Comp487/587 - Exercise 3

- This exercise is due at the **beginning** of class.
- Questions labeled with **(COMP 587)** are required only for students in COMP 587. Questions labeled with **(Extra Problem)** are not required, but may be interesting to consider. All other questions are required both for students in COMP 487 and students in COMP 587.
- Collaboration is allowed and encouraged, but every student must write the solutions in his/her own words. Indicate clearly the name(s) of people you collaborated with.
- Note on the submission the total number of hours spent on this assignment.

Problem 1: Time and Space Constructible (25 points)

Let 1^n denote the integer n written in unary $(\underbrace{11\cdots 1}_{n \text{ times}})$. Consider a function $T : \mathbb{N} \to \mathbb{N}$.

- T is **time-constructible** if $T(n) \geq n$ and there is a DTM M that, on input 1^n, outputs $1^{T(n)}$ in $O(T(n))$ time.
- T is **space-constructible** if there is a DTM M that, on input 1^n, outputs $1^{T(n)}$ in $O(T(n))$ space.

Part 1 [10 points]: Prove that $f(n) = n^2$ is both time-constructible and space-constructible.

Part 2 [10 points]: Prove that $g(n) = 2^n$ is both time-constructible and space-constructible.

Part 3 [5 points]: Prove that $h(n) = \log_2(n)$ is space-constructible.

(Extra Problem) Part 4 [0 points]: Is $\log_2 \log_2(n)$ space-constructible?

Problem 2: NL (25 points)

An undirected graph $G = (V, E)$ is **bipartite** if the set of vertices V can be divided into two disjoint sets A and B such that every edge in E connects a vertex in A to a vertex in B. Prove that the following language is in NL:

$$BIPARTITE = \{G \mid G \text{ is bipartite}\}$$

Hint: A graph G is bipartite if and only if G has no odd length cycles.

Problem 3: DL (25 points)

Prove that the following language is in DL:

$$FVAL = \{(\varphi, \sigma) \mid \varphi \text{ is a Boolean formula and } \sigma \text{ satisfies } \varphi\}$$

Hint: Recall that we defined Boolean formulas recursively. You may assume that the only operators in the Boolean formula are $\neg$, $\vee$, and $\wedge$.

Problem 4: Cook-Levin (25 points)

A DTM M is **oblivious** if the location of the head at every step of the computation does not depend on the input but only on the size of the input. That is, the running time of M on all the inputs of the same size n is equal, and further the location of the head at step i depends only on i and on n. For example, a DTM that scans the input from left to right is oblivious.

Suppose that there is an oblivious DTM M (with a single tape) that verifies a language $L \in \text{NP}$, with running time $O(n^c)$ for some constant c. Prove that, in this case, we can construct in polynomial time a Boolean formula φ of size $O(n^c)$ such that $x \in L$ iff $\varphi \in SAT$. (In class we built a formula of size $O(n^{2c})$ without assuming that M is oblivious.)

(COMP 587) Problem 5 (20 points)

A language L is called **unary** if $L \subseteq \{1^n \mid n \geq 0\}$ (that is, if every word in L consists of only ones). Prove that if there exists an NP-complete unary language L, then $P = NP$.

Hint: If L is NP-complete, then there is a polynomial-time reduction from SAT to L. Use this reduction to prune recursive branches of the self-reducibility algorithm for SAT to obtain a poly-time algorithm for deciding SAT.