CSE 532 Spring 2004
Computational Complexity Essentials
Exercises #2

Problems:

1. If you did not solve it for CSE 531 in Fall 2003, show that if \( \text{NP} \subseteq \text{BPP} \) then \( \text{NP} = \text{RP} \).

2. Show that if \( L \) is decided by a \( k \)-tape NTM in time \( O(T(n)) \) then \( L \) is decided by a 2-tape NTM in time \( O(T(n)) \).

3. Prove that #2SAT is #P-complete.

4. In the following problems you will consider approximations to #P problems.
   - An algorithm \( M \) is an \( \epsilon \)-approximation for a function \( f \) if for all \( x \), \((1 - \epsilon)f(x) \leq M(x) \leq (1 + \epsilon)f(x)\).
   - A randomized algorithm \( M \) is an \((\epsilon, \delta)\)-approximation for a function \( f \) if for all \( x \), \( \Pr[(1 - \epsilon)f(x) \leq M(x) \leq (1 + \epsilon)f(x)] \geq (1 - \delta) \).
   - A polynomial-time approximation scheme (PTAS) for a function \( f \) is a family of polynomial-time algorithms such that for any \( \epsilon > 0 \) there is a polynomial-time TM \( M_\epsilon \) such that \( M_\epsilon \) is an \( \epsilon \)-approximation for \( f \).
   - A fully-polynomial randomized approximation scheme (FPRAS) for a function \( f \) is a family of algorithms computing an \((\epsilon, \delta)\)-approximation for \( f \) whose running time is polynomial in the input size \( n \), \( 1/\epsilon \), and \( 1/\delta \).

FPRAS’s for some #P-complete problems have been known for about twenty years. Recently, solving a long-standing problem, Jerrum, Sinclair, and Vigoda developed one for PERM on 01-matrices.

(a) Show that unless \( P = \text{NP} \) there is no PTAS for any #P-complete problem.

(b) Show that unless \( \text{NP} = \text{RP} \), #HAMCYCLE does not have an FPRAS.

(c) Show that if \( \text{NP} = \text{RP} \), then any problem in #P has an FPRAS.

(d) Show that in the definition of an FPRAS we can without loss of generality improve the running time to polynomial in \( \log(1/\delta) \) but that we cannot improve the running time from polynomial in \( 1/\epsilon \) to polynomial in \( \log(1/\epsilon) \) unless \( \text{NP} = \text{RP} \).

5. **Approximate Counting is in the polynomial-time hierarchy:** Using similar ideas to the proof that \( \text{BPP} \in \Sigma_2^\text{P} \cap \Pi_2^\text{P} \), show that for any \( \epsilon > 0 \) there is an \( \epsilon \)-approximation in \( \text{FP}^{\Sigma_2^\text{P}} \) for any problem in #P.

6. **Unbounded fan-in circuits and Toda’s Theorem:** We can define unbounded fan-in circuits over a basis \( \Omega_0 \) consisting of \( \neg \) and unbounded fan-in \( \land \) and \( \lor \). The non-uniform complexity class \( \text{AC}^6 \) consists of all languages (or Boolean functions) computable by families of
polynomial-size, constant-depth unbounded fan-in circuits in this basis; that is, circuit families \{C_n\}_{n=0}^{\infty} in this basis such that \(\text{size}(C_n) = n^{O(1)}\) and \(\text{depth}(C_n) = O(1)\).

A function \(f \in \mathbb{B}_n\) is \textit{symmetric} iff for any permutation \(\sigma \in S_n\),

\[
f(x_1, \ldots, x_n) = f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}).
\]

Scale down the construction from the proof of Toda’s theorem to show that for any \(L \in \text{AC}^0\) there is a constant \(k\) such that \(L\) can be decided by a family of circuits \{\(C'_n\)\}_{n=0}^{\infty}, such that for each \(n\), the output gate of \(C'_n\) is a symmetric function of fan-in \(n^{O(\log^k n)} = 2^{O(\log^{k+1} n)}\), each of whose inputs is an \(\land\) of \(O(\log^k n)\) input variables or their negations.