

This homework has 8 questions, for a total of 100 points and 5 extra-credit points.

Unless otherwise stated, each question requires *clear*, *logically correct*, and *sufficient* justification to convince the reader.

For bonus/extra-credit questions, we will provide very limited guidance in office hours and on Piazza, and we do not guarantee anything about the difficulty of these questions.

We strongly encourage you to typeset your solutions in L^AT_EX.

If you collaborated with someone, you must state their name(s). You must *write your own solution* for all problems and *may not use any other student's write-up*.

(0 pts) 0. **Before you start; before you submit.**

If applicable, state the name(s) and username(s) of your collaborator(s).

Solution:

(10 pts) 1. **Self assessment.**

Carefully read and understand the posted solutions to the previous homework; you may also find the video “walkthroughs” in the Canvas Media Gallery helpful. Identify one part for which your own solution has the most room for improvement (e.g., has unsound reasoning, doesn't show what was required, could be significantly clearer or better organized, etc.). Copy or screenshot this solution, then in a few sentences, explain what was deficient and how it could be fixed.

(Alternatively, if you think one of your solutions is significantly *better* than the posted one, copy it here and explain why you think it is better.)

If you didn't turn in the previous homework, then (1) state that you didn't turn it in, and (2) pick a problem that you think is particularly challenging from the previous homework, and explain the answer in your own words. You may reference the answer key, but your answer should be in your own words.

Solution:

2. **Deciding undecidability.**

For each of the following languages, show that it is undecidable via a Turing reduction from a language we have already shown is undecidable.

(As a refresher, you may wish to re-read Handout 2: Turing Reductions.)

- (10 pts) (a) $L_{\text{DISJOINT}} = \{(\langle M_1 \rangle, \langle M_2 \rangle) : M_1, M_2 \text{ are TMs and } L(M_1) \cap L(M_2) = \emptyset\}.$

Solution:

- (10 pts) (b) Let $S = \{\langle 183 \rangle, \langle 280 \rangle, \langle 281 \rangle, \langle 370 \rangle, \langle 376 \rangle\}$ and define the language

$L_{\text{EECS}} = \{\langle M \rangle : M \text{ is a TM that accepts every string in } S \text{ and loops on every other string}\}.$

Solution:

3. **NP languages.** Prove that each of the following languages is in NP.

- (7 pts) (a) $\text{AFFORDABLEMENU} = \{\langle L, k \rangle : L \text{ is a list of menu items with positive integer prices, and there are at least } k \text{ distinct menu items that have total price exactly } 10k\}.$

Solution:

- (7 pts) (b) $\text{INDEPENDENTSET} = \{\langle G, k \rangle : G \text{ is an undirected graph with an independent set of size } k\}.$
An independent set in a graph is a subset C of the vertices for which the graph has no edge between any pair of vertices in C .

Solution:

4. **Understanding P, NP, and poly-time mapping reductions.**

- (7 pts) (a) We claim that $P \subseteq NP$, i.e., for any language $L \in P$, we have that $L \in NP$ as well. Here is an *incomplete* proof of this fact, which you will complete.

By the hypothesis that $L \in P$, there is an efficient Turing machine M that decides L . We define the following efficient verifier V for L .

- 1: **function** $V(x, c)$
- 2: **[MISSING PSEUDOCODE]**

State what the missing pseudocode should be, and prove that V is indeed an efficient verifier for L .

Solution:

- (10 pts) (b) Show that poly-time mapping reductions are transitive. That is, if $A \leq_p B$ and $B \leq_p C$, then $A \leq_p C$.

Solution:

- (10 pts) (c) Let $C = A \cup B$ where both A, B are languages in NP. State, with proof, whether C is in NP for *all*, *some* (but not all), or *no* such A, B .

Solution:

5. Understanding coNP.

This question explores the complexity class $\text{coNP} = \{\bar{L} : L \in \text{NP}\}$, i.e., the set of languages whose complement languages are in NP. (Note that this is *not* the complement of NP itself, which is the set of languages that are not in NP.)

Recall that conceptually, NP is the set of languages whose “yes” instances can be verified efficiently, given suitable certificates. Symmetrically, coNP is the set of languages whose “no” instances can be verified efficiently (given suitable certificates).

- (5 pts) (a) Prove that P is closed under set complement. That is, for any $L \in \text{P}$, we have that $\bar{L} \in \text{P}$.

Solution:

- (9 pts) (b) Use the previous part to prove that if $\text{P} = \text{NP}$, the following set inclusions hold: (i) $\text{NP} \subseteq \text{coNP}$ (that is, for any $L \in \text{NP}$, we have $L \in \text{coNP}$), and (ii) $\text{coNP} \subseteq \text{NP}$.

Solution:

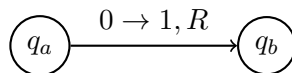
- (5 pts) (c) Conclude from the previous part that if $\text{NP} \neq \text{coNP}$, then $\text{P} \neq \text{NP}$.
(Therefore, you could collect the \$1 million Clay Math prize “simply” by proving that $\text{NP} \neq \text{coNP}$!)

Solution:

(10 pts) 6. Cook-Levin windows.

Recall from lecture that one critical ingredient in the proof of the Cook-Levin theorem is to specify which contents of the 2×3 “windows” of the verifier’s computation tableau are valid.

Specify *all* valid 2×3 window contents that could appear in two adjacent rows of the tableau, if those rows represent the verifier TM taking the following transition:



You do not need to provide justification.

For simplicity, you can assume that the symbols in the cells of the window are restricted to $\{0, 1, \perp, q_a, q_b\}$. Recall that there can be other symbols in the rows, such as the “boundary” symbol # and “separator” symbol \$, but we will ignore those here. To avoid explicitly writing out too many windows, you can use notation like $0/1/\perp$ to indicate that a cell could contain either 0, 1, or \perp . Also, to specify that a cell in the bottom row of a window must contain the same character as the cell above it, you can write “same”. For example, writing

0/1	\perp	1
same	0	\perp

represents the following two windows:

0	\perp	1
0	0	\perp

1	\perp	1
1	0	\perp

Be sure to include all valid window contents that represent the head of the TM being in *both* rows of the window, all those that represent the head being in *only one* of the rows of the window, and all those that represent the head not being in the window at all. Do not include any invalid windows.

Solution:

(5 EC pts) 7. **Optional extra-credit problem: NP in exponential time.**

Let EXP be the class of all languages that are decidable in exponential time, i.e., in time $O(2^{n^k})$ for some constant k (where n is the length of the input).

It remains unknown whether $\text{NP} = \text{EXP}$, but it is known that $\text{P} \neq \text{EXP}$. Prove that $\text{NP} \subseteq \text{EXP}$. In other words, show that any problem that can be verified efficiently can be decided in exponential time.

Solution: