Complexité avancée - TD 5

Guillaume Scerri

October 21, 2022

Exercise 1 (Number Expressions). A *Number Expression* (or NE) e is an expression made up of natural numbers, and symbols '$+$' and '$\cup$' according to the following inductive definition:

$$e ::= n \mid (e_1 \cup e_2) \mid (e_2 + e_2)$$

where $n \in \mathbb{N}$ is any number. We often omit some of the parentheses when writing NEs. We define $|e|$, the size of e, as the number of occurrences of the symbols '0', '1', $+$' and '$\cup$' in e, assuming that numbers are written in binary. (NB: the real size on a TM tape includes the parentheses, hence is $O(|e|)$.)

An NE is interpreted as a subset $V(e)$ of $\mathbb{N}$, defined by

$$V(n) = \{n\}\,, \quad V(e_1 \cup e_2) = V(e_1) \cup V(e_2)\,,$$
$$V(e_1 + e_2) = \{n_1 + n_2 \mid n_1 \in V(e_1), n_2 \in V(e_2)\}\,.$$

1. Let $0 < n \in \mathbb{N}$ be a positive number and consider $e_n = (1 \cup 2) + (2 \cup 4) + \cdots + (2^{n-1} \cup 2^n)$. What is $V(e_n)$ and $|e_n|$?
2. Let $\mathsf{ISOLATED} = \{(e, n) \mid n \in V(e) \wedge n-1, n+1 \notin V(e)\}$. In other words, we consider the problem of checking whether a given number appears as an isolated value in some set of numbers denoted by a NE.

   Show that $\mathsf{ISOLATED} \in \mathsf{DP}$.

Exercise 2 (Σ_2^p and Π_2^p membership).

1. Let $\mathsf{ONE}-\mathsf{VAL}$ be the problem of deciding whether a boolean formula is satisfied by exactly one valuation. Show that $\mathsf{ONE}-\mathsf{VAL} \in \Sigma_2^p$;
2. 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 $\mathsf{MIN}-\mathsf{FORMULA}$ be the problem of deciding whether a boolean formula is minimal. Show that $\mathsf{MIN}-\mathsf{FORMULA} \in \Pi_2^p$.

Exercise 3 (Σ_2^p and Π_2^p completeness).

1. The classical Σ_2^p-complete problem is Σ_2^p-SAT (note that it can be assumed that the Boolean formula is in DNF). Consider now a different version of SAT denoted $\exists\exists! - \mathsf{SAT}$:

   - Input: a CNF-formula $\varphi(x, y)$ depending on the variables in x and y;
   - Outout: yes iff there exists x such that there exists a unique y satisfying $\varphi(x, \cdot)$.

   Show that $\exists\exists! - \mathsf{SAT}$ is also Σ_2^p-complete.

2. Similarly, the classical Π_2^p-complete problem is Π_2^p-SAT (the Boolean formula can be assumed in 3-CNF). Consider now a new notion of satisfiability: we say that a valuation ν nae-satisfies (for *not all equal*) a 3-CNF formula ϕ, if in all clauses (with at least two literals) of ϕ, ν both sets a literal to true and a literal to false. The clauses with only one literal only need to be satisfied. We consider now this new version of SAT denoted nae-$\Pi_2^p - \mathsf{SAT}$:

   - Input: a Π_2^p-SAT formula $\forall x, \exists y,\ \varphi(x, y)$ with φ a 3-CNF;
   - Outout: yes iff for all x, there exists y nae-satisfying φ.

   Show that nae-$\Pi_2^p - \mathsf{SAT}$ is Π_2^p complete.

Exercise 4 (Collapse of PH)**.** 1. Prove that if $\Sigma_k^P = \Sigma_{k+1}^P$ for some $k \geq 0$ then $\mathsf{PH} = \Sigma_k^P$. (Remark that this is implied by $\mathsf{P} = \mathsf{NP}$).

2. Show that if $\Sigma_k^P = \Pi_k^P$ for some k then $\mathsf{PH} = \Sigma_k^P$.

3. Show that if $\mathsf{PH} = \mathsf{PSPACE}$ then 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 5 (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: $q_{query}, q_{yes}, q_{no}$. Whenever the machine enters the state q_{query}, with some word w written on the oracle tape, it moves **in one step** to the state q_{yes} or q_{no} depending on whether $w \in A$.

We denote by P^A (resp. 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 $\mathsf{P}^{\mathcal{C}} = \bigcup_{A \in \mathcal{C}} \mathsf{P}^A$ (and similarly for NP).

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

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

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

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

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

Exercise 6 (Family of Circuits)**.**

Definition. A *boolean circuit with n inputs* is an acylic graph where the n inputs $x_1, \ldots, x_n$ are part of the vertices. The internal vertices are labeled with $\wedge$, $\vee$ (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. 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. 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: $C_n(w) = 1 \Leftrightarrow w \in L$.

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

Definition.

$$\mathsf{P}/poly := \cup_{k\in\mathbb{N}} \mathsf{SIZE}(n^k)$$

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