Special Topics in Complexity Theory, Spring 2006, Yap
Homework 4
Due: Wed April 21.

INSTRUCTIONS

- REMEMBER OUR HOMEWORK RULES: you prepare a solution in writing, and then we will have appointments in which you need to give an oral defence of your solution. I will grade BOTH the oral defence AND your written solution.

---

1. (15 Points)
   Recall the transfer theorem we proved in class: suppose you have an algebraic algorithm (wlog, Real RAM algorithm) in which you use primitive operations from a set Ω. Each $f \in \Omega$ is a real function. Typically, $\Omega = \{\pm, \times, \div\} \cup \mathbb{Q}$. We call Ω the "computational basis" of the algorithm. The question is whether you can solve the same problem when implemented in a Turing machine. Our TRANSFER THEOREM says: if the evaluation of expressions over Ω can be computed to any relative precision by a Turing machine, then you can implement this on a Turing machine. Please talk to me if you are unclear about the framework or meaning of this theorem.

   Go over any of your favorite book on numerical analysis and look at the algorithms in this book.
   (i) Find any 3 algorithms, but the 3 algorithms must have *different* computational bases. Assume that the constants in the basis Ω is always $\mathbb{D}$ (dyadics). These algorithm will be assumed to be in the Real RAM model. Note that your 3 algorithms are allowed to solve the same problem.
   (ii) What is the definition of correctness of these algorithms? You must clearly specify the input and output formats,
   (iii) Give a reasonable definition of what it means to implement these Real RAM algorithms on a Turing machine (with error approximation).
   (iv) Can these algorithms be correctly implemented on a Turing machine according to your definition in (iii)?

2. (20 Points)
   Consider the following function $C(x, y)$ where $C(x, y) = (\min\{x, y\}, \max\{x, y\})$. This function is not reversible.
   (i) Construct a reversible circuit that realizes this function.
   (ii) Generalize C to a C_n that compares two n-quwords $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ where $C_n(\mathbf{x}, \mathbf{y}) = (\min\{\mathbf{x}, \mathbf{y}\}, \max\{\mathbf{x}, \mathbf{y}\})$. The comparison is lexicographic (or you can interpret this as comparison of binary numbers). Show how to realize C_n.

3. (20 Points)
   (i) Compute the eigenvalues and eigenvectors of the Hadamard matrix H

and the Pauli matrices $\sigma_x, \sigma_y, \sigma_z$.
(ii) Give the Bloch decomposition of these matrices.
(iii) Give the Spectral decomposition of these matrices.

4. (10 Points)
(i) Justify our calling QFT the quantum analogue of DFT.
(ii) Let $N = 2^n$. The Inverse QFT (IQFT) on N inputs is defined just like QFT, except that we replace $\omega_n = e^{\mathbf{i}2\pi/N}$ by $\overline{\omega_n} = e^{-\mathbf{i}2\pi/N}$. Prove that $IQFT(QFT(|x\rangle) = |x\rangle$ for all quantum states $|x\rangle$.

5. (10 Points)
Recall that the Bell state of an n-quword is $|x\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$.
(i) Show how to prepare such a state starting from $|0^n\rangle$ using the H-gate and T_2-gate.
(ii) What if you started with $|1^n\rangle$?

6. (10 Points)
Suppose you want to factor the number $n = 1991$ (the answer is $n = 11 \times 81$). Assume that you pick a random number $x \in \mathbb{Z}_n$ and wish to factor n by the reduction to order finding (as described in the last part of §14 in our notes). What is your probability of finding a factor of n with just one random choice of x?