Complexity Theory
Exercise 3: Time Complexity
1 November 2023

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. Show that P is closed under concatenation and star.

Exercise 3.3. A language $\mathbf{L} \in$ P is complete for P under polynomial-time reductions if $\mathbf{L}' \leq_p \mathbf{L}$ for every $\mathbf{L}' \in$ P. Show that every language in P except $\emptyset$ and Σ^* is complete for P under polynomial-time reductions.

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

Input: A propositional formula φ in CNF
Question: Does φ have at least 2 different satisfying assignments?

Exercise 3.5. We recall some definitions.

- Given some language $\mathbf{L}$, $\mathbf{L} \in$ CONP if, and only if, $\overline{\mathbf{L}} \in$ NP.

- $\mathbf{L}$ is CONP-hard if, and only if, $\mathbf{L}' \leq_p \mathbf{L}$ for every $\mathbf{L}' \in$ CONP.

- $\mathbf{L}$ is CONP-complete if, and only if, $\mathbf{L} \in$ CONP and $\mathbf{L}$ is CONP-hard.

Show that if any CONP-complete problem is in NP, then NP = 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.

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

Show that **Vertex-Cover** is NP-complete.

Hint:
Try to find a reduction from 3-Sat