Alex Hubers
CSC:341 Spring 2024

Exam #3 (Complexity)

Due: May 14, 2024

Academic Honesty & Collaboration. I expect you to work alone, and for your work to represent wholly your own efforts. You are, of course, allowed to ask your classmates (and me) general questions about the material. Please include the following information at the top of your submission, along with your name.

- Written sources used: (Include textbook(s), complete citations for web or other written sources. Write none if no sources used)
- Help obtained: (Include names of anyone other than the instructor.)

Extensions. I am happy to offer extensions, but I ask that you ask *in advance* and at least prior to *Friday, May 10*. Further, I expect you to be able to show either (i) proof of effort and some progress on the exam, or (ii) some minor proof of conflict in schedule.[1] (Exceptions will be made, of course, in cases of sickness, injury, or emergency.) Finally, please talk to me *as soon as possible* if you know you have a conflict in schedule (e.g., another take-home exam in a separate course) and you need some help meeting this deadline. *Unlike in the past, I will be enforcing the policy outlined above.*

The latest extension I *can* give is to Thursday, May 16. Unlike the rest of my deadlines, this is a hard one: as per college policy, the course is done, and I cannot ask you to keep doing work. In particular, I need to hand in your grades by Saturday—like, in an envelope, in person, literal hand-in. So I need to drive to Grinnell no later than Friday, May 17 and *also* have all senior exams graded by then.

Without exception, I will not be accepting work past EOD Thursday, May 16^{th}.

Learning outcomes. As per the syllabus, this exam is designed to test your mastery of the following course learning outcomes.

1. Prove that a problem is in a particular time complexity class.
2. Prove that a problem is NP-complete by way of a reduction.
3. Explain the practical ramifications of the P vs. NP problem.
4. Prove that a problem is in a particular space complexity class.
5. Describe the essential characteristics of problems belonging to each of the major complexity classes.
6. Describe the practial relationships between various time and space complexity classes.

[1] Obviously, in case of accommodations, you may have until May 16 without questions asked.

#1 (50 pts)

Define coP, the complement of P, as

$$\mathsf{coP} = \{L \mid \overline{L} \in \mathsf{P}\}.$$

In this question, we will prove (and think about) the following claim.

Claim 1. *$\mathsf{P} = \mathsf{coP}$.*

Specifically:

1. (30 pts). Prove that $\mathsf{P} \subseteq \mathsf{coP}$ and $\mathsf{coP} \subseteq \mathsf{P}$. *Hint: both directions should be similar.*
2. (10 pts). You have just proven an important property of languages in P. What is that property? (Fill in the blanks in this sentence: *Languages in P are* ______ *under* ______.)
3. (10 pts). Define coNP similarly to coP as

$$\mathsf{coNP} = \{L \mid \overline{L} \in \mathsf{NP}\}.$$

   Argue (in 1-3 sentences) why your proof that $\mathsf{P} = \mathsf{coP}$ <u>cannot</u> be replicated or modified to show that $\mathsf{NP} = \mathsf{coNP}$.

1. Let $A \in P$ and M be a TM that decides P. Modify M to create M' by turning q_{accept} into q_{reject} and vice versa. Observe that $L(M') = \overline{A} \in \mathsf{coP}$ and M' runs in polynomial time because it is identical to M except for its acceptance and rejection states.

   The other direction—that $\mathsf{coP} \subseteq \mathsf{P}$—is identical but dual.
2. We have proven that languages in P are <u>closed</u> under <u>complement</u>.
3. Applying this construction to show $\mathsf{NP} = \mathsf{coNP}$ does not work because of the asymmetric nature of nondeterminism. Only some branch of a nondeterministic computation must accept for the overall Turing machine to accept. However, all nondeterministic branches must fail for the overall Turing machine to fail. Because of this, we can't simply flip states—we must conceive of a way to "collect up" all the results of nondeterministic branches and analyze. However, nondeterminism does not allow us to share information between branches.

#2 (50 pts)

Prove that, if every NP-hard language is also PSPACE-hard, then $\mathsf{PSPACE} = \mathsf{NP}$.

Hint 1: We know already that $\mathsf{NP} \subseteq \mathsf{PSPACE}$. So your obligation is to prove the other direction.

