CPTH: Assignment 2

February 21, 2021

---

1. Try to write formal proofs unless stated otherwise
2. You should submit the solutions on **Moodle** by **EOD March 5th, 2021**

Of all forms of caution, caution in love is perhaps the most fatal to true happiness. - Bertrand Russell

Problem 1. In class, we defined the polynomial heirarchy using quantifiers. A language L is in Σ_2^p if there exists a polynomial time TM M and a polynomial q such that

$$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \forall u_2 \in \{0,1\}^{q(|x|)} M(x, u_1, u_2) = 1$$

.

Show that $\Sigma_2^p = NP^{NP}$

Problem 2. Show that $SPACE(n) \neq NP$.

Problem 3. Show that if a sparse language is **NP**-complete, then $\mathbf{P} = \mathbf{NP}$.

Problem 4. Prove the following:

a. **RP** and **BPP** are closed under union and intersection.

b. **BPP** is closed under complement. Is **RP** closed under complement?

c. There is a *decidable* language that is in **P/poly** but **not** in **P**.

d. If $\mathbf{NP} = \mathbf{P}^{SAT}$ then $\mathbf{NP} = \mathbf{coNP}$.

e. If $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{NP} = \mathbf{RP}$.

f. $\mathbf{BPP} \subseteq \mathbf{P/poly}$

Problem 5.

a. Prove that in the certificate definition of **NL** (Section 4.3.1 of Arora-Barak) if we allow the verifier machine to move its head back and forth on the certificate, then the class being defined changes to NP.

b. Show that the following language is NL-complete:

$$\{\langle G\rangle \mid G \texttt{ is a strongly connected digraph}\}$$

.

Problem 6. Let us define $\texttt{Maj}_n : \{0,1\}^n \mapsto \{0,1\}$ as:

$$\texttt{Maj}_n(x_1 \ldots x_n) = \begin{cases} 1 & \text{if } \Sigma_i x_i \geq n/2 \\ 0 & \text{otherwise} \end{cases}$$

Prove that $\texttt{Maj}_n$ can be computed by a circuit of size $O(n)$.

Problem 7. Let's say a language $L \subseteq \{0,1\}^*$ is in **P/poly** if there exists a polynomial $p : \mathbb{N} \mapsto \mathbb{N}$, a sequence of strings $\{\alpha_n\}_{n\in\mathbb{N}}$ with $\alpha_n \in \{0,1\}^{p(n)}$, and a deterministic polynomial time Turing Machine M such that for every $x \in \{0,1\}^n$

$$x \in L \Leftrightarrow M(x, \alpha_n) = 1$$

Let us call α_n to be the *advice string* for all x of the length n. Note that the *advice string* is **not** similar to a *witness* or *certificate* as used in the definition of **NP**. For example, all unary languages, even $UHALT$ which is undecidable, are in **P/poly** because the *advice string* can simply be a single bit that tells us if the given unary string is in $UHALT$ or not.

A set $S \subseteq \Sigma^*$ is said to be **sparse** if there exists a polynomial $p : \mathbb{N} \mapsto \mathbb{N}$ such that for each $n \in \mathbb{N}$, the number of strings of length n in S is bounded by $p(n)$. In other words, $|S^{=n}| \leq p(n)$, where $S^{=n} \subseteq S$ contains all the strings in S that are of length n.

1. Give the definition of **P/poly** as given in the class using polynomial size circuit families. Briefly describe how does this definition and the one given using advice string define the same class.

2. Given $k \in \mathbb{N}$ sparse sets $S_1, S_2 \ldots S_k$, show that there exists a sparse set S and a deterministic polynomial time TM M with oracle access to S such that given an input $\langle x, i \rangle$ the TM M will accept it if and only if $x \in S_i$.
Define the set S (note that it need not be computable), and give the description of M with oracle S.
Note that a TM M with oracle access to S can query whether $s \in S$ and get the correct answer in return in constant time.

3. Let us define a variant of **P/poly** called $\mathbf{P/poly}_{det}$ with a constraint that there should exists a polynomial time algorithm that can **compute** the advice string for any length $n \in \mathbb{N}$. In other words, there is a poly-time algorithm A such that $\alpha_n = A(n)$.
Is $\mathbf{P} = \mathbf{P/poly}_{det}$? Is $\mathbf{NP} = \mathbf{P/poly}_{det}$? Justify.

4. Let the language $L \in \mathbf{P/poly}$. Show that there exists a sparse set S_L and a deterministic polynomial time TM M with oracle access to S_L that can decide the language L.