COSC 240, Fall 2018: Problem Set $\#7$

Assigned: Wednesday, 11/28.
Due: Wednesday 12/5, at the beginning of class (hand in hard copy).
Lectures Covered: 22 to 25
Academic Integrity: You can work with other people in the class but you must write up your own answers in your own words. You can also use the textbook and talk to the professor. You may not use any other resources (e.g., material found online) or talk to people outside the class about these problems. See the syllabus for details on the academic integrity policy for problem sets.

Problems

1. In class we proved that there does not exist an algorithm that *decides* the following language:

$$A_{TM} = \{\langle A, w\rangle \mid A \text{ is an algorithm that accepts input } w\}.$$

As discussed, however, there does exist an algorithm that *accepts* A_{TM} (i.e., the algorithm that on input $\langle A, w\rangle$ simulates A on w and accepts $\langle A, w\rangle$ if A accepts w in the simulation).

This problem ask you to consider the language $\overline{A_{TM}}$, which contains every string that is *not* in A_{TM}. Prove that there does not exist an algorithm that *accepts* $\overline{A_{TM}}$.

(*Hints:* Your proof should leverage the two facts reviewed in the problem statement; i.e., that A_{TM} cannot be decided but can be accepted. Your proof should be a contradiction proof; e.g., assume for contradiction that some algorithm H accepts $\overline{A_{TM}}$, and then use this assumption to create a new algorithm that uses H as a subroutine to solve something we previously proved impossible.)

2. Prove that each of the following three languages are in NP:

   (a) $COMPOSITES = \{\langle x\rangle \mid x = pq, \text{ for integers } p, q > 1\}$.
   (b) $NEQ = \{\langle \phi, \phi'\rangle \mid \phi$ and ϕ' are boolean formulas that are *not* equivalent$\}$.
   (c) $BADPAIRS = \{\langle S, k\rangle \mid S$ is a set containing natural numbers, and for every $x, y \in S, xy \neq k\}$.

3. Prove that the language $5SAT = \{\langle \phi\rangle \mid \phi$ is a boolean formula in 5-cnf with a satisfying assignment$\}$ is NP-complete.

(*Note:* 5-cnf is defined the same as 3-cnf except there are 5 literals in each or clause instead of 3.)

Extra Credit

4. Prove that A_{TM} is not in NP.

(*Hint #1:* Start by assuming for contradiction that A_{TM} *is* in NP. Then build on this assumption to produce a contradiction to a result we previously proved about this language.)

(*Hint #2:* Recall that our definition of a polynomial-time verifier requires that there is some polynomial $f(n)$, such that when the verifier is run on any input $\langle x, y\rangle$, where $n = |x|$, it terminates in no more than $f(n)$ steps. You can assume, without loss of generality, that $|\langle x, y\rangle| \leq f(n)$.)

5. As discussed in class, a *Hamiltonian cycle* in a graph $G = (V, E)$ is a simple path that forms a cycle over the nodes in V. That is, a path that starts at some $u \in V$, visits every node in $V \setminus \{u\}$ exactly once, then ends back at u. We can formalize the decision problem of searching a graph for a Hamiltonian cycle as follows:

$$HAMCYCLE = \{\langle G \rangle \mid G \text{ is an undirected graph that contains a Hamiltonian cycle}\}$$

A related problem is to look for a *Hamiltonian path* from a specific start node s to a specific end node t, which we define as a simple path that starts at s, ends at t, and visits every node in the graph exactly once. We can formalize the problem of searching for these paths as follows:

$$HAMPATH = \{\langle G, s, t \rangle \mid G \text{ is an undirected graph that contains a Hamiltonian path from } s \text{ to } t\}$$

It is well-known that HAMCYCLE is NP-complete. Use this assumption to prove that HAMPATH is also NP-complete.