

## Midterm

Out: April 30

Due: May 7 at the beginning of class

This is a midterm. You may consult any of the course materials and the text (Papadimitriou), but not any other source or person. There are 5 problems on two pages. Please attempt all problems, and write clear, concise solutions. Good luck!

**To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.**

1. Show that  $\text{coNEXP} \subseteq \text{NEXP}/(n+1)$ . Here the “ $/(n+1)$ ” means that the nondeterministic machine takes exactly  $(n+1)$  bits of advice on an input of length  $n$ . Hint: use an idea similar to one you used for problem 2 on Problem Set 2.
2. A *branching program* is a directed acyclic graph in which each node is labelled by a variable  $x_i$ , one of these is designated as the *start node* and one is designated as the *accept node*. All of the nodes labelled with variables have exactly two outgoing edges, one labelled “0” and the other labelled “1”. An input  $x = x_1x_2 \dots x_n$  defines a path from the start node as follows: at a node labelled  $x_i$ , we follow the outgoing edge whose label coincides with the value of  $x_i$  in the input. If we reach the accept node, the input is accepted; otherwise the input is rejected.

Recall that **L/poly** is the class of languages decidable by a Turing machine in  $O(\log n)$  space with  $\text{poly}(n)$  bits of advice. Show that **L/poly** is exactly the class of languages decided by polynomial-size branching programs.

3. Recall that a restriction  $\rho$  applied to a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  assigns some variables 0 and 1 values, and leaves others free. We say that the *size* of a restriction  $\rho$  (denoted  $|\rho|$ ) is the number of variables to which it assigns 0 or 1 values. Define  $R_0(f) = \min_{\rho: f_\rho \equiv 0} |\rho|$  and  $R_1(f) = \min_{\rho: f_\rho \equiv 1} |\rho|$ . (By  $f_\rho \equiv 1$  we mean that every assignment to  $f_\rho$  evaluates to 1). For example, the parity function  $\bigoplus_n$  from Problem Set 3 has  $R_0(\bigoplus_n) = R_1(\bigoplus_n) = n$ .

For *monotone functions*  $f$  we denote by  $L_M(f)$  the leaf-size of the smallest *monotone* Boolean formula computing  $f$ .

- (a) Let  $f$  be a monotone function, and let  $\rho$  be a restriction for which  $f_\rho \equiv 1$ . Show that there exists a restriction  $\rho'$  that does not assign any variables the value 0 for which  $f_{\rho'} \equiv 1$ , and  $|\rho'| \leq |\rho|$ . Similarly, show that if  $\rho$  is a restriction for which  $f_\rho \equiv 0$ , then there exists a restriction  $\rho'$  that does not assign any variables the value 1 for which  $f_{\rho'} \equiv 0$ , and  $|\rho'| \leq |\rho|$ .
- (b) Prove the following lower bound for every monotone function  $f$ :  $R_0(f)R_1(f) \leq L_M(f)$ . Hint: as in Problem 3 of Problem Set 3, pick an optimal monotone formula for  $f$  and use induction on  $L_M(f)$ .

4. Show that  $\mathbf{NP} \subseteq \mathbf{BPP}$  implies  $\mathbf{NP} = \mathbf{RP}$ . Hint: first use error reduction to reduce the error probability of the  $\mathbf{BPP}$  machine.
5. (a) Let  $f$  be a family of one-way permutations, and let  $b = \{b_n\}$  be a hard bit for  $f^{-1}$ . Assume that both  $f$  and  $b$  are computable in polynomial time. Use  $f$  and  $b$  to describe a language  $L$  for which  $L \in (\mathbf{NP} \cap \mathbf{coNP}) - \mathbf{BPP}$ .  
(This shows that the assumption we used to construct the BMY pseudo-random generator placed *a priori* bounds on the power of  $\mathbf{BPP}$  – it presumed that  $\mathbf{BPP}$  was not powerful enough to simulate  $\mathbf{NP} \cap \mathbf{coNP}$ .)
- (b) Fix a constant  $\delta$ , and let  $g = \{g_n\}$  be a uniform family of functions for which:
- each  $g_n$  maps  $t = O(\log n)$  bits to  $m = n^\delta$  bits, and is computable in  $\text{poly}(n)$  time, and
  - for all circuits  $C : \{0, 1\}^m \rightarrow \{0, 1\}$  of size at most  $m$ ,

$$\left| \Pr_{y \in \{0,1\}^m} [C(y) = 1] - \Pr_{z \in \{0,1\}^t} [C(g_n(z)) = 1] \right| < 1/6.$$

Use  $g$  to describe a language  $L \in \mathbf{E}$  which does not have circuits of size  $2^{\epsilon n}$ , for some constant  $\epsilon > 0$ . Hint: refer to a function family obtained by truncating the output of  $g$  to  $t + 1$  bits.

(Notice that  $g$  is a “Nisan-Wigderson style” pseudo-random generator, which we were able to construct based on the assumption that there is some language in  $\mathbf{E}$  that does not have circuits of size  $2^{\epsilon n}$  for some constant  $\epsilon$ . This problem shows that this assumption is also *necessary* for the existence of such generators.)