Qualifying Examination

Theoretical Computer Science

Saturday, February 28, 2009

Part II: Complexity

September 18, 2009

Instructions:

1. This is a closed book exam.
2. The exam is for 3 hours and has four problems of 25 points each. Read all the problems carefully to see the order in which you want to tackle them.
3. Write clearly and concisely. You may appeal to some standard algorithms/facts from text books unless the problem explicitly asks for a proof of that fact or the details of that algorithm.
4. If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half baked ideas just to get partial credit.
5. Each question is worth 25 points.

Problem 1

(By Lenny.)

Define *bipolar-3-SAT* as the set of Boolean formulas in 3CNF such that no clause contains both an unnegated variable and a negated variable. Either give a polynomial-time algorithm for finding a satisfying assignment for a *bipolar-3-SAT* problem, or prove that the decision problem is NP-complete.

Problem 2

(By Lenny.)

Recall that a DNF (Disjunctive Normal Form) formula is a sum (OR) of terms, with each term a product (AND) of literals. For example, $A\overline{B}C + \overline{D}EF$ is a DNF. A literal is a variable or its negation. A *monotone DNF* is a DNF where no variable is negated.

(A) [**5 Points**] What is the complexity of the following problem? Given two DNF formulas, determine whether or not they represent the same function. A brief explanation/justification is sufficient.

(B) [**20 Points**] A unate formula is one in which no variable appears both negated and unnegated. (A special case of a unate formula is a monotone one, in which each variable appears only unnegated.) What is the complexity of the following problem? Given two unate DNF formulas, determine whether or not they represent the same function. Prove that your answer is correct. Partial credit is given for solving the special case question about monotone DNF formulas.

Problem 3

(By Manoj.)

Recall that **PP** is the class of languages of the form

$$\{x \mid \text{a strict majority of } w \in \{0,1\}^{m(|x|)} \text{ is s.t. } R(x,w) = 1\}$$

where m is some polynomial and R is a polynomial time computable relation. Also recall that $\sharp\mathbf{P}$ is the class of counting functions of the form

$$x \mapsto |\{w \mid w \in \{0,1\}^{m(|x|)} \text{and } R(x,w) = 1\}|$$

where again m is some polynomial and R is a polynomial time computable relation.

Show that $\mathbf{P}^{\mathbf{PP}} = \mathbf{P}^{\sharp\mathbf{P}}$.

Problem 4

(By Manoj.)

(A) [**10 Points**] A *tournament* is a directed graph with a single edge between every two nodes, directed one way or the other. Show that in a tournament, with a vertex set V of N nodes (i.e., $|V| = N$), there is a subset of vertices $X \subseteq V$ of size $|X| = O(\log N)$ such that for every vertex $u \in V \setminus X$ there exists a vertex $v \in X$ such that there is an edge directed from v to u.

(In other words, show that there is a small number of players in a tournament such that every other player has defeated at least one of them.)

(B) [**15 Points**] A function f is a *selection function* for a language L if, on being provided with two inputs, it selects and outputs one of the two, such that if either one or both of the inputs are in L, then the output is also in L. That is,

$$f(x, y) = z \implies z \in \{x, y\} \text{ s.t. } \text{ if } \{x, y\} \cap L \neq \emptyset \text{ then } z \in L.$$

The complexity class **P**-sel is defined as the class of languages L for which there is a polynomial time computable selection function f.

Show that **P**-sel $\subseteq$ **P**/poly (where **P**/poly is the class of languages decidable by polynomial sized circuits, or equivalently by non-uniform polynomial time algorithms which take a polynomial sized advice for each input length).

Hint: A selection algorithm defines a tournament on any set of inputs. Consider the tournament over the set of n-bit strings in L.