Hint 2: Let A be arbitrary such that $A \leq_p \mathrm{SAT}$. Is $A \in \mathsf{NP}$?

First, we know already that $\mathsf{NP} \subseteq \mathsf{PSPACE}$. So we need only show the other direction: that $\mathsf{PSPACE} \subseteq \mathsf{NP}$.

Let $A \in \mathsf{PSPACE}$ be arbitrary. As SAT is NP-hard, it follows by assumption that SAT is also PSPACE-hard. So we may conclude that $A \leq_p \mathrm{SAT}$—meaning we have a nondeterministic polynomial time decider for A, and so $A \in \mathsf{NP}$. This is enough to conclude that $\mathsf{NP} = \mathsf{PSPACE}$.

#3 (50 pts)

Define the language ADD as

$$\text{ADD} = \{a\#b\#c \mid a, b, c \text{ are binary natural numbers and } a + b = c\}$$

Prove that $\text{ADD} \in \mathsf{L}$. Specifically, describe (at a very-high level) how a TM M can decide ADD and argue that the amount of space it needs on its work tape is sublinear.[2]

I'll sketch the idea.

1. M checks if $c_0 = a_0 \,\texttt{XOR}\, b_0$, where each a_i, b_i, and c_i denote the i^{th} bit of a, b, and c, counting from the right. M uses an a cell on the work-tape mark if $a_0 \,\texttt{AND}\, b_0$ (denoting the carry).
2. M repeats, moving leftwards on a, b, and c, and doing an appropriate calculation. M is just *validating* that c is added correctly.
3. Reject if some calculation is wrong. Otherwise continue until we have parsed all of c.

The description above is not great, but it demonstrates that we need only 1 cell to verify the calculation. That is, we only need to store the carry. The rest of the validation can be done using states (recall how we used states to do incrementation in a previous exam).

#4 (50 pts)

The following claim is true, and follows from the Time Hierarchy Theorem, which we did not have time to cover.

Claim 2. *There exists a language $A \in \mathsf{EXPTIME}$ that is not in P.*

Using Claim 2, prove that, if $\mathsf{NP} = \mathsf{EXPTIME}$, then $\mathsf{P} \neq \mathsf{NP}$.

It is important to remember that we know the following to be true.

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{EXPTIME}$$

Now, suppose $\mathsf{P} \subseteq \mathsf{EXPTIME} = \mathsf{NP}$. As $\mathsf{P} \subseteq \mathsf{EXPTIME}$, we know that $\mathsf{P} = \mathsf{NP}$ iff $\mathsf{EXPTIME} \subseteq \mathsf{P}$. But this cannot be true, by claim 2. Hence $\mathsf{P} \neq \mathsf{EXPTIME}$ and, as $\mathsf{EXPTIME} = \mathsf{NP}$, we must have $\mathsf{P} \neq \mathsf{NP}$.

#5 (30 pts)

Mark each answer `true`, `false`, or `unknown`.

(This question is placed on the exam to make up the points lost from omitting Lab 14B, Quiz 14, and HW14.)

1. (5 pts). $\mathsf{NP} = \mathsf{EXPTIME}$.
2. (5 pts). SAT $\in \mathsf{P}$.
3. (5 pts). If 3SAT $\in \mathsf{P}$ then $\mathsf{P} = \mathsf{NP}$.
4. (5 pts). $\mathsf{HALT}_{\text{TM}} \in \mathsf{P}$.

[2] For our purposes, sublinear means that the number of cells used on the work-tape are $\mathcal{O}(\log n)$ or $\mathcal{O}(1)$ on an input length of n.

5. (5 pts). SAT is undecidable.

6. (5 pts). If TQBF $\leq_p$ CLIQUE, then CLIQUE is PSPACE-complete.

1. `unknown`.

2. `unknown`.

3. `true`, 3SAT is NP-complete.

4. `false`, as $\mathsf{HALT}_{\mathrm{TM}}$ is undecidable.

5. Demonstrably `false`.

6. This is actually `true`. CLIQUE is in NP, and NP $\subseteq$ PSPACE, so CLIQUE $\in$ PSPACE. Therefore a reduction from TQBF (which is PSPACE-complete) to CLIQUE would mean CLIQUE is PSPACE-complete.