

Complexity Theory  
**Exercise 3: Time Complexity**  
2 November 2022

**Exercise 3.1.** Use the approach presented in lecture 4 to create a quine in your favourite programming language (or just use Python). What is the equivalent of “TM concatenation” here? Also note that the function  $q$  is often more complicated than one might think, due to character escaping.

**Exercise 3.2.** A language  $L \in P$  is complete for  $P$  under polynomial-time reductions if  $L' \leq_p L$  for every  $L' \in P$ . Show that every language in  $P$  except  $\emptyset$  and  $\Sigma^*$  is complete for  $P$  under polynomial-time reductions.

**Exercise 3.3.** Let

$$A_{PNTM} = \{ \langle M, p, w \rangle \mid M \text{ is a non-deterministic TM that accepts } w \text{ in time } p(|w|) \text{ with } p \text{ a polynomial function} \}$$

Show that  $A_{PNTM}$  is NP-complete.

**Exercise 3.4.** Show that the following problem is NP-complete:

- Input: A propositional formula  $\varphi$  in CNF  
Question: Does  $\varphi$  have at least 2 different satisfying assignments?

**Exercise 3.5.** We recall some definitions.

- Given some language  $L$ ,  $L \in \text{coNP}$  if, and only if,  $\bar{L} \in \text{NP}$ .
- $L$  is coNP-hard if, and only if,  $L' \leq_p L$  for every  $L' \in \text{coNP}$ .
- $L$  is coNP-complete if, and only if,  $L \in \text{coNP}$  and  $L$  is coNP-hard.

Show that if any coNP-complete problem is in NP, then  $\text{NP} = \text{coNP}$ .

**Exercise 3.6.** If  $G$  is an undirected graph, a *vertex cover* of  $G$  is a subset of the nodes where every edge of  $G$  touches one of those nodes. The vertex cover problem asks whether a graph contains a vertex cover of a specified size.

$$\text{VERTEX-COVER} = \{ \langle G, k \rangle \mid G \text{ is an undirected graph that has a } k\text{-node vertex cover.} \}$$

Show that **VERTEX-COVER** is NP-complete.

**Hint:**

JLY to find a reduction from 3-SAT