CS 4510 : Automata and Complexity : Make up Assignment

Due on or before April 23, 2010

March 24, 2010

The Objective

The objective of this assignment is to help you learn some additional material in complexity theory, which would count for additional credit towards the course grade. This assignment shall replace the lower score out of your two midterm grades, if you score more than that in this assignment. This is a **purely optional** assignment.

What needs to be done

I have 5 topics listed below, with references and a short description of them. The interested students should email me back their top 2 (or 3) choices of topics by Wednesday March 31, 2010 before class. I expect you to read the reference(s) listed below (plus additional references if necessary), and prepare a write up, not exceeding 3 pages, by April 23, 2010. The report should contain all the main points and key techniques used in the proofs, **in your own words**. I **do not** want you to copy down the definitions, or the details to all the proofs in the report, but I expect you to fully understand the material. In fact, you would not get credit for any part that has been copied down from a reference. It is not mandatory, but I encourage you to use LaTeXto prepare your reports.

There would be a 15-20 minute oral part to this assignment. This shall be scheduled on a convenient time after the report is turned in. This is just to make sure that you have learned what was expected. The grade would be based on the coherence, clarity of exposition and the level of details in the report plus the reflection of your knowledge in the oral part.

Feel free to talk to me about the topics in the assignment. I also encourage you to discuss with the fellow student who has the same topic as you do. If you want help with LaTeX, talk to me. I shall help to get you started, and can give you a template.

The topics

1. **The complexity of *PRIMES*:** The complexity of determining if a given number a is prime or not.

   - Describe the Miller-Rabin primality test to show that *PRIMES* $\in$ BPP. This algorithm was discovered by Gary Miller (1976) and Michael Rabin (1980). Refer Sipser pages 371-375.
   - Show that *PRIMES* $\in$ NP. That primality has a short certificate. This was first shown by Vaughan Pratt in 1975. Refer the original paper by Pratt.

   Note: You may assume any necessary facts from algebra, as long as you state your assumptions clearly and explicitly.

2. **The classes FP, FNP and DP:** We have seen P and NP, classes of decision problems. The function classes FP and FNP are the function equivalents of P and NP respectively. Instead of a YES/NO answer, the function problems require the machine to compute a function. DP is the class of languages $L = L_1 \cap L_2$ where $L_1 \in$ NP and $L_2 \in$ co-NP.

   - The classes FP and FNP. Describe the polynomial time reductions between function problems, FNP-completeness, and the relation between P, NP, FP and FNP. Total functions which are FNP-complete. Refer Papadimitriou pages 227-234.
   - The class DP. DP-completeness. DP-completeness of *SAT-UNSAT* and *EXACT-TSP*. DP-completeness of *CRITICAL SAT*, *CRITICAL 3-COLORABILITY*. Show that *UNIQUE SAT* is in DP. Refer problem 17.3.2 and pages 412-415 from Papadimitriou.

3. **#P-completeness – Permanent:** The class #P and #P-completeness.

   - Counting classes. #P. Parsimonious reductions. #P-completeness. *#SAT*. Refer Papadimitriou pages 439-443.
   - *PERMANENT* is #P-complete. Define *PERMANENT* and show the completeness. Refer Papadimitriou pages 443-447 or chapter from the Arora-Barak book linked here.

   Note: The above two references for the #P-completeness of *PERMANENT* have *slightly* different reductions, though they are essentially the same. So please do not get confused with the two sources.

4. **P-completeness, Circuit Value, Odd flow:** The class P is closed under logspace reductions. We look at some P-complete problems.

   - We will see logspace reductions in class, when we study space complexity.
   - P-completeness. *CIRCUIT VALUE* is P-complete. *MONOTONE CIRCUIT VALUE* is P-complete. Refer Papadimitriou pages 165-171 or Sipser pages 404, 355-358 for the P-completeness of *CIRCUIT VALUE*. Thereafter, *CIRCUIT VALUE* is reduced to *MONOTONE CIRCUIT VALUE*.
   - *ODD MAX FLOW* is P-complete. Refer Papadimitriou pages 377-381 for the reduction from *MCV*.

You may also refer to the following notes from Berkeley and Harvard for the reduction to *CIRCUIT VALUE*.

5. **The complexity of cuts:** Show that *MAX CUT* is NP-complete, while *MIN CUT* is easy.

   - In class, we saw that *2SAT* is in P. Show that *MAX 2SAT* is NP-complete. Reduce *MAX 2SAT* to *NAE SAT* to *MAX CUT* to conclude that *MAX CUT* is NP-complete. Refer Papadimitriou pages 186-193.
   - You might have seen the max-flow min-cut equivalence in an algorithms class. Here we will see a simple efficient randomized algorithm for *MIN CUT* due to David Karger. Refer Jeff Erickson's notes on randomized min cut. (Jeff Erickson has a nice collection of notes on algorithms).

References

[1] Christos M. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.

[2] Michael Sipser, *Introduction to the Theory of Computation*, Course Technology, Boston, MA, 2006.