Tutorial 02 – NP

Exercise 1.

1. Prove that 3-Sat is NP-complete.
2. Prove that Clique is NP-complete.
3. Prove that O/1-IProg is NP-complete.

Exercise 2.

1. Let NDTM-Time be the problem of deciding, given a TM M and a time bound t not greater that the number of states in M, whether M halts in less than t steps when its input tape initially is empty. Prove that NDTM-Time is NP-complete.

Definition. Let T be a set of square tiles (of unit length), each side of which has a color from a set C. A **tiling of the** $n \times n$ **square** is a map from $\{1, \dots, n-1\}^2$ to T. It is **valid** if the common sides of two adjacent tiles have the same color, and the side of the square is monochrome. It is **nontrivial** if it contains at least two different tiles.

2. Let FTP be the problem of deciding, given a finite set T of tiles with color in a finite set C and an integer n, whether there exists a valid and nontrivial tiling of the $n \times n$ square. Prove that FTP is NP-complete.

Exercise 3. *Sat-solver*

✎ Prove that if $\mathsf{P} = \mathsf{NP}$, there exists a polynomial-time algorithm which, given a CNF formula φ, returns a valid assignment for φ if some exists and 0 otherwise.

Exercise 4.

1. Prove that if every tally language[1] of NP is in P, then $\mathsf{EXP} = \mathsf{NEXP}$.
2. If $\mathsf{EXP} \neq \mathsf{NEXP}$, what about P and NP?

Exercise 5. *Berman's Theorem (1978)*

1. Give a recursive algorithm to solve Sat.
2. Let S be an NP-complete tally language. Using a polynomial-time reduction from Sat to S, improve the previous algorithm so that it runs in polynomial time.
3. Therefore?

[1] A language is said tally if it is a subset of $\{1\}^*$.