

CS 159: Homework 6
Professor John Savage
Available: April 20, 2009
Due: April 27, 2009

Question 1

1. $\mathbf{P}/poly$ is the class of all languages recognized by a family of polynomial-sized circuit. In other words, for each $L \in \mathbf{P}/poly$, there is an infinite sequence of circuits such that the n^{th} circuit, C_n , recognizes all strings of length n , and has size at most $p(n)$, where $p(n)$ is a polynomial.

$\mathbf{P}/poly$ can also be defined as the class of all languages recognized by a polynomial-time deterministic Turing machines, M , with access to a polynomial-sized “piece of advice”, $a_M(n)$. Here $a_M(n)$ can be any arbitrary function of the input’s length, but not of the input itself.

Explain why these two definitions are equivalent.

2. \mathbf{BPP} is the class of all languages recognized by a polynomial-time probabilistic Turing machine, M , with completeness and soundness $2/3$. It is not hard to show that if $L \in \mathbf{BPP}$, there exists a polynomial time probabilistic Turing machine, M' , that recognizes L with completeness and soundness at least $1 - 1/2^m$.

Explain why this result implies that $\mathbf{BPP} \subseteq \mathbf{P}/poly$.

Hint: Given an input of length n , let $m = n + 2$ and then ask what advice can be given to a deterministic Turing machine.

3. **EXTRA CREDIT:** A probabilistic Turing machine, M , is simply a deterministic Turing machine with access to an infinite string of random bits. Each time M computes, each of these bits is randomly assigned the value 0 or 1 with equal probability.

Consider instead a “random-oracle Turing machine”, M' , which has access to a random infinite string that never changes. Each bit in this string is randomly chosen when M' is first created, but unlike with M , the bits do not change each time M' computes. As with M , M' begins each computation with its head over the first bit of the string.

Show that for any $L \in \mathbf{BPP}$ and $p < 1$, one can describe a random-oracle Turing machine, M' , such that when M' is created, the probability it recognizes L (with perfect completeness and soundness) is at least p .

Question 2

$\mathbf{BP} \cdot \mathbf{NP}$ is the class of all languages that can be recognized by a polynomial-time probabilistic Turing machine, M , with completeness and soundness $2/3$, where M has the added ability to make a single query to a machine, M' , that magically has the ability to recognize some \mathbf{NP} -complete language in constant time. In other words, $\mathbf{BP} \cdot \mathbf{NP}$ is the class of all languages that a polynomial-time probabilistic Turing machine can reduce to some \mathbf{NP} -complete problem with completeness and soundness $2/3$.

$\mathbf{NP}/poly$ is the class of all languages recognized by a family of polynomial-sized nondeterministic circuits. A **nondeterministic circuit**, C , is a circuit with two sets of input bits, X and Y . When X takes value x , the circuit outputs 1 if and only if there is some assignment to the remaining inputs, y , such that $C(x, y) = 1$.

Show that $\mathbf{BP} \cdot \mathbf{NP} \subseteq \mathbf{NP}/poly$

Question 3

In lecture we discuss why $\mathbf{IP} \subseteq \mathbf{PSPACE}$ but do not give a rigorous proof. We wish to show that, given a verifier, V , and an input, x , there is a polynomial-space algorithm that computes the maximum probability with which any prover, P , can cause V to accept.

Without loss of generality, assume that on any input, x , of size n , V asks P exactly $q(n)$ queries, where each query, q_i , and each response, a_i , is of length exactly $l(n)$. Furthermore, assume that each q_i is generated using $x, q_1, a_1, \dots, q_{i-1}, a_{i-1}$ and r_i , a sequence of $r(n)$ random bits. Here $q(n)$, $l(n)$ and $r(n)$ are all polynomials.

P 's response to q_k is a function of $x, q_1, a_1, \dots, q_{i-1}, a_{i-1}, q_k$, but none of the r_i 's. Let $P_V(q_1, a_1, \dots, q_{k-1}, a_{k-1}, q_k)$ denote the probability that V accepts when P provides the "optimal" response to the k^{th} query. Here optimal means that P responds so as to maximize the probability that V accepts (given that P does not know any r_i , nor x).

1. Consider V 's final query. Explain why $P_V(x, q_1, a_1, \dots, q_{p(n)-1}, a_{p(n)-1}, q_{p(n)})$ can be computed in polynomial-space.

Hint: It is fair to assume that P knows V . Consider enumerating over all possible answers, a_n , and for each, enumerating over all possible values of r_{n+1} (the random bits V uses to decide if it should accept).

2. Notice that q_{i+1} is a function of $(x, q_1, a_1, \dots, q_i, a_i)$ and r_{i+1} . Show that

$$P_V(x, q_1, a_1, \dots, q_{i-1}, a_{i-1}, q_i) = \max_{a_i} \sum_{r_{i+1}} \frac{1}{2^{r(n)}} P_V(x, q_1, a_1, \dots, q_i, a_i, q_{i+1})$$

and explain why the recurrence proves that P_V can be computed in polynomial space.

3. To conclude, explain that if V recognizes L with completeness and soundness $2/3$, $x \in L$ if and only if $\sum_{r_1} \frac{1}{2^{r(n)}} P_V(x, q_1) \geq 2/3$