Tutorial 02 – The class NP (Group 4)

Exercise 1. *Boolean circuits*

Let $f_n\colon \{0,1\}^n \to \{0,1\}$ be the function defined as

$$f_n(x_1,\dots,x_n) := \begin{cases} 1 & \text{if } x_1+\cdots+x_n \text{ is odd,} \\ 0 & \text{otherwise.} \end{cases}$$

1. Draw a boolean circuit that computes f_0, f_1, f_2. What can you say about f_2?
2. Show that f_n can be computed by a circuit of size and depth $O(n)$.

Exercise 2. *Union and Intersection*

Let A and B be two languages. Then show that:

- If A and B are in NP, then so are $A \cup B$ and $A \cap B$
- What can you say about the NP-completeness of $A \cup B$ and $A \cap B$ when A and B are both NP-complete?

Exercise 3. NP-*Completeness*

Show that the following languages are NP-complete.

1. NDTM-T: given $\langle \alpha, x, 1^t \rangle$, does the nondeterministic Turing machine M_α accept on input x in time $\leqslant t$?
   Hint: You can admit that there exists a universal nondeterministic Turing Machine.
2. 3-SAT: Does a 3CNF formula ϕ have a satisfying assignment?
   You can admit that CNF-SAT *is* NP-*complete.*
3. 1-IN-3SAT: Given a collection of clauses $C_1, ..., C_m$, $m > 1$; such that each C_i is a disjunction of exactly three literals, is there a truth assignment to the variables occurring so that exactly one literal is true in each C_i?
   Hint: You can admit that 3-SAT is NP-complete.

Exercise 4. *Tally Language (Berman's Theorem, 1974)*

A language is said to be *tally* (or unary), if it is included in a unary alphabet $\{a\}^*$ for a fixed symbol a.

Definition (Subset-Sum). Given n numbers $v_1,\dots,v_n \in \mathbb{Z}$, and a *target* number $T \in \mathbb{Z}$, we need to decide whether there exists a nonempty subset $S \subseteq [\![1,n]\!]$ such that $\sum_{i\in S} v_i = T$. The problem size is $|T|_2 + \sum_{i=1}^{n} |v_i|_2$.

1. Prove that Subset-Sum is NP-complete.
2. Let Unary-Subset-Sum be the tally variant of Subset-Sum where all numbers are represented by their unary representation. Show that Unary-Subset-Sum is in P.
3. Show that if there exists an NP-hard tally language, then P = NP.