Homework 9

Out: 9 Apr. Due: 16 Apr.

Instructions: *Submit your solutions in pdf format on Gradescope by* **5pm on Friday, April 16***. Solutions may be written either in LaTeX (with either machine-drawn or hand-drawn diagrams) or* **legibly** *by hand. (The LaTeX source for this homework is provided in case you want to use it as a template.) Please be sure to begin the solution for each problem on a new page, and to tag each of your solutions to the correct problem! Per course policy, no late solutions will be accepted. Take time to write* **clear** *and* **concise** *answers; confused and long-winded solutions may be penalized. You are encouraged to form small groups (two to four people) to work through the homework, but you* **must** *write up all your solutions on your own. Depending on grading resources, we reserve the right to grade a random subset of the problems and check off the rest; so you are advised to attempt all the problems.*

1. Your friend who works for a software company gives you a black box, which she claims has the following properties:[1]

- The box computes a bijection f from n-bit integers to n-bit integers, for any n (i.e., given as input an n-bit integer x, the box outputs an n-bit integer $f(x)$ such that $f(x) \neq f(y)$ when $x \neq y$).
- The box runs in polynomial time (i.e., given as input an n-bit integer, the running time is $O(n^k)$ for some fixed constant k).
- The inverse function f^{-1} *cannot* be computed in polynomial time.

(a) Show that, if your friend's claims are correct, then it must be the case that $\mathrm{P} \neq \mathrm{NP}$.
[HINT: Show that the language $L = \{(x, y) : f^{-1}(x) < y\}$ belongs to $\mathrm{NP} \setminus \mathrm{P}$. Note that you need to show separately that L beongs to NP and that L does not belong to P.]

(b) Show further that, if your friend's claims are correct, then $\mathrm{NP} \cap$ co-NP $\neq \mathrm{P}$.

2. Solve Problem 8.15 in Sipser. This problem asks you to show that determining who has a winning strategy in a certain two-person game called *PUZZLE* is PSPACE-complete.

[HINT: Don't forget to show that the problem is in PSPACE! To show hardness, try a reduction from TQBF or FORMULA_GAME.]

[continued on next page]

[1] Your friend says that this box can help you in cryptographic applications, because using f you can construct a "code" $f(x)$ from which x cannot be efficiently recovered.

3. The *in-place acceptance problem* for Turing machines, $\mathit{IPA}_{\mathrm{TM}}$ is the problem of deciding whether a given (one-tape, deterministic) Turing machine accepts a given input w without moving its head outside the leftmost $|w| + 1$ squares of its tape.

(a) Show that $\mathit{IPA}_{\mathrm{TM}}$ is PSPACE-complete. [HINTS: Don't forget to show that $\mathit{IPA}_{\mathrm{TM}}$ belongs to PSPACE. To show hardness, do a straightforward, direct reduction from any $L \in$ PSPACE by "padding" the input with null symbols. There should be no need for any complex construction here.]

(b) A *linear-bounded automaton* (LBA) is a one-tape *nondeterministic* Turing machine that on input w never moves its head outside the leftmost $c|w|$ squares of its tape, for some constant c[2]. Show that the problem of deciding whether a given string w is accepted by a given LBA is PSPACE-complete.

[2]LBAs are important because they recognize precisely the class of *context-sensitive* languages, i.e., languages defined by context-sensitive grammars (CSGs). CSGs are a generalization of context-free grammars in which production rules may be applied only when the non-terminal to be rewritten appears in a certain "context", i.e., is surrounded by certain specified other symbols.