Quantum Complexity Theory, UPB
Summer 2020, Assignment 1

To be completed by: Tuesday, November 10, start of tutorial

1 Exercises

1. **Turing Machines (TMs).** One of the reasons the TM model is highly revered is due to its *robustness*, i.e. the fact that its computational power does not change much under small changes to the model.

   (a) In class, we defined a TM as moving its head either one cell to the left (L) or one cell to the right (R) in one computation step. Prove that adding a third option, that the head can stay (S) put in a computation step, does not increase the power of the model. In other words, show how our "standard" TM model from class can simulate the option of keeping the head stationary.

   (b) Our standard TM model assumes access to a single work tape. Another option is to allow *two* work tapes, each with its own independent head. Show that our standard TM model can simulate a 2-tape TM.

2. **Languages and decision problems.** In class, we associated each language $L \subseteq \{0,1\}^*$ with a decision problem Π_L of the form: Given $x \in \{0,1\}^*$, is $x \in L$? Let us flesh out how this framework can be used to encode real-life decision problems that *a priori* have nothing to do with languages.

   We begin with an example. Recall the Halting Problem asks: Given as input a description of a TM M and input $x \in \{0,1\}^*$, does M halt on input x? The corresponding language for this is

   $$\text{HALT} = \left\{ \langle M, x \rangle \mid M \text{ is a TM, } x \in \{0,1\}^*, \text{ and } M \text{ halts on input } x \right\}.$$

   The angle brackets $\langle \cdot \rangle$ mean we are implicitly applying a fixed encoding scheme mapping (say) the input Turing machine M to a binary string.

   (a) Convince yourself that deciding membership in HALT is equivalent to solving the Halting Problem.

   (b) Write down the language L corresponding to the problem FACTOR from lecture.

3. **P and NP.**

   (a) Prove that P $\subseteq$ NP.

   (b) Prove that P is closed under complement. In other words, prove that a language $L \subseteq \{0,1\}^*$ is in P if and only if $\overline{L} := \{0,1\}^* \setminus L$ is in P.

   (c) Is NP closed under complement? Why or potentially why not?

   (d) Suppose we change the definition of NP so that the proof y has length only *logarithmic* in the input size, n, instead of polynomial in n. (Everything else about the definition of NP does not change.) Call this class ShortNP. What can you say about ShortNP versus P?

   (e) Suppose we change the definition of NP so that the proof y has length *exponentially long* in the input size, n, instead of polynomial in n. (Everything else about the definition of NP does not change.) Call this class LongNP. What can you say about LongNP versus NP?

4. **Reductions and NP-completeness.** The language INDEPENDENT-SET is defined as follows:

$$IS = \{\langle G, k\rangle \mid G = (V, E) \text{ has an independent set } S \text{ of size at least } k\}.$$

Here, $S \subset V$ is an *independent set* if for any $v, w \in S$, v and w are not connected by an edge (i.e. $(v, w) \notin E$).

(a) Show that $IS \in \text{NP}$ by demonstrating the existence of an efficiently checkable proof for YES-instances of the problem.

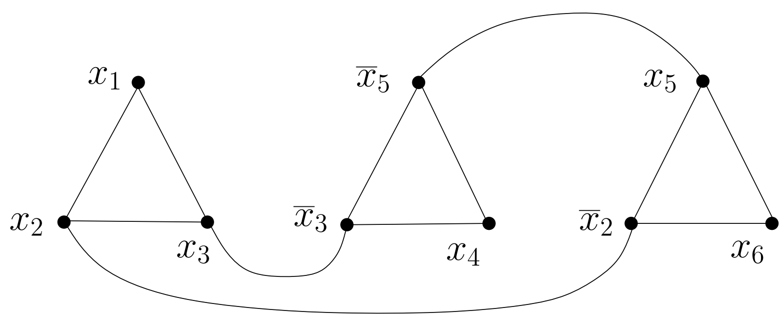
(b) Below, we sketch a reduction from 3-SAT to IS. Given a 3-CNF formula ϕ consisting of k clauses, we construct a graph G as follows:

- For each clause ϕ_i, we add a triangle gadget Δ_i to G, whose vertices correspond to the literals of ϕ_i.
- We connect vertices corresponding to conflicting literals with an edge.

For example, suppose

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_5} \vee \overline{x_3} \vee x_4) \wedge (x_5 \vee \overline{x_2} \vee x_6).$$

Then, G is given by:



i. Prove that if ϕ is satisfiable, then G has an independent set S of size k.

ii. Prove that if G has an independent set S of size k, then ϕ is satisfiable.