfy $d_H(x,y) < n/4$.