**Theory Qualifying Exam**
**Fall 2011**

**Directions.** You have four hours. There are 4 problems, please do them all. If you cannot completely solve a problem, we will award partial credit for work that is correct and relevant to the question.

1.

We say that a language $L \subseteq \{0,1\}^* \times \{0,1\}^*$ *admits partial evaluation* if there exists a polynomial-time Turing machine $M$ and a function $f : \{0,1\}^* \to \{0,1\}^*$ such that for all $x, y \in \{0,1\}^*$, $|f(x)| \le |x|^{O(1)}$ and

$$\langle x, y\rangle \in L \Leftrightarrow M \text{ accepts } \langle f(x), y\rangle.$$

Consider the language $L$ consisting of all pairs $\langle x, y\rangle$ such that $x$ is a 3-CNF formula, $y$ is a clause, and $x \Rightarrow y$. (A clause is a disjunction of literals, and the arrow denotes logical implication.) For example, if $p$, $q$, and $r$ are propositional variables, then $\langle p \wedge q, q \vee r\rangle \in L$, but $\langle p \vee q, q \vee r\rangle \notin L$ and $\langle p \wedge q, p \wedge q\rangle \notin L$, because the second component is not a clause.

a) Show that if NP $\subseteq$ P/poly, then $L$ admits partial evaluation.

b) Show the converse.

Hint: Introduce a new variable $p_C$ for each possible 3-clause $C$ on the original variables, and consider expressions of the form

$$\wedge_C(C \vee p_C) \Rightarrow \vee_{C \in I} p_C,$$

where on the left, $C$ ranges over all possible 3-clauses on the original variables, and on the right over some subset $I$ of those clauses.

2.

The purpose of this problem is to derive an approximation algorithm to count the number of 0-1 solutions to a Knapsack problem.

Let $0 \leq a_1 \leq a_2 \leq \ldots \leq a_n \leq b$ be integers, and consider

$$S = \{x \in \{0,1\}^n \mid \sum_{i=1}^{n} a_i x_i \leq b\}.$$

Define $m$ to be the largest integer $\leq n$ such that $a_i \leq b/n$ for all $i \leq m$. Let $C = \{0,1\}^m \times \{0\}^{n-m}$. Observe that $C \subseteq S$. Let $n_i = \lfloor n^2 a_i / b \rfloor$. Let $S' = \{x \in \{0,1\}^n \mid \sum_{i=1}^{n} n_i x_i \leq n^2\}$.

a) Show that $|S'|$, the cardinality of $S'$, can be computed exactly in polynomial time. What is your running time?

b) Show that $S \subseteq S'$. Show also that for any $x \in S' - S$, there exists an integer $p > m$ with $x_p = 1$.

Note that $n_p \geq n$. Now define a map $f : S' \to S$, as follows:

c) For $x \in S$, let $f(x) = x$. For $x \in S' - S$, define $y = f(x)$ in such a way that $y \in S$ (prove this is possible). Your $y$ should differ from $x$ by only one bit.

d) Prove that $f(S') = S$ and that $|f^{-1}(y)| \leq n + 1$ for all $y \in S$.

e) Give an approximation algorithm with approximation factor $\sqrt{n+1}$. It should be deterministic. What is your running time?

f) Given a uniform sampling algorithm for $S$. It should sample $k$ uniform points of $S$ in time polynomial in $n$ and $k$, with probability $1 - e^{-\Omega(n)}$. By estimating $|S|/|S'|$, which is at least $1/(n+1)$, give a randomized approximation algorithm with the following property. Given any $\epsilon > 0$, it estimates $|S|$ to within a factor $1 + \epsilon$, in time poly$(n, 1/\epsilon)$. What's your running time?

3.

Let $\Sigma = \{0, 1\}$. A *language* is a subset of $\Sigma^*$. Given a language $L$, we call two strings $v, w$ *right equivalent* if for each $x$, the strings $vx$ and $wx$ are either both in $L$ or both out of $L$. A language has *finite index* if there are only finitely many right-equivalence classes. It is *regular* if it has a regular expression. It is well known that these two concepts coincide (Myhill-Nerode theorem).

a) For the language

$$L = \{ww : w \in \Sigma^*\},$$

exhibit an infinite number of strings that are all right-inequivalent. Conclude that $L$ is not regular.

Now consider subsets of $M = \Sigma^* \times \Sigma^*$ (also called languages). Right equivalence is defined the same way, but using the concatenation operation for $M$, which is

$$\langle x, x' \rangle \circ \langle y, y' \rangle = \langle xy, x'y' \rangle.$$

As before, to be recognizable means there are only finitely many right-equivalence classes. Regular expressions are generated from finite languages (it is enough to start with the pairs $\langle x, y \rangle$ with $|x|, |y| \leq 1$) using concatenation, union, and Kleene star.

b) The analog of the Myhill-Nerode theorem fails for $M$. Prove this by exhibiting a regular language that does not have finite index.

4.

One of Shannon's information theory experiments constructed approximations to English in which word pairs would occur with the correct frequencies. His method was the following. Choose some book with English text. Start with some word $w_1$. Now, for $n = 1, 2, \ldots$, suppose you have generated the string of words

$$w_1 w_2 ... w_n.$$

To get the next word $w_{n+1}$, open the book at random, and read until you see $w_n$. The next word following is used.

Using this method, he produced the sample

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH
WRITER THAT THE CHARACTER OF THIS POINT IS
THEREFORE ANOTHER METHOD FOR THE LETTERS THAT
THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN
UNEXPECTED .

We now consider a simplified version of Shannon's experiment.

There is a list of $N$ distinct words. Assume we have a mechanism (such as an urn or asymmetric die) that outputs the $i$-th word in the list with probability $p_i$, for $i = 1, \ldots, N$. When the mechanism is used repeatedly, successive outputs are independent.

As before, we give a process for choosing the next word, assuming we have chosen $w_1 w_2 \ldots w_n$. This process is as follows: Use the mechanism to sample words, until we get another copy of $w_n$. Then, sample again and call the result $w_{n+1}$.

a) For the simplified process, compute the expected number of samples (words read) to produce a sequence of $n$ words. Assume the first word is given.

b) Given a text, how would you pre-process it to make Shannon's original experiment more efficient?