CMSC 28100-1 Spring 2015
Homework 8

May 22, 2015

1. (7 points) Show that if $\mathbf{P} = \mathbf{NP}$, then every language in $\mathbf{NP}$ is $\mathbf{NP}$-complete, except for $\emptyset$ and Σ^*.

2. Consider the following language:

$$K := \{(M, x, 1^t) \mid M \text{ is a NTM that accepts } x \text{ within } t \text{ steps}\}$$

(a) (2 points) Show that $K \in \text{NTIME}(n)$.

(b) (6 points) Show directly (not by reduction from another known $\mathbf{NP}$-complete language) that K is $\mathbf{NP}$-complete.

3. (10 points) Show that if SAT $\in \mathbf{P}$, then there is a deterministic polynomial-time Turing machine M such that for all formulas φ, if φ is satisfiable then $M(\varphi)$ outputs a satisfying assignment to φ, and otherwise M rejects. This is called solving the "search version" of SAT (searching for a witness, rather than merely determining if one exists).

4. (10 points) A language L is *p-selective* if there is a polynomial-time (deterministic) Turing machine M such that 1) $M(x, y) \in \{x, y\}$ –given (x, y), M outputs either x or y– for every pair of strings (x, y), and 2) if at least one of x or y is in L, then $M(x, y)$ outputs a string in L – which is necessarily either x or y, by (1).

Show that if SAT is p-selective, then $\mathbf{P} = \mathbf{NP}$. *Hint:* The solution to the previous problem contains a relevant idea.

5. (NOT GRADED, but if you want to submit a solution you will get feedback. Good practice of your understanding of Cook-Levin) A boolean formula $\varphi(x_1, ..., x_k)$ is a *succinct encoding* of a string s of length 2^k, as follows: the i-th bit of s is 1 if and only if $\varphi(i) = 1$, where we write i as k-bit binary number (some of its leading bits may be 0). For any formula φ, let decode(φ) denote the corresponding succinctly encoded string s. For any language L, define a new language SUCCINCT-L by

$$\text{SUCCINCT-}L := \{\varphi \mid \text{decode}(\varphi) \in L\}.$$

Let **NEXP** be the complexity class $\mathbf{NEXP} = \bigcup_k \text{NTIME}(2^{(n^k)})$. A language L is **NEXP**-complete if $L \in \mathbf{NEXP}$ and every $L' \in \mathbf{NEXP}$ polynomial-time reduces to L (the same as in **NP**-completeness).

Show that SUCCINCT-SAT is **NEXP**-complete. *Hint:* mimic the proof of the Cook-Levin theorem.