Exercise 1 Family of circuits

Definition 1 A boolean circuit with $n$ inputs is an acyclic graph where the $n$ inputs $x_1, \ldots, x_n$ are part of the vertices. The internal vertices are labeled with $\lor$, $\land$ (with 2 incoming edges) or $\neg$ (with 1 incoming edge), with an additional distinguished vertex $o$ that is the output (with no exiting edge). The size $|C|$ of a circuit $C$ is its number of vertices (excluding the input ones). For a word $x \in \{0, 1\}^*$, the notation $C(x)$ refers to the output of the circuit $C$ if the input vertices of $C$ are valued with the bits of $x$.

Definition 2 For a function $t : \mathbb{N} \to \mathbb{N}$, a family of circuit of size $t(n)$ is a sequence $(C_n)_{n \in \mathbb{N}}$ such that: $C_n$ is an $n$-input circuit and $|C_n| \leq t(n)$.

Definition 3 A language $L \subseteq \{0, 1\}^*$ is decided by a family of circuit $(C_n)_{n \in \mathbb{N}}$ if for all $n \in \mathbb{N}$, for all $w \in \{0, 1\}^n$, we have: $C_n(w) = 1 \iff w \in L$.

Definition 4 For a function $t : \mathbb{N} \to \mathbb{N}$, we define $\text{SIZE}(t) := \{L \subseteq \{0, 1\}^* \mid L$ is decided by a family of circuits of size $O(t(n)) \}$. 

Definition 5 $\text{P/poly} := \bigcup_{k \in \mathbb{N}} \text{SIZE}(n^k)$

1. Show that any language $L \subseteq \{0, 1\}^*$ is in size $\text{SIZE}(n \cdot 2^n)$.
2. Show that for all function $t(n) = 2^{o(n)}$, there exists $L \notin \text{SIZE}(t(n))$.
3. Show that $\text{P/poly}$ contains undecidable language.
4. Show that $\text{P/poly}$ is not countable.

Exercise 2 Some alternation

1. Exhibit a polynomial time alternating algorithm that solves QBF.
2. Let $\text{ONE-VAL}$ be the problem of deciding whether a boolean formula is satisfied by exactly one valuation. Show that $\text{ONE-VAL} \in \Sigma_2^p$.
3. A boolean formula is minimal if it has no equivalent shorter formula where the length of the formula is the number of symbols it contains. Let $\text{MIN-FORMULA}$ be the problem of deciding whether a boolean formula is minimal. Show that $\text{MIN-FORMULA} \in \Pi_2^p$.

Exercise 3 Collapse of PH
1. Prove that if $\Sigma^P_k = \Sigma^P_{k+1}$ for some $k \geq 0$ then $\text{PH} = \Sigma^P_k$. (Remark that this is implied by $\mathbf{P} = \mathbf{NP}$).

2. Show that if $\Sigma^P_k = \Pi^P_k$ for some $k$ then $\text{PH} = \Sigma^P_k$ (i.e. $\text{PH}$ collapses).

3. Show that if $\text{PH} = \text{PSPACE}$ then $\text{PH}$ collapses.

4. Do you think there is a polynomial time procedure to convert any QBF formula into a QBF formula with at most 10 variables?

Exercise 4 Oracles

Consider a language $A$. A Turing machine with oracle $A$ is a Turing machine with a special additional read/write tape, called the oracle tape, and three special states: $q_{\text{query}}, q_{\text{yes}}, q_{\text{no}}$. Whenever the machine enters the state $q_{\text{query}}$, with some word $w$ written on the oracle tape, it moves in one step to the state $q_{\text{yes}}$ or $q_{\text{no}}$ depending on whether $w \in A$.

We denote by $\mathbf{P}^A$ (resp. $\mathbf{NP}^A$) the class of languages decided in by a deterministic (resp. non-deterministic) Turing machine running in polynomial time with oracle $A$. Given a complexity class $\mathcal{C}$, we define $\mathbf{P}^\mathcal{C} = \bigcup_{A \in \mathcal{C}} \mathbf{P}^A$ (and similarly for $\mathbf{NP}$).

1. Prove that for any $\mathcal{C}$-complete language $A$ (for logspace reductions), $\mathbf{P}^\mathcal{C} = \mathbf{P}^A$ and $\mathbf{NP}^\mathcal{C} = \mathbf{NP}^A$.

2. Show that for any language $A$, $\mathbf{P}^A = \overline{\mathbf{P}^A}$ and $\mathbf{NP}^A = \overline{\mathbf{NP}^A}$.

3. Prove that if $\mathbf{NP} = \mathbf{P}^{\text{SAT}}$ then $\mathbf{NP} = \overline{\text{coNP}}$.

4. Show that there exists a language $A$ such that $\mathbf{P}^A = \mathbf{NP}^A$.

5. We define inductively the classes $\mathbf{NP}_0 = \mathbf{P}$ and $\mathbf{NP}_{k+1} = \mathbf{NP}_{k}^{\mathbf{NP}_k}$. Show that $\mathbf{NP}_k = \Sigma^P_k$ for all $k \geq 0$.

\footnote{In fact, there also exists a language $B$ such that $\mathbf{P}^B \neq \mathbf{NP}^B$, which does not prove that $\mathbf{P} \neq \mathbf{NP}$.}