Tutorial 02 – The class NP (Group 3)

Exercise 1. *Boolean circuits*

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

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

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. *Decision vs Search*

If $\mathsf{P} = \mathsf{NP}$, show that there exists a polynomial time algorithm that returns an affectation of the variables satisfying a boolean formula $\phi(x_1, \ldots, x_n)$.

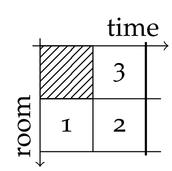
Exercise 3. NP-*Completeness*

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. k-Coloring: given $\langle G, k \rangle$, can we color the vertices of G using exactly $k \geq 1$ colors in such a way that there is no edge between any two vertices of the same color?
   Hint: This problem is NP*-complete even if we fix $k = 3$.*

3. Schedule: given $\langle C, (s_i)_{1 \leqslant i \leqslant C}, k \rangle \in \mathbb{N} \times \mathcal{P}(\llbracket 1, S \rrbracket)^C \times \mathbb{N}$, where C is the number of courses and s_i the list of students in course i. Is it possible to fit all the courses in k slots such that each student can go to every course he enrolled in?



Example of Schedule with entry $\langle 3, ([1,2],[1],[2,3]), 2 \rangle$.

   Hint: Reduce from the k-coloring problem.

4. 0/1 Integer Programming: Given a list of m linear inequalities with rational coefficients over n variables $u_1, \ldots, u_n$, find out if there is an assignments of 0s and 1 satisfying all the inequalities.

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 $v_1, \ldots, v_n \in \mathbb{Z}$, 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 $\mathsf{P} = \mathsf{NP}$.