CS46 practice problems 11

These practice problems are an opportunity for discussion and trying many different solutions. It is **not counted towards your grade**, and **you do not have to submit your solutions.** The purpose of these problems is to get more comfortable with reasoning and writing about P and NP.

If you are stumped or looking for guidance, **ask**.

1. **Closure properties.**
   (a) Prove that P is closed under complement. (We discussed this in lecture last week!)
   (b) Prove that P is closed under concatenation.
   (c) Prove that P is closed under union.
   (d) Show that P is closed under Kleene star.
   (Hint: you may want to consider a technique similar to how we built a table in lab 6 for determining if a grammar in Chomsky Normal Form could generate a string w. You probably showed decidable or recognizable languages were closed under Kleene star using non-determinism. Languages in P can only use deterministic Turing machines that run in polynomial time.)
   (e) Prove that NP is closed under union.
   (f) Prove that NP is closed under concatenation.

2. **Boolean formulas, NP, and an application of NFAs.**

   Some useful vocabulary:

   - a **literal** is a Boolean variable or a negated Boolean variable, like x or $\overline{x}$
   - The symbol "$\vee$" means "or". (This is a **disjunction**.)
   - The symbol "$\wedge$" means "and". (This is a **conjunction**.)
   - a **clause** is a disjunction of literals, like $x \vee y \vee \overline{z}$
   - A formula is **satisfiable** if there is a truth assignment (giving a truth value to each variable) which makes the entire formula evaluate to TRUE.

   (a) Show that SATISFIABILITY $\in$ NP, where

   $$\text{SATISFIABILITY} = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable Boolean formula}\}$$

   A Boolean formula is an expression involving Boolean variables and operations, for example $(\overline{x} \wedge y) \vee (x \wedge \overline{z})$.

   (b) A Boolean formula is in **conjunctive normal form** (CNF) if it is written as the conjunction of clauses, for example:

   $$(x_1 \vee x_2) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge (x_5 \vee \overline{x}_1 \vee x_6) \wedge (x_3)$$

   Define the language:

   $$L = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable CNF formula where each variable appears at most twice}\}$$

   Show that $L \in P$.

(c) For a CNF formula ϕ with m variables and c clauses, show you can construct in polynomial time an NFA with $O(cm)$ states that accepts all *non*satisfying assignments, represented as binary strings of length m.

(This implies that if $P \neq NP$, then NFAs cannot be minimized in polynomial time.)

3. Show that if P = NP, a polynomial-time algorithm exists that produces a satisfying assignment when given a satisfiable Boolean formula.

Note: The algorithm you are being asked to write computes a function, but NP contains languages, not functions. The P = NP assumption means that SATISFIABILITY $\in$ P, so there is a deterministic polynomial-time Turing machine M_{SAT} which can test if a formula is satisfiable. You don't know how this test is done, but you may use M_{SAT} in your algorithm.