

CS 6520: Computational Complexity

Problem Set 2

Due April 3, 2008

Problem 1

Prove that $\mathbf{BPL} \subseteq \mathbf{P}$.

Problem 2

Give a *directed* graph that has *exponential* cover time (in contrast with the polynomial bound for undirected graphs as shown in class). Conclude that the **RL**-algorithm for **USTCONN** based on a random walk fails when the graph is an arbitrary directed graph and thus fails to decide **STCONN**.

Problem 3

The complexity class **ZPP** is the class of languages decidable in *expected* polynomial time. That is, a language $L \in \mathbf{ZPP}$ is there is a probabilistic algorithm A and constants c, d such that for every $x \in \{0, 1\}^*$, both of the following conditions hold:

- $A(x) = \chi_L(x)$ always.
- The *expected* running time of A on x is at most $c|x|^d$.

Prove that $\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{coRP}$.

Problem 4

Prove that if $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{RP} = \mathbf{NP}$.

Problem 5

A *branching program* on variables x_1, \dots, x_n is a directed acyclic graph where all nodes are labeled with a variable x_i , except for two *output nodes*, one of which is labeled with 0 and the other labeled with 1. Both of the output nodes have outdegree 0. Every node other than the two output nodes has

outdegree 2, of which one outgoing edge is labeled with 0 and the other labeled with 1. One node with indegree 0 is designated as the start node.

A branching program on variables x_1, \dots, x_n defines a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, where for each $\alpha = (\alpha_1, \dots, \alpha_n) \in \{0, 1\}^n$, the value $f(\alpha)$ is defined as follows: Begin at the start node, and follow the path determined by taking the outgoing edge from each node that is labeled by the value assigned by α to the indicated variable,¹ until an output node is reached. The label of the output node reached is the value of $f(\alpha)$.

1. Prove that the class of Boolean functions computable by a polynomial-sized branching program is precisely **L/Poly**.
2. A branching program is *read-once* if every variable appears at most once in every path from the start node to an output node. Prove that the problem of deciding whether two read-once branching programs compute the same Boolean function, is in **coRP**. (**Hint:** You may find Polynomial Identity Testing useful.)

Problem 6

Prove that

1. Every language that has an interactive proof with a *deterministic* verifier, is in **NP**. More generally, every language that has an interactive proof with *zero soundness error* is in **NP**.
2. The class **IP** is not changed if the prover is allowed to be probabilistic.
3. **IP** \subseteq **PSPACE**.
4. Allowing a probability of error in the completeness condition does not change the class **IP**. That is, if a language has an interactive proof with completeness c and soundness error s , for any c and s satisfying $0 \leq s < c \leq 1$, then it has an interactive proof with perfect completeness.

¹That is, if the current node in the path is labeled with x_i , then take the edge labeled with α_i to reach the next node.

Problem 7

Read the definitions of complexity classes \mathbf{MA} , \mathbf{AM} , $\mathbf{AM}(k)$, $\mathbf{MA}(k)$, $\mathbf{IP}(k)$ in Section 8.4 of Arora-Barak.

- (a) Prove that $\mathbf{NP} \subseteq \mathbf{MA} \subseteq \mathbf{AM}$.
- (b) Prove that for every constant k , $\mathbf{AM}(k) = \mathbf{AM}(2) \stackrel{\text{def}}{=} \mathbf{AM}$.
- (c) Prove that every language in \mathbf{MA} has an \mathbf{MA} -proof with *perfect* completeness. (**Hint:** Use the ideas in the proof of $\mathbf{BPP} \subseteq \Sigma_2^{\mathbf{P}}$.)
- (d) Using similar ideas, prove that every language in \mathbf{AM} has an \mathbf{AM} -proof with *perfect* completeness. As a corollary, show that $\mathbf{AM} \subseteq \Pi_2^{\mathbf{P}}$.
- (e) Prove that if $\mathbf{coNP} \subseteq \mathbf{AM}$, then $\mathbf{PH} = \mathbf{AM}$.

It was shown by Goldwasser and Sipser that for every k , $\mathbf{IP}(k) \subseteq \mathbf{AM}(k+2)$. In particular, $\text{GRAPH NON-ISOMORPHISM} \in \mathbf{AM}$.

- (f) Using the above, prove that if GRAPH ISOMORPHISM is **NP-Hard**, then $\mathbf{PH} = \mathbf{AM}$.

Problem 8

1. Prove that if the error probability in the soundness condition of an interactive proof (with perfect completeness) is s , then *sequentially repeating* the proof system k times reduces the error to s^k .
2. Prove that the same result above holds for *parallel repetitions* of the proof system (with perfect completeness and a single prover), that is, when k copies of the protocol are executed simultaneously. (**Note:** For warm-up and partial credit, start with public-coin proof systems. For full credit, consider the general case. For the latter, you *cannot* assume that the cheating prover executes each copy of the protocol independently.)

Problem 9 (Extra Credit)

Prove that every *deterministic* algorithm for Polynomial Identity Test but with only oracle access to the tested polynomial $f : \mathbb{F}^n \rightarrow \mathbb{F}$, where \mathbb{F} is a finite field, the degree of f equals n which is the number of variables, and $|\mathbb{F}| > n$, requires at least 2^n queries to the f -oracle. Does this imply that $\mathbf{P} \neq \mathbf{RP}$? Justify your answer.

Problem 10 (Extra Credit)

In class the elegant **RL**-algorithm for USTCONN based on a random walk was analyzed using linear algebra. The following are a few results that were stated but not proven in class.

Let M be the transition matrix for a random walk on a d -regular undirected multi-graph² $G = (V, E)$ where $V = \{1, \dots, n\}$. Prove the following statements.

1. Every eigenvalue of M has absolute value at most 1.
2. G is connected if and only if 1 is an eigenvalue of multiplicity 1.
3. Prove that the second largest eigenvalue of M (*not* in absolute value) equals

$$\max_{x \perp \mathbf{1}, \|x\|=1} \langle Mx, x \rangle = 1 - \frac{1}{d} \cdot \min_{x \perp \mathbf{1}, \|x\|=1} \sum_{\{i,j\} \in E} (x_i - x_j)^2,$$

where both max and min are taken over unit vectors orthogonal to the all-1 vector.

4. Using the above, prove that if G is connected, then every eigenvalue of M other than 1 is at most

$$1 - O\left(\frac{1}{dn^3}\right).$$

5. G is bipartite if and only if -1 is an eigenvalue of M .
6. Using all above, prove that if G is connected and non-bipartite,

$$\lambda_2(G) \leq 1 - O\left(\frac{1}{dn^3}\right).$$

²A multi-graph is a graph where parallel edges and self loops are allowed.

7. Establish a tighter bound

$$1 - O\left(\frac{1}{dDn}\right)$$

for Part (4) and (6), where D is the diameter of G .