CPTH: Assignment 1

January 30, 2021

---

1. Try to write formal proofs unless stated otherwise
2. You should submit the solutions on **Moodle** by **EOD February 15th, 2021**

The only thing worse than being blind is having sight but no vision. - Helen Keller

Problem 1. Prove that any finite set of strings belongs to $DTIME(n)$.

Problem 2. Prove that if there is a polynomial time algorithm that converts a CNF formula to a DNF formula preserving the satisfiability, then $\mathbf{P} = \mathbf{NP}$.

Problem 3. Show that if **P=NP**, then every language $A \in P$ such that $A \neq \phi$ and $A \neq \Sigma^*$ is **NP**-complete. Explain why ϕ and Σ^* can never be **NP**-complete.

Problem 4. Let $S = \{\psi \mid \psi \text{ is Satisfiable 3CNF formula}\}$. Suppose we have a deterministic poly-time Turing machine M_S for deciding S. Describe a deterministic poly-time Turing machine M that given a 3CNF formula ϕ can write the satisfying assignment for ϕ on its output tape (using M_S).

Problem 5. A language L is said to be **unary language** if $L \subseteq \{1\}^*$. Prove that if unary lanuguage in **NP** is in **P**, then $EXP = NEXP$.

Problem 6. Prove or disprove: A language L is NP-complete iff L^C is coNP-complete.[1]

Problem 7. Prove or disprove the following statements:

1. If $L_1, L_2 \in \mathbf{NP}$ then $L_1 \cup L_2 \in \mathbf{NP}$ and $L_1 \cap L_2 \in \mathbf{NP}$
2. Let L be an **NP**-complete problem. If $L \in \mathbf{NP}$ and $L^C \in \mathbf{NP}$, then $\mathbf{NP} = \mathbf{coNP}$ [1]

---

[1] L^C represents the complement of L

3. $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$

Problem 8.

1. For any function $f(n) \geq \log n$, show that $\text{NSPACE}(f(n)) = \text{co-NSPACE}(f(n))$.

2. Is $\text{EXP}^{\text{EXP}} = \text{EXP}$. Justify your answer.

Problem 9. Σ_2SAT is the following decision problem: Given a CNF formula ϕ, decide whether $\psi = \exists x \forall y \phi(x, y) = 1$ is true. Show that if P=NP, then $\Sigma_2\text{SAT} \in \text{P}$.

Problem 10. Recall the definition of log-space transducer. A log-space transducer M, which is a Turing Machine, is said to computes a log-space computable function $f : \Sigma^* \mapsto \Sigma^*$ if on running M on input $w \in \Sigma^*$, it writes $f(w)$ on the output tape.
Let M_1 and M_2 be two log-space transducer computing the log-space computable functions f_1 and f_2. Show that there exists a log-space transducer M that computes the function $f : \Sigma^* \mapsto \Sigma^*$ such that $\forall w \in \Sigma^*. f(w) = f_1(f_2(w))$. In other words, show that the composition of two log-space transducers is another log-space transducer.