Computational Models
Instructor: Benny Chor
T.A.: Gadi Kimmel
Winter Semester, 2003/4

# Assignment 5

Handed: 20/1/04, Due: 3/2/04

1. Recall than an oracle TM, $M^L$, is a TM with a special *query tape*. It can write down any string $s$ on the query tape, and in one additional step get a reply whether $s \in L$ or $s \notin L$. We define the class $P^{SAT}$ as the collection of languages decided by a polynomial time deterministic TM, with access to an oracle for SAT.
   Prove the following:
   (a) $NP \subseteq P^{SAT}$.
   (b) $coNP \subseteq P^{SAT}$.
   (c) $P^{SAT}$ is closed under complement.
   (d) Show that if $NP = P^{SAT}$ then $NP = coNP$.
   (e) Define $MaxClique := \{(G, k)|$ the size of the largest clique in $G$ equals $k\}$.
   Prove that $MaxClick \in P^{SAT}$.

2. Prove that the language $NA_{TM} = \{(< M >, w) \mid M$ is a non-deterministic TM that accepts $w\}$ is NP-Hard.

3. We say that a language $L$ is in $AvTime(T(n))$ if there is a deterministic Turing machine solving $L$, whose average running time over all inputs of size $n$, is at most $T(n)$. When defining the average, assume $\Sigma = \{0, 1\}$, and each string of length $n$ is weighted $\frac{1}{2^n}$. We denote $AvP = \bigcup_{c>0} AvTime(n^c)$. We also denote $E = \bigcup_{c>0} Time(2^{cn})$.
   Prove that $P \subseteq AvP \subseteq E$.

4. We say that a non-deterministic Turing machine is *nice* if for every input $x$ the following holds:
   - Every computation path returns either 'accept', 'reject' or 'quit'.
   - There is at least one non-quit path.
   - All non-quit paths have the same value.

   Let $NICE$ be the class of all languages $L$ that are accepted by some nice non-deterministic, polynomial time, Truing machine.
   Prove that $NICE = NP \cap coNP$.

5. Given an undirected connected graph $G = (V, E)$, we define a *spanning tree* of $G$ to be a subset $T \subseteq E$, such that $T$ connects all the vertices of $G$, and there are no cycles in $T$ ($T$ must be a tree).
   Define $k - SpanTree$ problem to be:
   Input: An un-weighted connected graph $G = (V, E)$, and a natural number $k$.
   Question: Is there a spanning tree of $G$, with at most $k$ leaves?
   Prove that $k - SpanTree$ problem is NP-Complete.

6. The problem $MaxEx3SAT$ problem is the following optimization problem: The input is a CNF formula $\phi$ with *exactly* three literals per clause (you can assume that no clause contains both $x$ and $\neg x$). Goal: Find an assignment that satisfies the *maximum* number of clauses.
   Suppose $\phi$ has $m$ clauses. Finding an assignment satisfying all $m$ clauses will solve 3SAT, so we do not really expect to do this in polynomial time. But we can try an *approximation algorithm*.

   (a) Find a polynomial time algorithm that satisfies at least $\frac{m}{2}$ of the clauses (hint: you may use randomization).

   (b) Argue that (a) is an algorithm with approximation ratio $\frac{1}{2}$.

   (c) Improve the approximation ratio to $\frac{7}{8}$.

   (d) To complete the picture, it has been shown (a couple of years ago) that an efficient algorithm with approximation ratio $\frac{7}{8} + \varepsilon$ (where $\varepsilon > 0$ is a constant) implies P=NP. (No action in this item, unless you want to prove P=NP...)