Problem Set 2
Due at the beginning of class on Sept. 22

These problems are not particularly difficult, but they will require some thought.

1. Prove that any \( \mathcal{NP} \)-complete language \( L \) (more formally, the natural relation derived from \( L \)) is self-reducible in the sense discussed in class.

2. As part of the proof that the language \( CS \) (circuit satisfiability) is \( \mathcal{NP} \)-complete, we used the fact that any boolean function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) can be encoded as a circuit (using "and", "or", and "not" gates). Prove this by giving an explicit description of a circuit that computes a given function \( f \). What is the worst-case number of gates your circuit uses as a function of \( n \), the number of input bits?

3. (optional—unsolved problem) Prove that Sudoku (or, more formally, the generalization to \( n \times n \) boards) is \( \mathcal{NP} \)-complete. Note: by using Google you can find a solution, but I am looking for a simpler solution that relies on a reduction from a more well-known \( \mathcal{NP} \)-complete problem. I don’t know how difficult it is to find a “simpler” solution, so work on this problem at your own risk!

4. Answer the following questions regarding \( \mathcal{NP} \) and \( \text{co}\mathcal{NP} \).
   (a) In class we gave two definitions of \( \text{co}\mathcal{NP} \): first, \( \text{co}\mathcal{NP} = \{ L | \tilde{L} \in \mathcal{NP} \} \), and second, \( L \in \text{co}\mathcal{NP} \) if there exists a poly-time computable (and polynomially-bounded) relation \( R \) such that \( x \in L \iff \forall w : R(x, w) \). Prove that these definitions of \( \text{co}\mathcal{NP} \) are equivalent.
   (b) Let \( L \) be an \( \mathcal{NP} \)-complete language. Show a Cook reduction from any language \( L' \in \text{co}\mathcal{NP} \) to \( L \). Why is your reduction not a Karp reduction?
   (c) Show that if there exists a language \( L \in \mathcal{NP} \) such that there is a Karp reduction from every language \( L' \in \text{co}\mathcal{NP} \) to \( L \), then \( \mathcal{NP} = \text{co}\mathcal{NP} \).

5. For any complexity class \( \mathcal{C} \), show that \( \text{co}\mathcal{C} \subseteq \mathcal{C} \) implies \( \text{co}\mathcal{C} = \mathcal{C} \).

6. Recall that for any language \( L \), the Kleene star of the language is
   \[ L^* \overset{\text{def}}{=} \{ x_1 \cdots x_k \mid x_i \in L, k \geq 0 \}. \]

   A complexity class \( \mathcal{C} \) is closed under Kleene star if \( L \in \mathcal{C} \) implies \( L^* \in \mathcal{C} \). Prove that \( \mathcal{P}, \mathcal{NP}, \) and \( \text{co}\mathcal{NP} \) are closed under Kleene star. (Note: this problem is more difficult than the rest, but not too difficult to be optional!)