CS 4810: Homework 6

due 10/24 11:59pm

(your name + netid)

Collaborators: (names and netids)

Each problem is worth 25 points.

Problem 1

You may attempt parts a and b independently.

Part a

Consider the following function $h : \mathbb{N} \to \mathbb{N}$,

$$h(k) = \max\{q \in \mathbb{N} \mid \text{there exists a Turing machine } M \text{ and an input } w \text{ with } |\langle M, w\rangle| \leq k \text{ and } M \text{ halts on input } w \text{ in } q \text{ steps}\}.$$

Show that h *grows too fast to be computable*, in the sense that not only is h not computable, but neither is any function h' such that $h(k) \leq h'(k)$ for all natural numbers k.

Part b

Suppose h is a non-decreasing function that grows to be too fast to be computable (e.g., the function in the previous part). Define a strictly increasing function $f(k) = h(k) + k$. For $x \in \mathbb{N}$, let

$$g(x) = \sup\{k \mid f(k) \leq x\},$$

be the greatest natural number k such that $f(k)$ is not larger than x. Note that $g(f(k)) = k$.

Show that $g(k)$ *grows too slowly to be computable*, in the sense that not only is g not computable, but neither is any *unbounded* function g' such that $g'(k) \leq g(k)$ for all natural numbers k.

(A function f is *unbounded* if for any y, there is some x such that $f(x) > y$.)

Problem 2

Part a

Recall the acceptance problem for Turing machines,

$$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M \text{ accepts on input } w\}.$$

Prove or disprove: $A_{\mathrm{TM}} \leq_m \overline{A_{\mathrm{TM}}}$.

Part b

Suppose P = NP. Characterize the set of NP-complete languages in this case.

Problem 3

We say that two graphs G and H with vertex set $\{1, \ldots, n\}$ are *isomorphic*, if we can reorder the vertices of H such that G and H are the same graph.

Show that the following language is in NP:

$$\textsc{graph isomorphism} = \{\langle G, H\rangle \mid G \text{ and } H \text{ are isomorphic}\}.$$

Problem 4

Consider the language of satisfiable Boolean formulas:

$$\textsc{sat} = \{\varphi \mid \varphi \text{ is a satisfiable Boolean formula}\}.$$

Suppose there exists a t-time algorithm to decide $\textsc{sat}$ for some function $t : \mathbb{N} \to \mathbb{N}$. Show that there exists a $O(t^2)$-time algorithm that given a satisfiable Boolean formula outputs a satisfiying assignment for the formula.